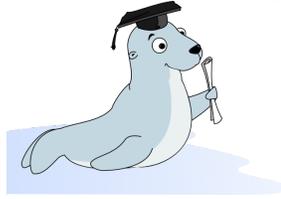


Guia Foca Linux



Gleydson Mazioli da Silva



Guia Foca Linux:

Gleydson Mazioli da Silva

Versão 7.00 - sábado, 11 de janeiro de 2020

data de publicação sábado, 11 de janeiro de 2020

Copyright © 1999-2020 - Gleydson Mazioli da Silva

Resumo

Este guia tem por objetivo ser uma referência ao aprendizado do usuário e um manual de consulta, operação e configuração de sistemas Linux (e outros tipos de *ix). A última versão oficial deste guia pode ser encontrada na Página Oficial do Foca Linux [<http://www.guiafoca.org>]. Novas versões são lançadas com uma frequência mensal e você pode receber avisos de novos lançamentos deste guia preenchendo um formulário na página Web ou assinando o twitter @guiafoca [<http://twitter.com/guiafoca>].

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 published by the Free Software Foundation; A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice

1. Introdução	1
Considerações sobre o nível Avançado	2
Pré-requisitos para a utilização deste guia	3
O Linux	3
Algumas Características do Linux	4
2. Explicações Básicas	7
Monitorando os logs	7
Destruindo arquivos/partições de forma segura	7
3. Hardware	9
Placa de expansão	9
Nomes de dispositivos	9
Configuração de Hardware	10
IRQ - Requisição de Interrupção	10
DMA - Acesso Direto a Memória	12
I/O - Porta de Entrada/Saída	13
Hardwares configuráveis por jumpers, dip-switches, jumperless e Plug-and-Play.	14
Jumpers	14
Dip-Switches	14
Jumperless (sem jumper)	14
Plug-and-Play	15
Listando as placas e outros hardwares em um computador	15
Conflitos de hardware	15
Barramento	16
Placas on-board / off-board	17
Hardwares específicos ou "For Windows"	18
Dispositivos específicos para GNU/Linux	19
Configurações de Dispositivos	19
Configurando uma placa de rede	19
Configurando uma placa de SOM no Linux	20
Configurando um gravador de CD/DVD no Linux	21
Configurando o gerenciamento de energia usando o APM	23
Configurando o gerenciamento de energia usando ACPI	23
Ativando WakeUP on Lan	24
Aterramento	24
Condutores de eletricidade	25
Tomadas	25
Descargas estáticas	26
Melhoria de performance	26
Particionamento	26
Spindles	27
Fazendo ajustes finos de performance do disco	27
Data de acesso a arquivos/diretórios	30
Periféricos SATA	30
Periféricos SCSI	30
Configurando uma SCSI ID e terminação	31
4. Rede	33
O que é uma rede	33
Protocolo de Rede	33
Endereço IP	33
Classes de Rede IP	34
Referência rápida de máscara de redes	35

Para instalar uma máquina usando o Linux em uma rede existente	35
Endereços reservados para uso em uma rede Privada	36
Interface de rede	36
A interface loopback	37
Atribuindo um endereço de rede a uma interface (ifconfig)	37
Roteamento	37
Configurando uma rota no Linux	38
Resolvedor de nomes (DNS)	39
O que é um nome?	39
Arquivos de configuração usados na resolução de nomes	40
Executando um servidor de nomes	42
Serviços de Rede	42
Serviços iniciados como Daemons de rede	43
Serviços iniciados através do inetd	43
Segurança da Rede e controle de Acesso	45
/etc/ftpusers	46
/etc/securetty	46
O mecanismo de controle de acessos tcpd	46
Firewall	50
Outros arquivos de configuração relacionados com a rede	50
/etc/services	50
/etc/protocols	51
Camadas de Rede	51
RFCs de referência sobre protocolos de rede	52
5. Configurações especiais de Rede	53
IP Alias	53
Bridge	54
Requerimentos para a Instalação	55
Configuração da bridge	55
Configurações mais avançadas de bridge	56
Configuração manual da bridge	56
Usando o iptables para construir um firewall na máquina da bridge	57
Filtrando pacotes não IP na bridge	57
Conectando dois computadores usando a porta paralela	58
Construindo um cabo LapLink Paralelo	59
Conectando dois computadores usando a porta serial	60
Construindo um cabo LapLink Serial	61
6. Arquivos e daemons de Log	63
Formato do arquivo de log	63
Daemons de log do sistema	63
syslogd	63
klogd	67
logger	68
Programas úteis para monitoração e gerenciamento de arquivos de logs	69
logcheck	69
logrotate	69
Configurando um servidor de logs	71
7. A distribuição Debian GNU/Linux	73
Porque usar a Debian?	73
Pacotes existentes na Debian	74
O que é sid/testing/frozen/stable?	74
Como obter a Debian	75
Programas de configuração	75
Arquivos de inicialização	75

Níveis de Execução	76
Entendendo o funcionamento dos níveis de execução do sistema (runlevels)	76
Rede no sistema Debian	77
Bug tracking system	77
Onde encontrar a Debian para Download?	78
Lista de pacotes para uma instalação rápida e manual	78
Pacotes Básicos (Altamente Recomendado)	79
Compilação do Kernel e programas em linguagem C	79
X11 (básico)	79
Window Managers para o X	79
Impressão (texto e gráfico com sistema de spool)	79
Som (mixer, mp3, Midi, wav, CD-Player)	79
Programas de Internet (clientes)	79
Acessórios	80
Rede	80
8. Personalização do Sistema	81
Variáveis de Ambientes	81
Modificando o Idioma usado em seu sistema	81
alias	82
Arquivo /etc/profile	82
Arquivo .bash_profile	83
Arquivo .bashrc	83
Arquivo .hushlogin	83
Arquivo /etc/environment	84
Diretório /etc/skel	84
9. Impressão	85
Portas de impressora	85
Imprimindo diretamente para a porta de impressora	85
Imprimindo via spool	85
Impressão em modo gráfico	87
Ghost Script	87
Magic Filter	89
Instalação e configuração do Magic Filter	89
Outros detalhes técnicos sobre o Magic Filter	89
Impressão remota	90
Dando permissão para impressão remota via lpd/lprng	90
Impressão via rlpr	91
Impressão via printcap	91
10. Firewall iptables	93
Introdução	93
Versão	93
Um resumo da história do iptables	94
Características do firewall iptables	94
Ficha técnica	94
Requerimentos	95
Arquivos de logs criados pelo iptables	95
Instalação	95
Enviando Correções/Contribuindo com o projeto	95
O que aconteceu com o ipchains e ipfwadm ?	95
Tipos de firewalls	95
O que proteger?	96
O que são regras?	97
O que são chains?	97
O que são tabelas?	97

Habilitando o suporte ao iptables no kernel	98
Ligando sua rede interna a Internet	99
Manipulando chains	99
Adicionando regras - A	99
Listando regras - L	100
Apagando uma regra - D	102
Inserindo uma regra - I	102
Substituindo uma regra - R	102
Criando um novo chain - N	103
Renomeando um chain criado pelo usuário - E	104
Listando os nomes de todas as tabelas atuais	104
Limpando as regras de um chain - F	105
Apagando um chain criado pelo usuário - X	105
Zerando contador de bytes dos chains - Z	105
Especificando a política padrão de um chain - P	106
Outras opções do iptables	106
Especificando um endereço de origem/destino	106
Especificando a interface de origem/destino	107
Especificando um protocolo	108
Especificando fragmentos	110
Especificando uma exceção	110
Especificando um alvo	111
Salvando e Restaurando regras	115
A tabela nat (Network Address Translation) - fazendo nat	115
Criando um novo chain na tabela NAT	116
Fazendo IP masquerading (para os apressados)	116
Fazendo SNAT	116
Fazendo DNAT	118
Monitorando conexões feitas na tabela nat	119
A tabela mangle	119
Especificando o tipo de serviço	119
Especificando o TOS para tráfego de saída	120
Outros módulos do iptables	120
Conferindo de acordo com o estado da conexão	120
Limitando o número de vezes que a regra confere	121
Proteção contra ping da morte	121
Proteção contra syn flood	122
Proteção contra IP spoofing	122
Especificando múltiplas portas de origem/destino	123
Especificando o endereço MAC da interface	123
Conferindo com quem criou o pacote	123
Conferindo com o conteúdo do pacote	124
Conferindo com o tempo de vida do pacote	124
Conferindo com números RPC	125
Conferindo com tipo de pacote	125
Conferindo com o tamanho do pacote	125
Caminho percorrido pelos pacotes nas tabelas e chains	126
Ping de 192.168.1.1 para 192.168.1.1	126
Conexão FTP de 192.168.1.1 para 192.168.1.1	127
Conexão FTP de 192.168.1.1 para 192.168.1.4	128
Conexão FTP de 200.217.29.67 para a máquina ftp.debian.org.br	129
Ping de 192.168.1.4 para 192.168.1.1	130
Conexão FTP de 192.168.1.4 para 192.168.1.1	130
Conexão FTP de 192.168.1.4 para ftp.debian.org.br	131

Conexão FTP de 200.198.129.162 para 200.217.29.167	132
Gráfico geral da passagem dos pacotes	133
Exemplos de configurações do iptables	133
Bloqueando conexões de fora para sua máquina	133
Monitorando tentativa de conexão de trojans em sua máquina	134
Conectando sua rede interna a Internet	134
Um exemplo de firewall simples	135
11. Gerenciamento de contas e cuidados para a proteção de senhas	139
Introdução	139
Criação, monitoramento e segurança de contas	139
Definindo valores padrões de restrição	140
Senhas fáceis de adivinhar e escolha de boas senhas	141
Atualização de senhas de múltiplas contas	142
A senha do usuário root	142
Tipos de ataques mais comuns para se conseguir uma senha.	143
Dedução	143
Engenharia Social	143
Ataques por dicionário	144
Brute Force	145
Monitoramento de toques do teclado	145
Login falso	145
Melhorando a segurança das senhas armazenadas em seu sistema	145
Shadow Passwords	145
Senhas MD5	146
12. Apache	147
Introdução	147
Versão	147
Um resumo da História do Apache	147
Enviando Correções/Contribuindo com o projeto	148
Características do Apache	148
Ficha técnica	148
Requerimentos	149
Arquivos de log criados pelo Apache	149
Instalação	150
Iniciando o servidor/reiniciando/recarregando a configuração	150
Opções de linha de comando	151
Configurações Básicas do Apache	151
Configurando a porta padrao	151
Adicionando uma página no Apache	152
Configurando as interfaces que o Apache atenderá	153
Especificando endereços/portas adicionais (a diretiva <i>Listen</i>)	153
Restrições de Acesso	153
Especificando opções/permisões para as páginas	153
Autorização	158
Autenticação	162
Usando autorização e autenticação juntos	166
O arquivo <code>.htaccess</code>	167
Usando a diretiva <code>SetEnvIf</code> com <code>Allow</code> e <code>Deny</code>	169
A diretiva <code><Limit></code>	169
Diretiva <code><LimitExcept></code>	170
Definindo documentos de erro personalizados	170
Módulos DSO	171
Sistema de Log do Apache	174
AgentLog	174

ErrorLog	174
CustomLog	174
RefererLog	175
RewriteLog	175
RewriteLogLevel	175
ScriptLog	175
ScriptLogBuffer	175
ScriptLogLength	176
LogFormat	176
TransferLog	177
LogLevel	177
Anonymous_LogEmail	178
CookieLog	178
Relatório gráfico de acesso ao sistema	178
Configurando o Apache como servidor proxy	178
Controlando o acesso ao servidor proxy	180
Redirecionamento de conexões no Apache	181
Virtual Hosts	181
Virtual hosts baseados em IP	182
Virtual hosts baseados em nome	183
Segurança no uso de IP's em Virtual Hosts	184
Uso de criptografia SSL	185
Servidor apache com suporte a ssl	185
Instalando o suporte a módulo SSL no Apache	186
Gerando um certificado digital	186
Exemplo de configuração do módulo mod-ssl	186
Autorizando acesso somente a conexões SSL	188
Iniciando o servidor Web com suporte a SSL	188
Exemplo comentado de um arquivo de configuração do Apache	189
httpd.conf	189
srm.conf	197
access.conf	204
Códigos HTTP	208
13. Servidor ident	210
Introdução	210
Versão	210
Contribuindo	210
Características	210
Ficha técnica	210
Requerimentos de Hardware	211
Arquivos de log criados pelo Ident	211
Instalação	212
Instalação via Inetd	212
Usando tcpwrappers com oidentd	212
Iniciando o servidor/reiniciando/recarregando a configuração	213
Opções de linha de comando	213
Exemplos	214
14. Servidor telnet	215
Introdução	215
Versão	215
Características	215
Ficha técnica	215
Requerimentos de Hardware	216
Arquivos de log criados pelo servidor telnet	216

Instalação	216
Iniciando o servidor/reiniciando/recarregando a configuração	216
Opções de linha de comando	216
Controle de acesso	217
Recomendações	217
Fazendo conexões ao servidor telnet	217
15. Servidor ssh	218
Introdução	218
Versão	218
História	218
Contribuindo	218
Características	218
Ficha técnica	219
Requerimentos de Hardware	220
Arquivos de log criados pelo servidor ssh	220
Instalação do servidor openSSH	220
Iniciando o servidor/reiniciando/recarregando a configuração	220
Opções de linha de comando	220
Usando aplicativos clientes	221
ssh	221
scp	223
sftp	224
Servidor ssh	225
sshd	225
Controle de acesso	225
Usando autenticação RSA/DSA - chave pública/privada	225
Execução de comandos específicos usando chaves	227
Criando um gateway ssh	227
Criando um tunel proxy	227
Diferenças nas versões do protocolo	228
Exemplo de <code>sshd_config</code> com explicações das diretivas	229
16. Servidor pop3	233
Introdução	233
Versão	233
Contribuindo	233
Características	233
Ficha técnica	233
Requerimentos de Hardware	234
Arquivos de log criados pelo qpopper	234
Instalação	234
Iniciando o servidor/reiniciando/recarregando a configuração	234
Teste de acesso no pop3	235
Opções de linha de comando	235
Enviando boletins de mensagens	236
Especificando quotas para as caixas de correio	237
Restringindo acesso ao servidor pop3	237
17. CVS	238
Introdução ao CVS	238
Versão	238
História	238
Contribuindo com o CVS	239
Características	239
Ficha técnica	239
Requerimentos de Hardware	240

Arquivos de log criados pelo CVS	240
Instalação	240
Iniciando o servidor/reiniciando/recarregando a configuração	240
Opções de linha de comando	241
Servidor de CVS - configurando métodos de acesso ao repositório	241
local	242
fork	242
ext	242
pserver (password server)	243
Configurando um servidor pserver	244
gssapi	247
Criando projetos para serem usados no CVS	247
Repositório	247
Criando um repositório	248
Logando no servidor de CVS via pserver	248
Encerrando uma seção de CVS	249
Baixando arquivos	249
Adicionando um novo projeto	249
Sincronizando a cópia remota com a cópia local	250
Enviando as mudanças para o servidor remoto	250
Adicionando um arquivo ao módulo CVS do servidor	251
Adicionando um diretório ao módulo CVS do servidor	251
Removendo um arquivo do módulo CVS remoto	251
Removendo um diretório do módulo CVS remoto	251
Dizendo que o módulo atual não está mais em uso	252
Visualizando diferenças entre versões de um arquivo	252
Visualizando o status de versão de arquivos	252
Outros utilitários para trabalho no repositório	253
Arquivos administrativos em CVSROOT	253
config	253
modules	253
cvswrappers	253
commitinfo	254
verifymsg	254
loginfo	254
cvsignore	254
checkoutlist	254
history	254
Clientes de CVS	254
cvs	254
gcvs - Linux	254
WinCVS - Windows	256
MacCVS - Macintosh (PPC)	256
viewcvs	256
Exemplo de uma seção CVS	256
18. SAMBA	258
Introdução	258
Versão documentada	258
História	258
Contribuindo	259
Características	259
Ficha técnica	260
Requerimentos de Hardware	260
Arquivos de log criados	260

Instalação	261
Iniciando o servidor/reiniciando/recarregando a configuração	261
Opções de linha de comando	261
Conceitos gerais para a configuração do SAMBA	262
Nome de máquina (nome NetBios)	262
Grupo de trabalho	262
Domínio	263
Compartilhamento	263
Mapeamento	263
Navegação na Rede e controle de domínio	263
Arquivo de configuração do samba	264
Seção [global]	265
Seção [homes]	270
Seção [printers]	271
Buscando problemas na configuração	271
Níveis de sistema para eleição de rede	272
Variáveis de substituição	273
Compartilhamento de arquivos e diretórios	274
Descrição de parâmetros usados em compartilhamento	274
Configuração em Grupo de Trabalho	277
Resolução de nomes de máquinas no samba	279
Arquivo /etc/samba/lmhosts	279
WINS	280
Servidor de data/hora	282
Configuração do serviço de data/hora no SAMBA	282
Sincronizando a data/hora no Cliente	282
Configuração em Domínio	282
Uma breve introdução a um Domínio de rede	283
Local Master Browser	283
Domain Master Browser	283
Configurando um servidor PDC no SAMBA	284
Contas de máquinas de domínio	285
Criando uma conta de administrador de domínio	286
Criando Scripts de logon	287
Configurando perfis de usuários	289
Modificações de permissões de acesso pelos clientes do domínio	290
Gerenciamento de senhas	290
Ativando o suporte a senhas criptografadas	290
Ativando o suporte a senhas em texto plano	294
Mapeamento de usuários/grupos em clientes	296
Mapeamento de usuários/grupos domínio em Windows	296
Mapeamento de usuários/grupos domínio em Linux	296
Compartilhamento de impressão no servidor SAMBA	298
Configurando o Linux como um servidor de impressão Windows	298
Controle de acesso ao servidor SAMBA	299
Nível de acesso de usuários conectados ao SAMBA	299
Restringindo o acesso por IP/rede	299
Restringindo o acesso por interface de rede	300
Restringindo o acesso por usuários	301
Evite o uso do parâmetro <i>hosts equiv!</i>	301
Evite o uso de senhas em branco!	302
Criando um compartilhamento para acesso sem senha	302
Criando um compartilhamento com acesso somente leitura	303
Criando um compartilhamento com acesso leitura/gravação	303

Excessão de acesso na permissão padrão de compartilhamento	304
Restringindo o IPC\$ e ADMIN\$	304
Criando um compartilhamento invisível	305
Executando comandos antes e após o acesso ao compartilhamento	306
Considerações de segurança com o uso do parâmetro "public = yes"	306
Senhas criptografadas ou em texto puro?	307
Mapeamento de nomes de usuários	307
Melhorando a performance do compartilhamento/servidor	308
Configuração de Clientes NetBEUI	310
Considerações sobre o Windows for Workgroups e LanManager	310
Configurando clientes em Grupo de Trabalho	310
Configurando clientes em Domínio	314
Erros conhecidos durante o logon do cliente	319
Programas de navegação gráficos	319
Cliente de configuração gráficos	319
Exemplos de configuração do servidor SAMBA	320
Grupo de Trabalho com acesso público	320
Grupo de Trabalho com acesso por usuário	321
Domínio	323
19. Restrições de acesso, recursos e serviços	326
Limitando recursos no bash	326
Uso do comando <code>readonly</code> para exportar variáveis	326
Restrições nos diretórios de usuários e root	326
Restrições básicas do shell <code>bash</code> com <code>bash -r/--restricted</code> , <code>rbash</code>	327
Finalizando consoles inativos	327
Desabilitando o registro de comandos digitados	328
Desabilitando serviços de shell para usuários	328
Limitação de recursos usando PAM	328
Descobrir se um determinado programa tem suporte a PAM	329
Definindo uma política padrão restritiva	329
Restringindo/Bloqueando o login	330
Restringindo o acesso a root no <code>su</code>	330
Restrições de serviços PAM baseados em dia/hora	331
Permitindo acesso a grupos extras	332
Limitação de recursos do shell	333
Restrições de acesso a programas/diretórios/arquivos usando grupos	334
Dando poderes de root para executar determinados programas	335
Restringindo o comando su	337
Restrições baseadas em usuário/IP	337
Restrições por MAC Address/IP	338
Desabilitando serviços não usados no <code>Inetd</code>	339
Evitando o uso de <code>hosts.equiv</code> e <code>.rhosts</code>	339
Restringindo o uso do <code>shutdown</code>	339
Restringindo o acesso ao sistema de arquivos <code>/proc</code>	340
Limitando o uso de espaço em disco (quotas)	341
Instalando o sistema de quotas	341
Editando quotas de usuários/grupos	343
Modificando a quota de todos os usuários de uma vez	345
Verificando a quota disponível ao usuário	345
Verificando a quota de todos os usuários/grupos do sistema	346
Avisando usuários sobre o estouro de quota	347
Suporte a senhas ocultas	348
Suporte a senhas <code>md5</code>	348
Restrições no hardware do sistema	349

BIOS do sistema	349
Retirada da unidade de disquetes	349
Placas de rede com eprom de boot	349
Protegendo o LILO	349
Disco rígido	349
20. Introdução ao uso de criptografia para transmissão/armazenamento de dados	351
Introdução	351
Sniffer	351
Detectando a presença de sniffers	352
Alternativas seguras a serviços sem criptografia	352
http	352
Transmissão segura de e-mails	353
Servidor pop3	353
Transferência de arquivos	353
login remoto	353
Bate papo via IRC	353
Transmissão de mensagens via ICQ	354
Sistemas de arquivos criptográfico	354
Usando pgp (gpg) para criptografia de arquivos	356
Instalando o PGP	356
Criando um par de chaves pública/privada	356
Encriptando dados	357
Decriptando dados com o gpg	357
Assinando arquivos	357
Checando assinaturas	358
Extraindo sua chave pública do chaveiro	358
Adicionando chaves públicas ao seu chaveiro pessoal	358
Listando chaves de seu chaveiro	358
Apagando chaves de seu chaveiro	359
Mudando sua FraseSenha	359
Assinando uma chave digital	359
Listando assinaturas digitais	360
Recomendações para a assinatura de chaves gpg	360
21. Como obter ajuda	365
Páginas de Manual	365
Info Pages	366
Ajuda na própria linha de comandos	367
help	367
apropos	367
whatis	367
locate	368
which	368
Documentos HOWTO's	368
Documentação de Programas	368
FAQ	368
RFC's	369
Internet	372
Páginas Internet de Referência	372
Listas de discussão	373
Netiqueta	375
Recomendações Gerais sobre a Comunicação Eletrônica	375
Email	376
Telegram/Whatsapp/Messenger/Gtalk/Skype	376
Talk	377

Listas de Discussão via Email	377
22. Apêndice	380
Sobre este guia	380
Sobre o Autor	380
Referências de auxílio ao desenvolvimento do guia	381
Onde encontrar a versão mais nova do guia?	382
Colaboradores do Guia	382
Marcas Registradas	383
Futuras versões	383
Guia do Linux	383
Chave Pública PGP	383

Capítulo 1. Introdução

Bem vindo ao Guia *Foca Linux*. O nome *FOCA* significa *FO*nte de *C*onsulta e *A*prendizado. Este guia está dividido em 3 níveis de aprendizado e versão que está lendo agora contém os níveis:

- Avançado

Entre o conteúdo do guia, você encontrará:

- Análise de logs do sistema GNU/Linux e aplicação para a solução de problemas (Capítulo 6, *Arquivos e daemons de Log*).
- Gerenciamento de contas de usuários, definição de período automático para troca de senha periódica, políticas de segurança, etc (Capítulo 11, *Gerenciamento de contas e cuidados para a proteção de senhas*).
- Principais tipos de ataques para descoberta de senhas e alguns métodos de como evita-las (“Senhas fáceis de adivinhar e escolha de boas senhas”).
- Integrar máquinas **Windows** e **Linux** em uma mesma rede compartilhando dados entre si e impressão (???)
- Sistemas de proteção de senhas do sistema (“Melhorando a segurança das senhas armazenadas em seu sistema”).
- Criptografia e segurança na transmissão de dados, usando exemplos práticos do uso de sniffers para entender o porque da uso de criptografia para transmissão segura de dados (Capítulo 20, *Introdução ao uso de criptografia para transmissão/armazenamento de dados*).
- Uso de serviços alternativos criptográficos (“Alternativas seguras a serviços sem criptografia”).
- Criptografia usando **gnupgp** (“Usando **pgp (gpg)** para criptografia de arquivos”).
- Uso de sistema de arquivos criptográficos para armazenamento de dados (“Sistemas de arquivos criptográfico”).
- Otimização de performance na transferência de dados do disco rígido através de particionamento e **hdparm**, uso de spindles para criação de swap (“Melhoria de performance”).
- O que são descargas estáticas e a importância do aterramento da instalação elétrica do computador (dirigido a usuários domésticos e de pequenas instalações) (“Descargas estáticas”).
- Maiores considerações a segurança de sistema e a problemas de segurança relativos a falhas de configuração (distribuída entre os capítulos de daemons e servidores).
- Montagem de um servidor de publicação Web usando o **Apache** (???)
- Montagem de um firewall avançado para proteção do sistema (filtragem de pacotes) usando o **iptables**, redirecionamento de pacotes, nat, bloqueio de tráfego P2P, masquerading, balanceamento de carga, marcação de pacotes, log, proteção contra port scans (Capítulo 10, *Firewall iptables*).
- Servidor de acesso para permitir o acesso a distância ao seu computador usando o **telnetd** (???)
- Servidor de acesso para permitir o acesso a distância a seu computador com criptografia usando o **ssh** (???)

- Servidor de identificação usando o **oidentd** (??).
- Montagem de um servidor pop3 para que suas estações de rede possam acessar o email na máquina servidor **Linux** usando programas como **Outlook**, **Communicator**, **Mutt**, **sylpheed** e outros que utilizem o protocolo **pop3** (??).
- Restrições de acesso a instalação do computador, acesso a grupos do sistema, restrições de login usando **PAM** (Capítulo 19, *Restrições de acesso, recursos e serviços*).
- Restrições de espaço usado em disco por usuários/grupos usando o sistema de quotas (“Limitando o uso de espaço em disco (quotas)”).
- Uso de grupos dos sistema para restrições de acesso (“Restrições de acesso a programas/diretórios/arquivos usando grupos”).
- Restrições de acesso via hardware: BIOS, disquete, placa com boot via rede, LILO, disco rígido (“Restrições no hardware do sistema”).
- Manipulações de variáveis no bash (*TMOUT*, *PS1*, *PS2*, *PS3*, *PS4*, *HISTORY*, etc).
- Montagem de shell básico restrito (“Restrições básicas do shell bash com bash -r/--restricted, rbash”).
- Uso do **sudo** para dar privilégio de execução de programas como root a determinados usuários (“Dando poderes de root para executar determinados programas”).

Para melhor organização, dividi o guia em 3 versões: *Iniciante*, *Intermediário* e *Avançado*. Sendo que a versão *Iniciante* é voltada para o usuário que não tem nenhuma experiência no **GNU/Linux**. A última versão deste guia pode ser encontrada em: Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>].

Caso tiver alguma sugestão, correção, crítica para a melhoria deste guia, preencha o formulário de sugestões disponíveis na página oficial do guia ou envie um e-mail para gleydson@guiafoca.org.

O *Foca GNU/Linux* é atualizado freqüentemente, por este motivo recomendo que preencha a ficha do aviso de atualizações na página web em Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>] no fim da página principal. Após preencher a ficha do aviso de atualizações, você receberá um e-mail sobre o lançamento de novas versões do guia e o que foi modificado, desta forma você poderá decidir em copia-la caso a nova versão contenha modificações que considera importantes.

Tenho recebido elogios de pessoas do Brasil (e também de outros países) elogiando o trabalho e a qualidade da documentação. Agradeço a todos pelo apoio, tenham certeza que este trabalho é desenvolvido pensando em repassar um pouco do conhecimento que adquiri ao começar o uso do Linux.

Também recebo e-mails de pessoas comemorando a aprovação na prova LPI nível 1, 2 e 3 após estudar usando o guia Foca GNU/Linux. Fico bastante feliz por saber disso, pois nunca tive a intenção de tornar o guia uma referência livre para estudo da LPI e hoje é usado para estudo desta difícil certificação que aborda comandos, serviços, configurações, segurança, empacotamento, criptografia, etc.

Considerações sobre o nível Avançado

Este guia foi compilado incluindo o nível *Avançado* do guia FOCA GNU/Linux, ele não tem a intenção de ser a única referência na configuração de serviços, servidores, aplicativos, nem garantia que ele atenderá a determinada finalidade específica do usuário (principalmente de uma rede, que depende de uma perfeita compreensão para adaptação de acordo com os requisitos de uma instalação local). Seu foco principal é a instalação do serviço, abordando considerações voltadas a segurança, e exemplos de configuração e seu funcionamento.

Com relação a capítulos sobre servidores, é importante observar qual versão é documentada no guia e se confere com a instalada em seu sistema, a fim de que tudo funcione corretamente. Entretanto, na maioria dos casos, as explicações relacionadas a uma versão de um programa são válidas em uma nova versão.

Pré-requisitos para a utilização deste guia

É assumido que você já tenha experiência na configuração de sistemas **Linux**, conheça boa parte dos comandos e sua utilização, tenha noções de rede e saiba como procurar documentação para complementar o que vem aprendendo. Enfim, requer que se tiver interesse em se aprofundar em determinada área, que utilize os métodos de busca de documentação sugeridos no guia para complementação do aprendizado. O guia não contém todos os materiais para que a pessoa se torne um *expert* no assunto, mas contém as referências para documentações mais específicas sobre determinadas áreas do sistema.

Este guia não cobre a instalação do sistema. Para detalhes sobre instalação, consulte a documentação que acompanha sua distribuição **GNU/Linux**.

O Linux

O **Linux** é um sistema operacional criado em 1991 por *Linus Torvalds* na universidade de Helsinki na Finlândia. É um sistema Operacional de código aberto distribuído gratuitamente pela Internet. Seu código fonte é liberado como *Free Software* (software livre), sob licença GPL, o aviso de copyright do kernel feito por Linus descreve detalhadamente isto e mesmo ele não pode fechar o sistema para que seja usado apenas comercialmente.

Isto quer dizer que você não precisa pagar nada para usar o Linux, e não é crime fazer cópias para instalar em outros computadores, nós inclusive incentivamos você a fazer isto. Ser um sistema de código aberto pode explicar a performance, estabilidade e velocidade em que novos recursos são adicionados ao sistema.

O requisito mínimo para rodar o **Linux** depende do kernel que será usado:

- 2.2.x - Computador 386 SX com 2 MB de memória
- 2.4.x - Computador 386 SX com 4MB de memória
- 2.6.x - Computador 486 DX com no mínimo 8MB
- 3.x.x - Computador 586 com no mínimo 16MB
- 4.x.x - Computador 586 com no mínimo 32MB
- 5.x.x - Computador 686 com no mínimo 32MB

Para espaço em disco é requerido 900MB para uma instalação básica usando modo texto com suporte a rede. Claro que não é considerada a execução de ambiente gráfico ou serviços de rede em produção, que neste caso é exigido mais memória RAM e espaço em disco para armazenamento de dados de programas e usuários.

O sistema segue o padrão *POSIX* que é o mesmo usado por sistemas *UNIX* e suas variantes. Assim, aprendendo o **Linux** você não encontrará muita dificuldade em operar um sistema do tipo **UNIX**, **FreeBSD**, **HPUX**, **SunOS**, etc., bastando apenas aprender alguns detalhes encontrados em cada sistema.

O código fonte aberto permite que qualquer pessoa veja como o sistema funciona (útil para aprendizado), corrigir algum problema ou fazer alguma sugestão sobre sua melhoria, esse é um dos motivos de seu rápido crescimento, do aumento da compatibilidade de periféricos (como novas placas sendo suportadas logo após seu lançamento) e de sua estabilidade.

Outro ponto em que ele se destaca é o suporte que oferece a placas, CD/DVD-RWs, BluRay e outros tipos de dispositivos de última geração e mais antigos (a maioria deles já ultrapassados e sendo completamente suportados pelo sistema operacional). Este é um ponto forte para empresas que desejam manter seus micros em funcionamento e pretendem investir em avanços tecnológicos com as máquinas que possui.

O **Linux** é desenvolvido por milhares de pessoas espalhadas pelo mundo, cada uma fazendo sua contribuição ou mantendo alguma parte do kernel gratuitamente. *Linus Torvalds* ainda trabalha em seu desenvolvimento e na coordenação dos grupos de trabalho do kernel.

O suporte ao sistema também se destaca como sendo o mais eficiente e rápido do que qualquer programa comercial disponível no mercado. Existem milhares de consultores e empresas especializadas no suporte e treinamento espalhados ao redor do mundo. Outra opção de suporte é através da comunidade Linux; você pode se inscrever em uma lista de discussão e relatar sua dúvida ou alguma falha, e sua mensagem será vista por centenas de usuários na Internet e algum irá te ajudar ou avisará as pessoas responsáveis sobre a falha encontrada para devida correção.

Algumas Características do Linux

- É livre e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional.

Muitos deles estavam cansados do excesso de propaganda (Marketing) e baixa qualidade de sistemas comerciais existentes

- Também recebe apoio de grandes empresas como IBM, Sun, RedHat, Intel, HP, etc. para seu desenvolvimento
- Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o **Windows, OS/2**) no mesmo computador.
- Multitarefa real
- Multiusuário
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres)
- Conectividade com outros tipos de plataformas como *Apple, Sun, Macintosh, Sparc, Alpha, PowerPc, ARM, Unix, Windows, DOS, etc.*
- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
- Proteção entre processos executados na memória RAM
- Suporte a mais de 256 terminais virtuais (consoles)
- Modularização - O **Linux** somente carrega para a memória o que é usado durante o processamento, liberando totalmente a memória assim que o programa/dispositivo é finalizado
- Devido a modularização, os drivers dos periféricos e recursos do sistema podem ser carregados e removidos completamente da memória RAM a qualquer momento. Os drivers (módulos) ocupam pouco espaço quando carregados na memória RAM (cerca de 6Kb para a Placa de rede NE 2000, por exemplo)
- Suporte nativo a rede e tecnologias avançadas como: balanceamento de carga, ips alias, failover, vlans, bridge, trunking, OSPF, BGP, MPLS.

- Não há a necessidade de se reiniciar o sistema após a modificar a configuração de qualquer periférico ou parâmetros de rede. Somente é necessário reiniciar o sistema no caso de uma instalação interna de um novo periférico, falha em algum hardware (queima do processador, placa mãe, etc.).
- Excepcional em escalabilidade desde computadores extremamente simples, dispositivos móveis (sistema Android utiliza kernel Linux), Raspberry PI, sistemas embarcados, geladeiras inteligentes, carros com centrais inteligentes, etc. até sistemas de clusters em nuvem gigantescos (como Amazon, Digital Ocean, entre maiores datacenters utilizados em núvens no mundo).
- Suporte nativo a múltiplas CPUs e multi threads, assim processadores como Dual Core, Core Duo, Athlon Duo, Quad Core, XEON, i3-i9 tem seu poder de processamento integralmente aproveitado, tanto em 32 ou 64 bits.
- Suporte nativo a dispositivos SSD, SATA, PATA, Fiber Channel
- Suporte nativo a virtualização, onde o **Linux** se destaca como plataforma preferida para execução de múltiplos sistemas operacionais com performance e segurança. Nuvens como Amazon, Digital Ocean utilizam nativamente KVM para execução de plataformas, assim como o sistema CGROUPs para execução de containers.
- O crescimento e novas versões do sistema não provocam lentidão, pelo contrário, a cada nova versão os desenvolvedores procuram buscar maior compatibilidade, acrescentar recursos úteis e melhor desempenho do sistema (como o que aconteceu na passagem do kernel 2.0.x para 2.2.x, da 2.2.x para a 2.4.x, da 3 para a 4.x.x, da 4.x.x para a 5.x.x)
- O **GNU/Linux** é distribuído livremente e licenciado de acordo com os termos da GPLv2.
- Acessa corretamente discos formatados pelo **DOS, Windows, Novell, OS/2, NTFS, SunOS, Amiga, Atari, Mac**, etc.
- O **LINUX POSSUI MECANISMOS DE HARDENING AVANÇADOS CONTRA VÍRUS E MALWARES!** Devido a separação de privilégios entre processos e respeitadas as recomendações padrão de política de segurança e uso de contas privilegiadas (como a de root, como veremos adiante), programas como vírus tornam-se inúteis pois tem sua ação limitada pelas restrições de acesso do sistema de arquivos e execução.

Qualquer programa (nocivo ou não) poderá alterar partes do sistema que possui permissões (será abordado como alterar permissões e tornar seu sistema mais restrito no decorrer do guia). Frequentemente são criados exploits que tentam se aproveitar de falhas existentes em sistemas desatualizados e usa-las para causar danos. *Erroneamente* este tipo de ataque é classificado como vírus por pessoas mal informadas e são resolvidas com sistemas bem mantidos. Em geral, usando uma boa distribuição que tenha um eficiente sistema de atualização e bem configurado, você terá 99.9% de sua tranquilidade.

- Rede TCP/IP mais rápida que no Windows e tem sua pilha constantemente melhorada. O **GNU/Linux** tem suporte nativo a redes TCP/IP e não depende de uma camada intermediária como o WinSock. Em acessos via modem a Internet, a velocidade de transmissão é 10% maior.
- Executa outros sistemas operacionais como *Windows, MacOS, DOS* ou outro sistema **Linux** através de consagrados sistemas de virtualização como **KVM, Xen, vmware VirtualBox**, ou emulação como o **DOSEMU, QEMU, WINE**.
- Suporte completo e nativo a diversos dispositivos de comunicação via infravermelho, Bluetooth, Firewire, USB. Basta conectar e o seu dispositivo é automaticamente reconhecido. Raramente são necessários drivers externos, exceto no caso de dispositivos muito novos que não tenham o suporte ainda adicionado no sistema.

- Suporte a fiber channel.
- Suporte a rede via rádio amador.
- Suporte a dispositivos Plug-and-Play.
- Suporte nativo a pen drivers, dispositivos de armazenamento e cartões de memória.
- Suporte nativo a dispositivos I2C
- Integração com gerenciamento de energia ACPI e APM
- Dispositivos de rede Wireless. Tanto com criptografia WEB e WPA2/3 PSK
- Vários tipos de firewalls avançados de alta qualidade na detecção de tráfego indesejável, dando ao administrador uma excelente ferramenta de proteção e controle de sua rede.
- Roteamento estático e dinâmico de pacotes.
- Ponte entre Redes, proxy arp
- Proxy Tradicional e Transparente.
- Possui recursos para atender a mais de um endereço IP na mesma placa de rede, sendo muito útil para situações de manutenção em servidores de redes ou para a emulação de "múltiplos computadores".

O servidor WEB e FTP podem estar localizados no mesmo computador, mas o usuário que se conecta tem a impressão que a rede possui servidores diferentes.

- Os sistemas de arquivos usados pelo **GNU/Linux** (Ext2, Ext3, reiserfs, xfs, jfs) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.
- Permite a montagem de um servidor de publicação Web, E-mail, News, etc. com um baixo custo e alta performance. O melhor servidor Web do mercado, o **Apache**, é distribuído gratuitamente junto com a maioria das distribuições Linux. O mesmo acontece com o **Sendmail**.
- Por ser um sistema operacional de código aberto, você pode ver o que o código fonte (instruções digitadas pelo programador) faz e adapta-lo as suas necessidades ou de sua empresa. Esta característica é uma segurança a mais para empresas sérias e outros que não querem ter seus dados roubados (você não sabe o que um sistema sem código fonte faz na realidade enquanto esta processando o programa).
- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Pode ser executado em 16 arquiteturas diferentes (Intel, Macintosh, Alpha, Arm, etc.) e diversas outras sub-arquiteturas.
- Empresas especializadas e consultores especializados no suporte ao sistema espalhados por todo o mundo.
- Entre muitas outras características que você descobrirá durante o uso do sistema (além de poder criar outras, caso seja um administrador avançado ou desenvolvedor).

TODOS OS ÍTENS DESCRITOS ACIMA SÃO VERDADEIROS E TESTADOS PARA QUE TIVESSE PLENA CERTEZA DE SEU FUNCIONAMENTO.

Capítulo 2. Explicações Básicas

Este capítulo traz explicações sobre os principais componentes existentes no computador e do sistema operacional **Linux**.

Monitorando os logs

Os arquivos e diretórios de logs residem em `/var/log` e registram tudo o que acontecem com o kernel, com os daemons e utilitários do sistema. Eles são muito importantes tanto para monitorar o que acontece com o seu sistema como para ajudar na solução de problemas diversos. É comum programas como o servidor web, e-mail, mensagens instantaneas, firewall, irc, banco de dados, gravarem os arquivos de log em diretórios próprios dentro de `/var/log/programa`, desta forma evitam misturar seus arquivos com os de log do sistema residentes em `/var/log`.

Acostume-se a olhar constantemente os arquivos de log em seu sistema, isto pode ser importante para encontrar possíveis falhas de segurança, tentativa de acesso ao sistema e, principalmente, solucionar problemas (principalmente os mais complicados). Leia Capítulo 6, *Arquivos e daemons de Log* para mais detalhes.

Destruindo arquivos/partições de forma segura

Esta seção tem a intenção de conscientizar o administrador do uso devido de técnicas para garantir que dados sensíveis sejam apagados de forma um pouco mais segura em seu sistema.

Quando um arquivo é apagado, apenas a entrada na tabela de inodes é mexida, e ele pode ainda ser recuperado com o **debugfs** e um pouco de paciência e engenharia. O mesmo acontece com as partições, que podem ser recuperadas com facilidade (isto é explicado no nível Intermediário do guia). Esta recuperação é proporcionada pelas regras de funcionamento do sistema de arquivos e do esquema de particionamento, ou seja, são permitidas pelo SO.

Vou um pouco mais além: O disco rígido é uma mídia magnética e opera de forma mecânica para ler/gravar dados. Quando um arquivo é apagado, seja por qualquer motivo, ainda é possível recupera-lo. O que permite isto é porque o HD nem sempre tem a precisão de gravar **exatamente** no mesmo lugar (pois a cabeça é movida mecanicamente), gravando em trilhas microscópicamente vizinhas a anterior. Então a imagem do arquivo que foi apagada continua lá. Segundo ouvi falar, a NASA possui recursos para recuperar até 60 regravações posteriores no disco. É claro que isto pode ocorrer em pouco tempo, dependendo do tamanho de sua partição e se esta for uma `/var/spool` em um servidor de e-mails :-)

Baseado nesta teoria, você poderá apagar os dados de forma destrutiva usando o programa **shred**, que regrava o arquivo repetidamente com dados aleatórios. Sua sintaxe é a seguinte:

```
shred -n 70 -v -u arquivo
```

Isto faz com que ele regrava o conteúdo do arquivo 70 vezes com dados aleatórios. O `-u` trunca e remove o arquivo após concluído.

Note que o uso de dados aleatórios serve para destruir as possibilidades de uma recuperação simples, este é o motivo de se recomendar sempre o uso de `/dev/urandom` ao invés de `/dev/zero` para destruição de arquivos.

OBS1: Saiba exatamente o que está fazendo pois estes procedimentos servem para dificultar ao máximo a recuperação de dados.

OBS2: Devido as tecnologias de sistemas que utilizam journaling (*XFS, EXT3, EXT4, JFS e ReiserFS*) e sistemas RAID, o **shred** não funcionará. O **shred** também não funcionará com sistemas de arquivos via rede (*NFS, SMB, etc.*). Se procura alguma forma de proteger seus dados, mesmo que apagados, utilize um método de criptografia como o *DM-CRYPTO, crypto-loop, gpg, etc.*

OBS3: Caso esteja usando um sistema de arquivos criptografado, estes procedimentos são quase desnecessários (dependendo do nível de segurança e algoritmos que você utiliza).

Capítulo 3. Hardware

Hardware é tudo que diz respeito a parte física do computador. Nesta seção serão abordados assuntos relacionados com a configuração de hardwares, escolha de bons hardwares, dispositivos for Windows, etc.

Placa de expansão

É um circuito eletrônico encaixado na placa mãe que tem por objetivo adicionar novas funcionalidades ao computador. Esta placa pode ser uma:

- placa de som - para fazer o computador emitir sons, músicas, ligar um joystick, etc.
- Placa de vídeo 3D - Para obter imagens mais rápidas para jogos e ambientes de desktop 3 dimensões
- Placa de captura - Para assistir televisão/rádio e gravar a programação de TV em seu micro.
- fax-modem - para enviar/receber fax, conectar-se a internet, acesso remoto, bina, etc.
- rede - para permitir a comunicação com outros computadores em uma rede interna
- controladora de periféricos - Para ligar discos rígidos, unidades de disquete, impressora, mouse, joystick, etc.
- SCSI - Para ligar unidades de disco rígidos e periféricos de alto desempenho.
- Controladora de Scanner - Para ligar um Scanner externo ao micro computador.

O encaixe da placa mãe que recebe as placas de expansão são chamados de *Slots*.

Nomes de dispositivos

Seria terrível se ao configurar CADA programa que utilize o mouse ou o modem precisássemos nos referir a ele pela IRQ, I/O, etc... para evitar isso são usados os *nomes de dispositivos*.

Os *nomes de dispositivos* no sistema **GNU/Linux** são acessados através do diretório `/dev`. Após configurar corretamente o modem, com sua porta I/O 0x2F8 e IRQ 3, ele é identificado automaticamente por `/dev/ttyS1` (equivalente a COM2 no **DOS**). Daqui para frente basta se referir a `/dev/ttyS1` para fazer alguma coisa com o modem.

Você também pode fazer um link de `/dev/ttyS1` para um arquivo chamado `/dev/modem` usando:
`ln -s /dev/ttyS1 /dev/modem`, faça a configuração dos seus programas usando `/dev/modem` ao invés de `/dev/ttyS1` e se precisar reconfigurar o seu modem e a porta serial mudar para `/dev/ttyS3`, será necessário somente apagar o link `/dev/modem` antigo e criar um novo apontando para a porta serial `/dev/ttyS3`.

Não será necessário reconfigurar os programas que usam o modem pois eles estão usando `/dev/modem` que está apontando para a localização correta. Isto é muito útil para um bom gerenciamento do sistema.

Abaixo uma tabela com o nome do dispositivo no **GNU/Linux**, portas I/O, IRQ, DMA e nome do dispositivo no **DOS** (os nomes de dispositivos estão localizados no diretório `/dev`):

Dispos. Linux	Dispos. DOS	IRQ	DMA	I/O
ttyS0	COM1	4	-	0x3F8
ttyS1	COM2	3	-	0x2F8
ttyS2	COM3	4	-	0x3E8
ttyS3	COM4	3	-	0x2E8
lp0	LPT1	7	3 (ECP)	0x378
lp1	LPT2	5	3 (ECP)	0x278
/dev/hda1	C:	14	-	0x1F0, 0x3F6
/dev/hda2	D: *	14	-	0x1F0, 0x3F6
/dev/hdb1	D: *	15	-	0x170, 0x376

* A designação de letras de unidade do **DOS** não segue o padrão do **GNU/Linux** e depende da existência de outras unidades físicas/lógicas no computador.

Configuração de Hardware

A configuração consiste em ajustar as opções de funcionamento dos dispositivos (periféricos) para comunicação com a placa mãe bem como a configuração do software correspondente para fazer acesso ao hardware. Um sistema bem configurado consiste em cada dispositivo funcionando com suas portas I/O, IRQ, DMA bem definidas, não existindo conflitos com outros dispositivos. Isto também permitirá a adição de novos dispositivos ao sistema sem problemas.

Dispositivos PCI, PCI Express, AMR, CNR possuem configuração automática de recursos de hardware, podendo apenas ser ligados na máquina para serem reconhecidos pela placa mãe. Após isso deverá ser feita a configuração do módulo do kernel para que o hardware funcione corretamente.

Os parâmetros dos módulos do kernel usados para configurar dispositivos de hardware são a *IRQ*, *DMA* e *I/O*. Para dispositivos plug and play, como hardwares PCI, basta carregar o módulo para ter o hardware funcionando.

IRQ - Requisição de Interrupção

Existem dois tipos básicos de interrupções: as usadas por dispositivos (para a comunicação com a placa mãe) e programas (para obter a atenção do processador). As *interrupções de software* são mais usadas por programas, incluindo o sistema operacional e *interrupções de hardware* mais usado por periféricos. Daqui para frente será explicado somente detalhes sobre interrupções de hardware.

Os antigos computadores 8086/8088 (XT) usavam somente 8 interrupções de hardware operando a 8 bits. Com o surgimento do AT foram incluídas 8 novas interrupções, operando a 16 bits. Os computadores 286 e superiores tem 16 interrupções de hardware numeradas de 0 a 15. No kernel 2.4 e superiores do Linux, a função APIC (*Advanced Programmable Interruption Controller*) permite gerenciar de forma avançada mais de 15 interrupções no sistema operacional. Estas interrupções oferecem ao dispositivo associado a capacidade de interromper o que o processador estiver fazendo, pedindo atenção imediata.

As interrupções do sistema podem ser visualizadas no kernel com o comando `cat /proc/interrupts`. Abaixo um resumo do uso mais comum das 16 interrupções de hardware:

```
0      Timer do Sistema - Fixa
01     Teclado - Fixa
```

- 02 Controlador de Interrupção Programável - Fixa.
Esta interrupção é usada como ponte para a IRQ 9 e vem dos antigos processadores 8086/8088 que somente tinham 8 IRQs. Assim, para tornar processadores 8088 e 80286 comunicáveis, a IRQ 2 é usada como um redirecionador quando se utiliza uma interrupção acima da 8.
- 03 Normalmente usado por /dev/ttyS1 mas seu uso depende dos dispositivos instalados em seu sistema (como fax-modem, placas de rede 8 bits, etc).
- 04 Normalmente usado por /dev/ttyS0 e quase sempre usada pelo mouse serial a não ser que um mouse PS2 esteja instalado no sistema.
- 05 Normalmente a segunda porta paralela. Muitos micros não tem a segunda porta paralela, assim é comum encontrar placas de som e outros dispositivos usando esta IRQ.
- 06 Controlador de Disquete - Esta interrupção pode ser compartilhada com placas aceleradoras de disquete usadas em tapes (unidades de fita).
- 07 Primeira porta de impressora. Pessoas tiveram sucesso compartilhando esta porta de impressora com a segunda porta de impressora. Muitas impressoras não usam IRQs.
- 08 Relógio em tempo real do CMOS - Não pode ser usado por nenhum outro dispositivo.
- 09 Esta é uma ponte para IRQ2 e deve ser a última IRQ a ser utilizada. No entanto pode ser usada por dispositivos.
- 10 Interrupção normalmente livre para dispositivos. O controlador USB utiliza essa interrupção quando presente, mas não é regra.
- 11 Interrupção livre para dispositivos
- 12 Interrupção normalmente livre para dispositivos. O mouse PS/2, quando presente, utiliza esta interrupção.
- 13 Processador de dados numéricos - Não pode ser usada ou compartilhada
- 14 Esta interrupção é usada pela primeira controladora de discos rígidos e não pode ser compartilhada.
- 15 Esta é a interrupção usada pela segunda controladora de discos e não pode ser compartilhada. Pode ser usada caso a segunda controladora esteja desativada.

Dispositivos ISA, VESA, EISA, SCSI não permitem o compartilhamento de uma mesma IRQ, talvez isto ainda seja possível caso não haja outras opções disponíveis e/ou os dois dispositivos não acessem a IRQ ao mesmo tempo, mas isto é uma solução precária.

Conflitos de IRQ ocorriam nesse tipo de hardware acima ocasionando a parada ou mal funcionamento de um dispositivo e/ou de todo o sistema. Para resolver um conflito de IRQs, deve-se conhecer quais

IRQs estão sendo usadas por quais dispositivos (usando `cat /proc/interrupts`) e configurar as interrupções de forma que uma não entre em conflito com outra. Isto normalmente é feito através dos jumpers de placas ou através de software (no caso de dispositivos jumperless ou plug-and-play).

Dispositivos PCI, PCI Express são projetados para permitir o compartilhamento de interrupções. Se for necessário usar uma interrupção normal, o chipset (ou BIOS) mapeará a interrupção para uma interrupção normal do sistema (normalmente usando alguma interrupção entre a IRQ 9 e IRQ 12) ou usando APIC (se estiver configurado).

Prioridade das Interrupções

Cada IRQ no sistema tem um número que identifica a prioridade que será atendida pelo processador. Nos antigos sistemas XT as prioridades eram identificadas em seqüência de acordo com as interrupções existentes:

```
IRQ 0 1 2 3 4 5 6 7 8
PRI 1 2 3 4 5 6 7 8 9
```

Com o surgimento do barramento AT (16 bits), as interrupções passaram a ser identificadas da seguinte forma:

```
IRQ 0 1 2 (9 10 11 12 13 14 15) 3 4 5 6 7 8
PRI 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

Note que a prioridade segue em seqüência através da ponte da IRQ 2 para IRQ 9. Os dispositivos com prioridade mais baixa são atendidos primeiro, mas é uma diferença de desempenho praticamente imperceptível de ser notada nos sistemas atuais.

DMA - Acesso Direto a Memória

A *DMA* é usada para permitir a transferência de dados entre dispositivos I/O e a memória sem precisar do processador para fazê-lo. Ele livra esta carga do processador e resulta em uma rápida transferência de dados.

O PC padrão tem dois controladores de DMA. O primeiro controla os canais 0, 1, 2, 3 e o segundo os canais 4, 5, 6, 7, assim temos 8 canais. No entanto, o canal 4 é perdido porque é usado pelo *controlador de acesso direto a memória*. Os canais 0-3 são chamados de canais baixos porque podem somente mover um byte (*8 bits*) por transferência enquanto canais altos movem 2 bytes (*16 bits*) por transferência.

Os dados movidos usando a DMA **não** são movidos através do controlador de DMA. Isto oferece uma limitação porque a DMA somente podem mover dados entre os dispositivos (portas I/O) e a memória. Não é possível mover dados entre as portas ou entre a memória.

Existem dois controladores de DMA nos computadores AT e superiores. Ao contrário do que acontece com os dois controladores de IRQ, o primeiro controlador é ligado ao segundo e não o segundo ao primeiro. Os canais de DMA altos (5 ao 7) somente podem ser acessados por dispositivos de 16 bits (aqueles que utilizam a segunda parte do slot AT). Como resultado temos 8 canais de DMA, de 0 a 7, sendo que a DMA 4 é usada como ligação entre eles.

Os canais de DMA em uso no sistema podem ser visualizados com `cat /proc/dma`. Abaixo uma listagem de uso mais comum dos canais de DMA.

DMA	Barram.	Uso
0	-	Usada pelo circuito de refresh da memória DRAM
1	8/16 bits	Normalmente usado por placas de som (canal 8 bits), porta paralela ECP, adaptadoras SCSI, placas de rede ou controladora de scanner.
2	8/16 bits	Normalmente usado pela controladora de disquetes ou controladoras de tapes.
3	8/6 bits	Usado pela porta paralela ECP, placa de som, controladoras de tapes, controladoras SCSI ou controladora de scanner antiga.
4	-	Usada como ponte para a outra controladora de DMA (0-3)
5	16 bits	Normalmente usada pela placa de som (canal 16 bits), placas controladoras SCSI, placas de rede ou controladora de scanner.
6	16 bits	Placa de som (canal 16 bits), controladora de scanner ou placa de rede.
7	16 bits	Placa de som (canal 16 bits), controladora de scanner ou placa de rede.

Somente dispositivos ISA e derivados dele, como o EISA e VESA, usam os canais de DMA padrão. Os atuais dispositivos de alta taxa de transferência (normalmente PCI) possuem seu próprio controlador de DMA embutido, muito mais rápido do que a DMA padrão. Este controlador de DMA é chamado de *Bus Mastering* e muito usado nos discos rígidos atuais e pode atingir taxas de 33,3MB/s (no modo 2) e 66MB/s (no modo 4 - requer um cabo IDE com aterramento para evitar interferências de ruídos externos).

Conflitos de DMA

Um canal de DMA não pode ser compartilhado entre dispositivos. Ainda é possível configurar dois dispositivos para usarem um mesmo canal de DMA, desde que ele não seja usado ao mesmo tempo. Isto acontece com Scanners paralelos que compartilham a mesma porta paralela com a impressora. Se você for uma pessoa que explora os recursos de multitarefa de seu Linux e seu desempenho, evite estes tipos de dispositivos, prefira aqueles que utilizam seus próprios recursos.

Quando ocorre um conflito de DMA, os dados podem ser misturados e ocorrerem coisas estranhas até o travamento total do sistema. Este tipo de conflito é difícil de se diagnosticar, a não ser que o técnico seja experiente o bastante e tenha desconfiado do que o problema se trata...

I/O - Porta de Entrada/Saída

Cada dispositivo possui um endereço de porta. O endereço é uma localização da memória usada pelo computador para enviar dados ao dispositivo e onde o dispositivo envia dados ao computador. Ao contrário da IRQ e DMA, o dispositivo pode usar mais de uma porta de Entrada/Saída ou uma faixa de endereços. Por exemplo, uma placa de som padrão usa as portas 0x220, 0x330 e 0x388, respectivamente audio digital, midi e opl3.

As placas de rede normalmente transferem grandes quantidades de dados, assim ocupam uma faixa de endereços. Uma NE2000, por exemplo, ocupa a faixa de endereços 0x260 a 0x27F (0x260-0x27F). O tamanho da faixa de endereços varia de acordo com o tipo de dispositivo.

Os endereços de I/O em uso no sistema podem ser visualizados com o comando `cat /proc/ioports`.

Endereços das portas de entrada/saída não podem ser compartilhados

Hardwares configuráveis por jumpers, dip-switches, jumperless e Plug-and-Play.

Jumpers

Hardwares configuráveis por *jumpers* (pinos metálicos protegidos por uma capa plástica) tem sua configuração alterada através da colocação, retirada ou mudança de posição física do pino. Este tipo de hardware, antigamente presente em placas ISA e VESA, não é mais usado atualmente devido a configuração Plug and Play de dispositivos PCI, PCI express, etc.

As disposição dos jumpers são normalmente definidas em *fechado/aberto* e *multi-posição*. Na disposição *fechado/aberto*, o jumper pode ou não ser colocado, definindo a configuração do dispositivo:

:: | ::

Esta disposição é facilmente encontrada na seleção de IRQ e I/O em placas de fax-modem.

Na disposição *multi-posição*, os pinos de encaixe são numerados de 1 a 3 (ou 1 a 4, 1 a 5, etc) e os pinos podem ou não ser colocados na placa e a posição que são colocados também influencia os valores escolhidos para o funcionamento do dispositivo (a posição 1-2 especificam um valor enquanto 2-3 especificam outro). A associação entre a posição dos jumpers e a configuração desejada é feita consultando o mapa desenhado no circuito impresso da placa ou o manual de instruções da placa.

A configuração de jumper através de multi-posição é normalmente usada em placas mãe para definir a *frequência de operação do barramento*, a *frequência de multiplicação* ou o *tipo do processador*.

Se não possuir o mapa de configuração de sua placa e/ou o manual de instruções, será necessário fazer um mapeamento manual da placa, mas para isto você precisará conhecer detalhadamente a configuração de portas I/O, DMA, IRQ usadas na máquina que será usada e anotar as diferenças obtidas através da modificação da pinagem do dispositivo. Isto não é fácil, mas técnicos de informática experientes conhecerão as *armadilhas* encontradas pelo mapeamento manual de placas e farão o esquema de configuração completo do dispositivo, obtendo um excelente manual de instruções. Nesta hora a experiência conta mais que o uso de programas de diagnóstico.

Outra característica de hardwares configurados através de jumpers é que raramente apresentam problemas de funcionamento, a não ser que seus parâmetros como IRQ, DMA, ou I/O estejam em conflitos com outro dispositivo, mas isso não é culpa do fabricante e nem mesmo do dispositivo...

Dip-Switches

É a mesma coisa que os hardwares configuráveis por jumpers exceto que são usados *dip-switches* no lugar de jumpers. O *dip-switches* é um conjunto de chaves numeradas que podem ser colocadas para cima ou para baixo (como um disjuntor ou vários interruptores *LIGA/DESLIGA* colocados um ao lado do outro) para se modificar a configuração do dispositivo.

Jumperless (sem jumper)

Os hardwares *jumperless* não possuem jumpers e são configurados através de um programa que acompanha a própria placa. Neste programa é escolhida a IRQ, DMA, I/O e a configuração é salva na própria placa ou restaurada após cada inicialização por um programa carregado na memória. Devido a configuração via

software, se obtém uma configuração fixa com muito mais facilidade do que via jumpers (por não haver a necessidade de se retirar a placa).

A maioria das placas jumperless podem funcionar também como Plug-and-Play. Existem muitas placas de rede, fax-modem, scanner jumperless no mercado.

Plug-and-Play

O *Plug-and-Play* é um protocolo que lê os valores de operação disponíveis para a placa e permitem que o usuário possa especificar facilmente qual será sua IRQ, DMA, I/O. Hardwares PCI possuem configuração Plug-and-Play nativa, registrando suas interrupções, portas e dma na tabela de hardwares PCI do sistema.

A diferença em relação ao modo jumperless é que toda a configuração do hardware (IRQ, DMA e I/O) é feita pelo kernel do **Linux**, onde ele passa a configuração detectada durante a inicialização do sistema para os módulos carregados, garantindo o perfeito funcionamento do dispositivos e evitando conflitos. Na época de hardwares ISA e VESA, o programa **isapnp** era a preferencia para a configuração de placas ISA Plug and Play.

Veja a próxima seção para entender como funciona o arquivo de configuração `isapnp.conf` e assim poder ativar seu dispositivo Plug-and-Play.

Listando as placas e outros hardwares em um computador

Administradores e técnicos ao configurar uma máquina precisarão saber quais os hardwares ela possui, periféricos e até mesmo a revisão de dispositivos e clock para configurar as coisas e ver a necessidade de atualizações de dispositivos atuais.

Dispositivos PCI/AMR/CNR podem ser listados executando o comando `cat /proc/pci`. Outra forma de listar tais dispositivos é usando o **lspci**, se você precisa de mais detalhes como o mapeamento de memória, use `lspci -vv`.

O mapeamento de memória de dispositivos podem ser mostrados com o comando `cat /proc/ioproports`, ou usando o comando **lsdev**.

O barramento USB e dispositivos conectados a ele podem ser listados com o comando **lsusb** ou com `cat /proc/bus/usb/devices`.

Hardwares disponíveis na máquina, como placa mãe, clock multiplicador, discos, placas diversas, versões e números seriais de dispositivos podem ser mostrados através do comando **lshw**. Use `lshw -html` para produzir a listagem em formato HTML, bem interessante para relatórios :-)

Conflitos de hardware

Ocorre quando um ou mais dispositivos usam a mesma *IRQ*, *I/O* ou *DMA*. Um sistema com configurações de hardware em conflito tem seu funcionamento instável, travamentos constantes, mal funcionamento de um ou mais dispositivos e até mesmo, em casos mais graves, a perda de dados. Conflitos geralmente ocorriam em placas ISA, VESA onde era necessário conhecer e usar uma tabela de valores padrões para a configuração de periféricos (como a mostrada no início desse capítulo).

Para resolver conflitos de hardware é necessário conhecer a configuração de cada dispositivo em seu sistema. Os comandos `cat /proc/interrupts`, `cat /proc/dma` e `cat /proc/ioproports` podem ser úteis para se verificar as configurações usadas.

Barramento

O tipo de *slot* varia de acordo com o barramento usado no sistema, que pode ser um(s) do(s) seguinte(s):

ISA 8 Bits	Industry Standard Architecture - É o padrão mais antigo, encontrado em computadores PC/XT.
ISA 16 Bits	Evolução do padrão ISA 8 Bits, possui um conector maior e permite a conexão de placas de 8 bits. Sua taxa de transferência chega a 2MB/s.
VESA	Video Electronics Standard Association - É uma interface feita inicialmente para placas de vídeo rápidas. O barramento VESA é basicamente um ISA com um encaixe extra no final. Sua taxa de transferência pode chegar a 132MB/s.
EISA	Enhanced Industry Standard Architecture - É um barramento mais encontrado em servidores. Tem a capacidade de bus mastering, que possibilita a comunicação das placas sem a interferência da CPU.
MCA	Micro Channel Architecture - Barramento 32 bits proprietário da IBM. Você não pode usar placas ISA nele, possui a característica de bus mastering, mas pode procurar por dispositivos conectados a ele, procurando configuração automática.
	Este barramento estava presente no PS/1 e PS/2, hoje não é mais usado.
PCI	Peripheral Component Interconnect - É outro barramento rápido produzido pela Intel com a mesma velocidade que o VESA. O barramento possui um chipset de controle que faz a comunicação entre os slots PCI e o processador. O barramento se configura automaticamente (através do Plug-and-Play). O PCI é o barramento mais usado por Pentiums e está se tornando um padrão no PC.
PCI Express	Peripheral Component Interconnect Express - Idêntico ao barramento PCI, funcionando nativamente no clock de 64 bits.
AGP	Accelerated Graphics Port - É um novo barramento criado exclusivamente para a ligação de placas de vídeo. É um slot marrom (em sua maioria) que fica mais separado do ponto de fixação das placas no chassis (comparado ao PCI). Estas placas permitem obter um desempenho elevado de vídeo se comparado as placas onboards com memória compartilhada e mesmo PCI externas. O consumo de potência em placas AGP x4 podem chegar até a 100W, portanto é importante dimensionar bem o sistema e ter certeza que a fonte de alimentação pode trabalhar com folga.
PCMCIA	Personal Computer Memory Card International Association - É um slot especial usado para conexões de placas externas (normalmente revestidas de plástico) e chamadas de <i>cartões PCMCIA</i> . Estes cartões podem adicionar mais memória ao sistema, conter um fax-modem, placa de rede, disco rígido, etc.

Os cartões PCMCIA são divididos em 3 tipos:

- Tipo 1 Tem a espessura de 3.3 milímetros, e podem conter mais memória RAM ou memória Flash.
- Tipo 2 Tem a espessura de 5 milímetros e capacidade de operações I/O. É um tipo usado para placas de fax-modem, rede, som. Computadores que aceitam cartões PCMCIA do tipo 2, mantêm a compatibilidade com o tipo 1.

Tipo 3 Tem a espessura de 10.5 milímetros e normalmente usado para discos rígidos PCMCIA. Slots PCMCIA do tipo 3 mantém a compatibilidade com o tipo 2 e 1.

AMR Audio Modem Raise - Pequeno barramento criado pela Intel para a conexão de placas de som e modem. Placas de som e modem AMR usam o HSP (host signal processor) e são como as Placas on-board e todo o processamento é feito pela CPU do computador (veja detalhes em “Placas on-board / off-board” e “Hardwares específicos ou “For Windows””).

Sua vantagem é o preço: um modem ou placa de som AMR custa em torno de R\$ 25,00.

CNR Communication and Networking Rise - Pequeno barramento criado pela Intel para a conexão de placas de som, modems e placas de rede. Este é um pequenino slot marrom que é localizado no ponto de fixação das placas no chassi do gabinete. Elas são como as Placas on-board e todo o processamento é feito pela CPU do computador (veja detalhes em “Placas on-board / off-board” e “Hardwares específicos ou “For Windows””).

Placas on-board / off-board

Placas *on-board* são embutidas na placa mãe (*motherboard*). Placas *off-board* são placas externas encaixadas nos slots de expansão da placa mãe.

No início da era do PC/XT todas as placas eram embutidas na placa mãe (na época eram somente a placa de vídeo e controladora). Com o surgimento do padrão AT, diversas empresas de informática desenvolveram dispositivos concorrentes e assim o usuário tinha a liberdade de escolha de qual dispositivo colocar em sua placa mãe (ou o mais barato ou o de melhor qualidade e desempenho), isto permitiu a adição de periféricos de qualidade sem romper com seu orçamento pessoal (comprando uma placa de som, depois uma de fax-modem, placa de vídeo melhor, etc).

Atualmente parece que voltamos ao ponto de partida e tudo vem embutido na placa mãe (*on-board*) e o usuário não tem como escolher qual dispositivo usar em seu computador. É muito difícil (praticamente impossível) encontrar uma placa mãe que satisfaça completamente as necessidades do usuário ou recomendações de um bom técnico de informática (a não ser que seja um técnico experiente e encontre alguma alternativa).

Certamente o único dispositivo que funciona melhor se embutido na placa mãe é a *placa controladora de periféricos*. Esta placa é usada para se conectar unidades de disquete, discos rígidos, CD-ROM, portas seriais, paralelas, joystick ao computador. Os HDs conectados em uma controladora embutida conseguem ter um desempenho muito maior do que em placas conectadas externamente, sem causar nenhum tipo de problema.

Hardwares embutidos na placa mãe (como fax-modem, vídeo, som) são em média 30% mais baratos que os vendidos separadamente mas quase sempre são usados dispositivos de baixo desempenho e qualidade para reduzir o preço da placa mãe e quase sempre usados hardwares *For Windows*.

Hoje em dia por causa do preço da placa mãe, é comum encontrar pessoas que verificam somente o preço e sequer procuram saber ou conhecem a qualidade das placas embutidas na placa mãe. Pior ainda é encontrar vendedores despreparados que sequer sabem explicar o porque que uma placa de som Sound Blaster 128 é mais cara que uma de modelo genérico...

Geralmente dispositivos on-board trazem problemas caso tal dispositivo queime e geralmente é colocado um hardware de baixa qualidade para baratear o custo de placas mãe, que na maioria das vezes também oferece grande dificuldade para ser configurada no **Linux**.

Outro periférico que traz problemas e carga para o processador é o fax-modem for Windows, HSP, AMR, micromodem, etc. utilizando o processador do sistema para realizar seu trabalho e algumas vezes não trazem nem mesmo o chip UART. Isso resulta em perda de qualidade na conexão e maior consumo telefônico.

Se você estiver em uma situação destas, certamente os computadores de menor potência e com hardwares inteligentes (que possuem seus próprios chips de controle e processamento) não terão o desempenho comprometido. O preço pode ser maior mas você estará pagando por um dispositivo de melhor qualidade e que certamente trará benefícios a você e ao seu sistema.

Consulte um técnico em informática experiente para te indicar uma placa mãe de bom preço e de qualidade. É muito comum encontrar falta de profissionalismo em pessoas que não sabem distinguir as características, funções e vantagens entre uma placa de boa qualidade e um hardware for Windows a não ser o preço mais barato.

Hardwares específicos ou "For Windows"

Esta seção foi retirada do manual de instalação da Debian GNU/Linux. Uma tendência que perturba é a proliferação de Modems e impressoras específicos para Windows. Em muitos casos estes são especialmente fabricados para operar com o Sistema Operacional Microsoft Windows e costumam ter a legenda WinModem, for Windows, ou Feito especialmente para computadores baseados no Windows.

Geralmente estes dispositivos são feitos retirando os processadores embutidos daquele hardware e o trabalho deles são feitos por drivers do Windows que são executados pelo processador principal do computador. Esta estratégia torna o hardware menos caro, mas o que é poupado não é passado para o usuário e este hardware pode até mesmo ser mais caro quanto dispositivos equivalentes que possuem inteligência embutida.

Você deve evitar o hardware baseado no Windows por duas razões:

1. O primeiro é que aqueles fabricantes não tornam os recursos disponíveis para criar um driver para Linux. Geralmente, o hardware e a interface de software para o dispositivo é proprietária, e a documentação não é disponível sem o acordo de não revelação, se ele estiver disponível. Isto impede seu uso como software livre, desde que os escritores de software grátis descubram o código fonte destes programas.
2. A segunda razão é que quando estes dispositivos tem os processadores embutidos removidos, o sistema operacional deve fazer o trabalho dos processadores embutidos, frequentemente em prioridade de tempo real, e assim a CPU não esta disponível para executar programas enquanto ela esta controlando estes dispositivos.

Assim o usuário típico do Windows não obtém um multi-processamento tão intensivo como um usuário do Linux, o fabricante espera que aquele usuário do Windows simplesmente não note a carga de trabalho que este hardware põe naquela CPU. No entanto, qualquer sistema operacional de multi-processamento, até mesmo Windows 9X, XP e Vista, são prejudicados quando fabricantes de periféricos retiram o processador embutido de suas placas e colocam o processamento do hardware na CPU.

Você pode ajudar a reverter esta situação encorajando estes fabricantes a lançarem a documentação e outros recursos necessários para nós desenvolvermos drivers para estes hardwares, mas a melhor estratégia é simplesmente evitar estes tipos de hardwares até que ele esteja listado no HOWTO de hardwares compatíveis com Linux.

Note que hoje já existem muitos drivers para WinModems e outros hardwares for Windows para o Linux. Veja a lista de hardwares compatíveis no HARDWARE-HOWTO ou procure o driver no site do fabricante de seu dispositivo. Mesmo assim a dica é evitar hardwares for Windows e comprar hardwares inteligentes onde cada um faz sua função sem carregar a CPU.

Dispositivos específicos para GNU/Linux

Esta seção foi retirada do manual de instalação da Debian GNU/Linux. Existem diversos vendedores, agora, que vendem sistemas com a **Debian** ou outra distribuição do GNU/Linux pré-instaladas. Você pode pagar mais para ter este privilégio, mas compra um nível de paz de mente, desde então você pode ter certeza que seu hardware é bem compatível com GNU/Linux. Praticamente todas as placas que possuem processadores próprios funcionam sem nenhum problema no Linux (algumas placas da Turtle Beach e mwave tem suporte de som limitado).

Se você tiver que comprar uma máquina com Windows instalado, leia cuidadosamente a licença que acompanha o Windows; você pode rejeitar a licença e obter um desconto de seu vendedor.

Se não estiver comprando um computador com **GNU/Linux** instalado, ou até mesmo um computador usado, é importante verificar se os hardwares existentes são suportados pelo kernel do **GNU/Linux**. Verifique se seu hardware é listado no *Hardware Compatibility HOWTO*, na documentação do código fonte do kernel no diretório `Documentation/sound` ou consulte um técnico de **GNU/Linux** experiente.

Deixe seu vendedor (se conhecer) saber que o que está comprando é para um sistema **GNU/Linux**. Desta forma isto servirá de experiência para que ele poderá recomendar o mesmo dispositivo a outras pessoas que procuram bons dispositivos para sistemas **GNU/Linux**. Apóie vendedores de hardwares amigos do **GNU/Linux**.

Configurações de Dispositivos

As seções abaixo explicam como fazer configurações em dispositivos diversos no sistema **Linux** como placas de rede, som, gravador de CD entre outras.

Configurando uma placa de rede

Para configurar sua placa de rede no **Linux** siga os passos a seguir:

1. Identifique se sua placa de rede é ISA ou PCI. Caso seja ISA, pode ser preciso alterar a configuração de jumpers ou plug-and-play, evitando conflitos de hardware ou o não funcionamento da placa (veja como configura-la em "Hardwares configuráveis por jumpers, dip-switches, jumperless e Plug-and-Play").
2. Identifique a marca/modelo de sua placa. O programa **lshw** é útil para isto. Caso sua placa seja PCI ou CNR, execute o comando **lspci** e veja a linha "Ethernet".

Em último caso, abra a máquina e procure a marca na própria placa. Quase todos os fabricantes colocam a marca da placa no próprio circuito impresso ou no CI principal da placa (normalmente é o maior).

3. Depois de identificar a placa, será preciso carregar o módulo correspondente para ser usada no **Linux**. Em algumas instalações padrões o suporte já pode estar embutido no kernel, neste caso, você poderá pular este passo.

Para carregar um módulo, digite o comando `modprobe modulo` (Veja ???) . Em placas ISA, geralmente é preciso passar a IRQ e porta de I/O como argumentos para alocar os recursos corretamente. O **modprobe** tentará auto-detectar a configuração em placas ISA, mas ela poderá falhar por algum motivo. Por exemplo, para uma NE 2000: `modprobe ne io=0x300 irq=10`.

Para evitar a digitação destes parâmetros toda vez que a máquina for iniciada é recomendável colocá-lo no arquivo `/etc/modules.conf` da seguinte forma:

```
options ne io=0x300 irq=10
```

A partir de agora, você pode carregar o módulo de sua placa NE 2000 apenas com o comando `modprobe ne`. O parâmetro `io=0x300 irq=10` será automaticamente adicionado. Em sistemas **Debian**, o local correto para colocar as opções de um módulo é em arquivos separados localizados dentro de `/etc/modutils`. Crie um arquivo chamado `/etc/modutils/ne` e coloque a linha:

```
options ne io=0x300 irq=10
```

Depois disso, execute o comando `update-modules` para o sistema gerar um novo arquivo `/etc/modules.conf` com todos os módulos de `/etc/modutils` e substituir o anterior.

4. Após carregar o módulo de sua placa de rede, resta apenas configurar seus parâmetros de rede para coloca-la em rede. Veja “Atribuindo um endereço de rede a uma interface (ifconfig)”.

Configurando uma placa de SOM no Linux

A configuração de dispositivos de audio no Linux é simples, bastando carregar o módulo da placa e ajustar o mixer. Atualmente existem 2 padrões de som no sistema Linux: OSS (Open Sound System) e ALSA (Advanced Linux Sound Architecture).

O OSS foi o primeiro padrão adotado em sistemas **Linux**, que tinha como grande limitação a dificuldade em usar diversas placas e a impossibilidade dos programas utilizarem ao mesmo tempo a placa de som. O ALSA é mais novo, suporta full duplex e outros recursos adicionais, além de manter a compatibilidade com OSS. O ALSA é um padrão mais moderno e garante mais performance para a CPU da máquina, principalmente para a exibição de vídeos, etc.

Configurando uma placa de som usando o padrão OSS

OSS é o presente por padrão desde que o suporte a som foi incluído no kernel. Para configurar uma placa de som para usar este sistema de som, primeiro compile seu kernel com o suporte ao módulo de sua placa de som. Caso seja uma placa ISA, você provavelmente terá que habilitar a seção "Open Sound System" para ver as opções disponíveis (entre elas, a Sound Blaster e compatíveis). Uma olhada na ajuda de cada módulo deve ajudá-lo a identificar quais placas cada opção do kernel suporta.

Caso seu kernel seja o padrão de uma distribuição **Linux**, provavelmente terá o suporte a todas as placas de som possíveis. Siga o passo a passo abaixo para configurar sua placa de som no sistema:

1. Primeiro descubra se sua placa de som é ISA. Caso seja, verifique se os seus recursos estão alocados corretamente (veja “Conflitos de hardware”). Caso seja PCI, AMR, execute o comando `lspci`, procure pela linha "Multimedia" e veja o nome da placa. Você também poderá executar o comando `lshw` para descobrir qual placa você possui (veja “Listando as placas e outros hardwares em um computador”) para detalhes.
2. Carregue o módulo da placa de som com o comando `modprobe módulo` (veja ???). Na **Debian**, você pode executar o comando `modconf` para navegar visualmente entre os módulos disponíveis e carregar os módulos necessários.

Algumas placas (principalmente ISA) requerem que seja especificado o recurso de hardware sejam passados para seu módulo, ou simplesmente você quer especificar isto para manter o uso de hardware sobre seu controle. Alguns dos parâmetros mais usados em placas Sound Blaster são os seguintes:

```
modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330
```

Para evitar ter que passar estes parâmetros todas as vezes para o módulo, você poderá coloca-los no arquivo `/etc/modules.conf` da seguinte forma:

```
options sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330
```

Assim, quando der o comando **modprobe sb** ele será carregado com as opções acima. Na distribuição **Debian**, você deverá criar um arquivo chamado `/etc/modutils/sb` contendo a linha acima, depois execute o **update-modules** para "juntar" todos os arquivos do `/etc/modutils` e criar o `/etc/modules.conf`.

3. Após carregar o módulo correto de sua placa de som, seu sistema de som deverá estar funcionando. Se você utiliza uma distribuição **Linux**, os dispositivos de som como `/dev/audio`, `/dev/dsp`, `/dev/mixer` estarão criados e então poderá passar para o próximo passo. Caso não existam, entre no diretório `/dev` e execute o comando `MAKEDEV audio`.
4. O próximo passo consiste em instalar um programa para controle de volume, tonalidade e outros recursos de sua placa de som. O recomendado é o **aumix** por ser simples, pequeno e funcional, e permitindo restaurar os valores dos níveis de volumes na inicialização (isso evita que tenha que ajustar o volume toda vez que iniciar o sistema).

Caso o **aumix** apareça na tela, sua placa de som já está funcionando! Caso acesse o sistema como usuário, não se esqueça de adicionar seu usuário ao grupo audio para ter permissão de usar os dispositivos de som: `adduser usuario audio`.

Configurando um gravador de CD/DVD no Linux

Caso seu gravador seja IDE, veja "Configurando o suporte a um gravador IDE" caso seja um autêntico gravador com barramento SCSI, vá até "Configurando o suporte a um gravador SCSI".

Configurando o suporte a um gravador IDE

Caso tenha um gravador IDE e use um kernel 2.6 ou superior, não é necessário fazer qualquer configuração, pois seu gravador já está pronto para ser usado, sendo acessado através de seu dispositivo tradicional (`/dev/hdc`, `/dev/hdd`, etc). De qualquer forma, você poderá realizar a configuração da unidade IDE com emulação SCSI, assim como utilizava no kernel 2.4 e inferiores seguindo as instruções abaixo.

Para configurar seu gravador de CD/DVD IDE para ser usado no **Linux** usando o método para o kernel 2.4 e inferiores, siga os seguintes passos:

1. Tenha certeza que compilou o suporte as seguintes características no kernel:

```
Em "ATA/IDE/MFM/RLL support" marque as opções:  
* Include IDE/ATAPI CDRom support  
* SCSI emulation support
```

```
Depois em "SCSI support" marque as opções:  
* SCSI support  
M SCSI CD-ROM Support  
M SCSI Generic Support
```

As opções marcadas como "*" serão embutidas no kernel e as "M" como módulos. Note que ambas as opções "IDE/ATAPI CDRom" e "SCSI Emulation" foram marcadas como embutidas. Isto faz com

que o driver ATAPI tenha prioridade em cima do SCSI, mas vou explicar mais adiante como dizer para o kernel para carregar o suporte a SCSI para determinada unidade. Isto é útil quando temos mais de 1 unidade de CD IDE no sistema e queremos configurar somente o gravador para SCSI, pois alguns aplicativos antigos não se comunicam direito tanto com gravadores SCSI como emulados.

Você também pode marcar somente a opção "SCSI Emulation" para que sua(s) unidade(s) seja(m) automaticamente emulada(s) como SCSI. Caso tenha usado esta técnica, vá até a seção "Testando o funcionamento".

2. O próximo passo é identificar o dispositivo de CD/DVD. Isto é feito através do comando `dmesg`. Supondo que sua unidade de CD é "hdc" (primeiro disco na segunda controladora IDE) e que compilou ambos o suporte a "IDE ATAPI" e "SCSI emulation" no kernel, adicione o argumento "hdc=ide-scsi" no `/etc/lilo.conf` ou no **grub**:

```
# Lilo
vmlinuz=/vmlinuz
append="hdc=ide-scsi"
```

Isto diz para o kernel que a unidade "hdc" usará emulação "ide-scsi". Caso tenha outras unidades de CD no sistema, estas ainda utilização ATAPI como protocolo de comunicação padrão. Execute o **lilo** para gerar novamente o setor de inicialização com as modificações e reinicie o computador.

OBS: Cuidado ao colocar um disco rígido IDE como hdc! A linha `hdc=ide-scsi` deverá ser retirada, caso contrário, seu disco rígido não será detectado.

Agora, siga até "Testando o funcionamento".

Configurando o suporte a um gravador SCSI

Caso tenha um autentico gravador SCSI, não será preciso fazer qualquer configuração de emulação, a unidade estará pronta para ser usada, desde que seu suporte esteja no kernel. As seguintes opções do kernel são necessárias para funcionamento de gravadores SCSI:

```
Depois em "SCSI support" marque as opções:
* SCSI support
M SCSI CD-ROM Support
M SCSI Generic Support
```

Além disso, deve ser adicionado o suporte EMBUTIDO no kernel a sua controladora SCSI. Se o seu disco rígido também é SCSI, e seu CD está ligado na mesma controladora SCSI, ela já está funcionando e você poderá seguir para o passo "Testando o funcionamento". Caso contrário carregue o suporte da sua placa adaptadora SCSI antes de seguir para este passo.

Testando o funcionamento

Para testar se o seu gravador, instale o pacote e execute o comando: `wodim -scanbus` para verificar se sua unidade de CD-ROM é detectada.

Você deverá ver uma linha como:

```
scsibus0:
    0,0,0    0) 'CREATIVE' 'CD-RW RWXXXX' '1.00' Removable CD-ROM
```

```
0,1,0    1) *
0,2,0    2) *
```

O que significa que sua unidade foi reconhecida perfeitamente pelo sistema e já pode ser usada para gravação. Vá até a seção ??? para aprender como gravar CDs no **Linux**. Note que gravadores IDE nativos, não são listados com esse comando.

Configurando o gerenciamento de energia usando o APM

O APM (*Advanced Power Management - Gerenciamento Avançado de Energia*) permite que sistemas gerenciem características relacionadas com o uso e consumo de energia do computador. Ele opera a nível de BIOS e tenta reduzir o consumo de energia de várias formas quando o sistema não estiver em uso (como reduzindo o clock da CPU, desligar o HD, desligar o monitor, etc.).

O uso de advanced power management também permite que computadores com fonte de alimentação ATX sejam desligados automaticamente quando você executa o comando **halt**. Caso sua máquina tenha suporte a *ACPI*, este deverá ser usado como preferência ao invés do APM por ter recursos mais sofisticados (veja “Configurando o gerenciamento de energia usando ACPI”).

Para ativar o suporte a APM no **Linux**, compile seu kernel com o suporte embutido a APM e também a "Advanced Power Management" (senão sua máquina não desligará sozinha no **halt**). Caso deseje compilar como módulo, basta depois carregar o módulo `apm` adicionando no arquivo `/etc/modules`. Depois disso instale o daemon **apmd** para gerenciar as características deste recurso no sistema.

Você pode desativar o uso de APM de 3 formas: removendo seu suporte do kernel, passando o argumento `apm=off` (quando compilado estaticamente no kernel) ou removendo o nome do módulo do arquivo `/etc/modules` (quando compilado como módulo). Depois disso remova o daemon **apmd**.

Configurando o gerenciamento de energia usando ACPI

O ACPI (*Advanced Configuration and Power Interface - Interface de Configuração e Gerenciamento de Energia Avançado*) é uma camada de gerenciamento de energia que opera a nível de sistema operacional. Apresenta os mesmos recursos que o APM, e outros como o desligamento da máquina por teclas especiais de teclado, controle de brilho e contraste de notebooks, suspend para RAM, suspend para disco, redução de velocidade de CPU manualmente, monitoramento de periféricos, temperatura, hardwares, etc.

Desta forma, o ACPI varia de sistema para sistema em questões relacionadas com suporte a recursos especiais, estes dados são armazenados em tabelas chamadas DSDT. O **Linux** inclui suporte a recursos ACPI genéricos entre placas mãe, recursos específicos devem ser extraídos diretamente da BIOS e disassemblados manualmente para a construção de um kernel com suporte específico a tabela DSDT do hardware (não falarei das formas de se fazer disso aqui, somente do suporte genérico).

Quanto mais nova a versão do kernel, maiores as chances do seu hardware ser suportado plenamente pelo ACPI, principalmente no caso de notebooks. Para compilar estaticamente, marque com `Y` a opção ACPI, depois marque os módulos que você quer que ele monitore: `button` (botão power), `fan` (ventoinhas), etc. Se compilou como módulo, adicione o nome do módulo `acpi` no arquivo `/etc/modules`. Não há problema em compilar também o suporte a APM, pois não causará problemas com um kernel com ACPI também compilado.

Caso não saiba quais módulos ACPI seu sistema aceita, marque o suporte a todos e carregue-os. Após isto, entre no diretório `/proc/acpi` e de um `ls` entrando nos diretórios e vendo se existem arquivos dentro deles. Remova o módulo correspondente daqueles que não tiver conteúdo.

Após isto, instale o daemon **acpid** e configure-o para monitorar algumas características do seu sistema. Por padrão o **acpid** monitora o botão POWER, assim se você pressionar o power, seu sistema entrará automaticamente em run-level 0, fechando todos os processos e desligando sua máquina.

O suporte a ACPI pode ser desativado de 3 formas: Removendo seu suporte do kernel, passando o argumento `acpi=off` ao kernel (caso esteja compilado estaticamente) ou removendo o módulo de `/etc/modules` (caso tenha compilado como módulo. Após isto, remova o daemon **acpid** do seu sistema.

Ativando WakeUP on Lan

Algumas placas mãe ATX possuem suporte a este interessante recurso, que permite sua máquina ser ligada através de uma rede. Isto é feito enviando-se uma seqüência especial de pacotes diretamente para o MAC (endereço físico) da placa de rede usando um programa especial.

Para usar este recurso, seu sistema deverá ter as seguintes características:

- Placa mãe ATX
- Fonte de alimentação ATX compatível com o padrão 2.0, com fornecimento de pelo menos 720ma de corrente na saída +3v.
- Placa de rede com suporte a WakeUP-on-Lan (WOL), você poderá confirmar isto vendo um conector branco de 3 terminais instalado na placa que é o local onde o cabo wake-up é conectado.
- Suporte na BIOS também deverá ter a opção para WakeUP-on-Lan.

Com todos esses itens existentes, instale em uma máquina da rede o pacote . Depois disso, pegue o MAC address a placa de rede da máquina que tem o wakeup on lan e na máquina da rede onde instalou o pacote execute o seguinte comando:

```
ether-wake AA:BB:CC:DD:EE:FF
```

Onde AA:BB:CC:DD:EE:FF é o endereço MAC da placa de rede. A máquina deverá ligar e realizar o procedimento padrão de POST normalmente.

Algumas das situações onde o WOL não funciona é quando sua rede é controlada por Switches (devido a natureza de funcionamento deste equipamentos) ou caso esteja atrás de um roteador que não faz proxy arp.

Aterramento

O aterramento correto da instalação elétrica é *essencial* para garantir a proteção de seu microcomputador (e outros aparelhos que requerem isto). Muitos usuários simplesmente removem o pino central da tomada de seu computador, ou ligam o terra junto ao neutro da rede elétrica, isto é errado e pode trazer sérias conseqüências. O computador possui componentes sensíveis que geram descargas estáticas durante seu funcionamento (fonte, discos, placas, etc), estas descargas e ruídos são absorvidas pelo sistema de aterramento (que é ligado no gabinete do computador e outros componentes internos). Sem aterramento o seu gabinete passará a dar choques elétricos (teste com uma chave de testes, ela acenderá indicando a circulação de corrente elétrica) e a corrente acumulada poderá queimar componentes internos sensíveis (placa mãe, HD, memórias, placas expansoras).

A ligação do terra ao neutro da rede é menos perigosa em condições normais, mas se um raio cair na rede elétrica as conseqüências poderão ser piores. Mesmo a rede de iluminação pública tendo aterramento em cada poste isto pode não ser o suficiente para reduzir a carga de um raio que caia nas proximidades.

O sistema de aterramento residencial para PC deve ser feito com uma estaca de cobre com no mínimo 2 metros de altura. O cobre é um ótimo condutor de eletricidade, perdendo somente para a prata (veja “Condutores de eletricidade”). Cave um buraco no solo com a ajuda de uma cavadeira (hehe, nunca ouviu falar nisso? :-), se estiver com dificuldades para cavar por causa de solo ressecado, molhe a terra para facilitar as coisas. Com a estaca enterrada, prenda um cabo elétrico em sua extremidade.

O ideal para testar este sistema de aterramento seria ter um equipamento chamado *terrômetro* (medidor de aterramento), mas utilizaremos 2 alternativas mais acessíveis:

- Ligue uma lâmpada incandescente de 100W em um bocal com uma ponta ligada na extremidade positiva da rede elétrica (fase) e a outra ponta no fio da barra de cobre. O aterramento está bem feito quando a lâmpada acender quase em sua potência total. Ligue o fio do aterramento no pino central da tomada de seu computador.

OBS: Cuidado para não tomar um baita choque durante esta operação em alguns casos pode ser fatal. Utilize sandalhas ou sapatos de borracha (materiais isolantes) isto evitará tomar o choque caso aconteça.

- Ligue a outra extremidade do fio que vem da barra de cobre no pino central da tomada de seu computador e ligue-o. Consiga um multímetro (analogico ou digital) e coloque para medir em escala DC 10V. Coloque a ponta negativa (preta) no *neutro* da rede elétrica e encoste a ponta positiva (vermelha) no gabinete de seu computador. O aterramento estará aprovado caso o valor medido seja de no máximo 2.5 volts.

Caso algo ocorra errado, cheque novamente os passos acima. Se desconfiar das condições do solo, use uma barra maior ou ligue 2 barras de cobre juntas.

Condutores de eletricidade

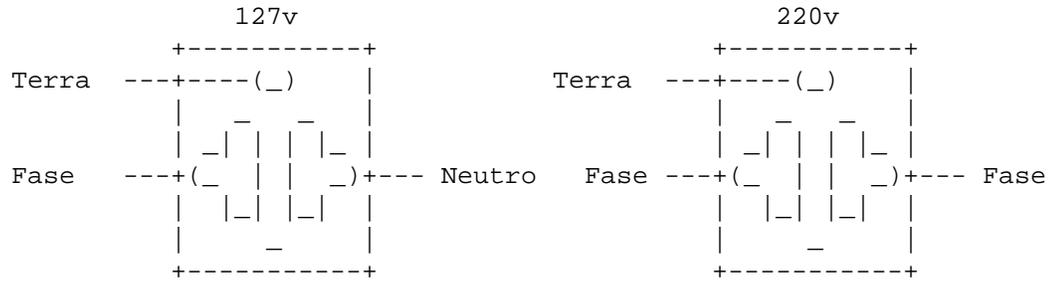
A tabela abaixo está classificada em ordem do material que possui melhor condução de eletricidade (elétrons com circulações livres) baseada no fator mm²/m. (da mais condutora para a menos condutora).

1. Prata - 0,0164
2. Cobre - 0,0172
3. Ouro - 0,0230
4. Alumínio - 0,0283
5. Zinco - 0,0600
6. Platina - 0,0950
7. Ferro - 0,1200
8. Chumbo - 0,2100
9. mercúrio - 0,9680

Tomadas

As tomadas elétricas de 127V ou 220V AC 60Hz de três pinos, pelas normas técnicas da ABNT, no. ABNT 6147 devem ficar distantes no máximo a 1,5 metro dos equipamentos e com terceiro pino ligado à terra. É interessante que a tensão das tomadas esteja identificada nas mesmas, em caso de mais de uma voltagem fornecida no local, evitando a queima de equipamentos.

Segue abaixo um exemplo de tomada fêmea e a recomendação para sua montagem. Note que a entrada para o pino terra é voltado para cima, pois o caimento dos fios da maioria dos equipamentos fabricados estarão desta forma voltados para baixo.



Como comentando anteriormente, não utilize como ponto de terra os sistemas de aterramento das companhias de eletricidade, telefonia, ar condicionado e sistema de pára-raios.

Descargas estáticas

É a energia que se acumula durante o choque das moléculas de ar seco ou atrito com outros objetos. Pode acontecer de em dias secos você tomar um "choque" ao abrir seu carro ou tocar em algum objeto metálico, isto é uma descarga estática. Na realidade você não tomou um choque, ao tocar em um objeto metálico esta energia é descarregada violentamente. Esta energia pode chegar na ordem de 5 mil volts quando acumulada (assustador não?).

É por este motivo que caminhões que transportam combustível arrastam uma corrente no chão, esta corrente funciona como um aterramento (veja "Aterramento") eliminando descargas estáticas que possam gerar faíscas e causar um desastre. Pulseiras, cordões, objetos metálicos podem ser usados para eliminar descargas estáticas de pessoas. O contato freqüente com o solo é um método muito útil. Existem casos em que um colar salvou a vida de pessoas atingidas por raio, justamente pelas explicações acima. O colar derrete com a drenagem da eletricidade do raio mas a pessoa tem mais chances de sair viva.

Em indivíduos realmente sensíveis, uma chapinha de metal pode ser colocada no sapato fazendo contato com o calcanhar drenando constantemente estas descargas, isto é eficaz e bem melhor que sair arrastando correntes por ai :-)

Se você trabalha com hardwares ou é apenas mais um fuçador de PCs, agora você entenderá porque é recomendável sempre tocar em partes metálicas do computador antes de mexer em qualquer placa e porque aquele seu amigo disse que a placa dele queimou depois que resolveu limpar seus contatos.

Melhoria de performance

Particionamento

Para um melhor desempenho, os dados que são solicitados constantemente deverão ser armazenados em uma partição no início do disco rígido. Esta área é a mais rápida e checa a ser 60% mais rápida que o final do HD (em alguns modelos). Em especial, a partição de boot, swap e binários do sistema poderão ser armazenados nesta partição para aumentar a velocidade da carga de programas e não prejudicar a performance do sistema quando o uso da partição de troca (swap) for necessária.

Em discos rígidos grandes (6GB ou maiores) é recomendável criar no mínimo uma partição pequena para /boot, outra para /, outra para swap e outra para /usr. Ficando distribuídas da seguinte maneira no disco rígido:

```

BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
SSSSSSSSSSUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUU

```

```

B - /boot
R - Raíz /
S - Swap
U - /usr

```

Mas a swap não ficaria ainda mais rápida sendo a primeira partição no disco? Sim e não: Realmente fica rápida (na teoria conforme explicado acima), mas levando em consideração que o deslocamento das cabeças de leitura/gravação do disco rígido leva certo tempo, é mais vantajoso mantê-la entre as 2 partições mais acessadas, isto diminui o tempo de acesso caso um programa esteja fazendo uso constante de / ou /usr e precisar trocar dados na partição swap.

Além do mais, a partição / geralmente é pequena (no máximo 800M) deixando a swap em uma área muito próxima do início do disco rígido. Com base nisto, você poderá ter uma melhor visão técnica para a construção de suas partições dependendo da função do sistema.

Spindles

Em sistemas que utilizam um disco rígido dedicado para fazer swap, a ligação deste em uma placa controladora independente aumentará bastante a performance do sistema, pois enquanto o disco principal ligado em sua controladora estiver fazendo uma operação de leitura, o outro poderá estar fazendo sua operação de swap simultaneamente. O mesmo não acontece quando dois discos rígidos IDE estão ligados no mesmo cabo (isto não acontece no SCSI).

Fazendo ajustes finos de performance do disco

O **hdparm** é um programa que permite modificar características diversas da unidade de disco rígido e de CD como modo de transferência de dados, leitura adiante, dma, cache, leitura simultânea de setores, hibernação, etc.

Por padrão as transferências de dados entre a controladora do HD (a plaquinha que fica embaixo dele) e a controladora de periféricos é feita em 16 bits. Para exibir a configuração atual do disco rígido `/dev/hda` (por exemplo), digite o seguinte comando: `hdparm /dev/hda`

```

/dev/hdb:
multcount      = 0 (off)
I/O support    = 0 (16-bit)
unmaskirq     = 0 (off)
using_dma      = 1 (off)
keepsettings   = 0 (off)
nowerr         = 0 (off)
readonly       = 0 (off)

```

readahead = 8 (on)

Imediatamente podemos modificar os seguintes campos para melhorar sensivelmente o desempenho do disco rígido:

multcount Pode ser modificada com `-m[num]` e especifica o número máximo de setores que serão acessados de uma só vez na operação de leitura da unidade. O valor máximo recomendado é igual a capacidade máxima suportada pelo seu disco rígido, que pode ser obtida com o comando: `hdparm -i /dev/hda`

```
Model=TS6324A2, FwRev=.340      , SerialNo=A99B99JA
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs RotSpdTol>.5% }
RawCHS=13228/15/63, TrkSize=0, SectSize=0, ECCbytes=0
BuffType=unknown, BuffSize=256kB, MaxMultSect=16, MultSect=16
CurCHS=13228/15/63, CurSects=12500460, LBA=yes, LBAsects=12500460
IORDY=on/off, tPIO={min:240,w/IORDY:120}, tDMA={min:120,rec:120}
PIO modes: pio0 pio1 pio2 pio3 pio4
DMA modes: mdma0 mdma1 mdma2 udma0 udma1 *udma2 udma3 udma4 udma5
```

O campo `MaxMultSect=16` indica o valor de 16 como máximo suportado em uma única operação pela unidade. Valores maiores poderão ser especificados mas não trarão ganho de performance. Para discos rígidos *Western Digital* é recomendável deixar este valor como 0, porque eles possuem um mecanismo embutido para leitura de setores. Para experimentar valores fora dos padrões, coloque seu sistema de arquivos como somente leitura para não perder dados caso saia errado.

Note que o comando `hdparm -i` mostra alguns detalhes interessantes sobre a configuração do disco rígido e modos de operação suportados.

I/O support Modificado com `-c[num]`. O número especificado pode ser 0 para transferência de dados em 16 bits, 1 para 32 bits e 3 para 32 bits com uma seqüência especial de sincronismo (alguns chips requerem esta ao invés da 1).

using_dma Modificado com `-d[num]`. Habilita ou não o uso de DMA para a transferência de dados do HD, ativando o controle de algumas operações pelo chipset livrando a CPU para processamento. 0 desativa DMA e 1 ativa. Note que nem todos os chipsets aceitam esta operação. Esta usada em conjunto com a opção `-X` oferece um excelente ganho de performance pelo uso correto de sua controladora.

A ativação de dma também pode ser feita automaticamente na recompilação do kernel ou especificando o parâmetro `ideX=dma` (X é o número da controladora IDE) na linha de comando de `boot:` ou no arquivo `/etc/lilo.conf`.

xfermode Modificado pela opção `-X[num]`. Permite selecionar o melhor modo de transferência de dados do seu disco rígido, é nesta parte onde você seleciona o modo UltraDMA para transferência de dados, caso seu HD/CD-ROM suporte. Os seguintes valores são válidos:

- 32 - PIO Mode 0
- 33 - PIO Mode 1
- 34 - PIO Mode 2
- 35 - PIO Mode 3
- 36 - PIO Mode 4
- 64 - Ultra DMA Mode 0

- 65 - Ultra DMA Mode 1
- 66 - Ultra DMA Mode 2
- 67 - Ultra DMA Mode 3
- 68 - Ultra DMA Mode 4
- 69 - Ultra DMA Mode 5
- 70 - Ultra DMA Mode 6

Para descobrir os modos PIO e UltraDMA do seu HD, utilize o comando `hdparm -I /dev/hd?`. NÃO UTILIZE UM MODO PIO/ULTRA DMA NÃO SUPOSTADO PELA SUA CONTROLADORA. CASO SUA PLACA CONTROLADORA DO HD SUPORTE UM MODO ALTO PIO/ULTRADMA MAS SUA CONTROLADORA IDE NÃO SUPORTA, VOCÊ DEVERÁ UTILIZAR O VALOR MAIS ADEQUADO PARA AMBAS. FAÇA TESTES SEMPRE QUE ALTERAR O MODO DE FUNCIONAMENTO E ESTEJA ATENTO A MENSAGENS DE ERROS PARA QUE NÃO TENHA PERDA DE DADOS!!!

- unmaskirq** Modificado com `-u[num]`. Habilita ou não o controlador de disco mascarar as interrupções de processador durante o processamento das interrupções de disco. 0 desativa esta função e 1 ativa. Use esta opção com cuidado e sob seu próprio risco: algumas placas controladores de HD e controladoras de periféricos não trabalham bem com a taxa de transferência aumentada, podem ocorrer perda de dados. Coloque o sistema de arquivos como somente leitura antes de testar esta característica.
- readonly** Modificado com `-r[num]`. Coloca o disco em modo somente leitura. A montagem da partição com a opção `ro` no `/etc/fstab` é preferida.
- readahead** Modificado com `-a[num]`. Configura o número de blocos que serão lidos antecipadamente no sistema de arquivos (por padrão é usado 8 blocos - 4 Kb). Este número poderá ser modificado para se adequar a utilização do computador. Em sistemas com muita procura de arquivos pequenos (servidores web), um valor pequeno (como o padrão) é recomendável. Se a máquina é um servidor de arquivos dedicado, um valor maior trará maiores benefícios.

Veja mais detalhes sobre o comando **hdparm** em sua página de manual. Depois de selecionado o melhor valor de performance, você deverá salvar em um arquivo que será lido na inicialização para ativação destes valores. Para fazer teste de performance de leitura bruta utilize o comando `hdparm -t /dev/hd?`, para fazer testes com o uso de cache, use o comando `hdparm -T /dev/hd?`.

OBS: Se o **Linux** resetar o disco rígido, a maioria das configurações retornarão ao seu valor padrão. Isto ocorre devido a opções mau utilizadas no **hdparm**, não suportadas pelo disco rígido ou por problemas no HD/controladora.

Exemplos:

```
# Ajusta o número de setores simultâneos para 16 e o modo de transferência para
# 32 bits no disco rígido /dev/hda
hdparm -c1 -m16 /dev/hda

# Programa a leitura adiante do HD para 64 blocos (32Kb), o modo de transferência
# para 32 bits, usar DMA, e 16 setores simultâneos.
hdparm -c1 -d1 -m16 -a64 /dev/hda

#Mostra os valores de configuração atuais do disco rígido
hdparm /dev/hda
```

Data de acesso a arquivos/diretórios

Toda vez que acessamos um arquivo ou diretório da máquina **Linux** a data/hora é atualizada. Em máquinas normais isto é OK mas em servidores onde o acesso a arquivos é constante (como no diretório `/var/spool` em servidores de e-mail ou `/usr/` em servidores diskless) é recomendável desativar esta característica. Isto reduzirá a quantidade de buscas das cabeças do disco rígido para a atualização deste atributo e conseqüentemente aumentará a performance na gravação de arquivos (o disco rígido usa o sistema mecânico para ler/gravar dados, muito mais lento que a memória RAM eletrônica).

```
chattr -R +A /var/spool
```

O atributo `+A` desativa a gravação da "data de acesso" dos arquivos e sub-diretórios dentro de `/var/spool`. Para desativar a atualização da "data de acesso" para toda a partição, você pode incluir a opção de montagem `noatime` no seu `/etc/fstab`:

```
/dev/hda1    /var/spool    ext2    defaults,noatime    0    1
```

OBS: O **Linux** utiliza três atributos de data para controle de arquivos:

- `atime` - Data/Hora de acesso: é atualizado toda vez que o arquivo é lido ou executado.
- `mtime` - Data/Hora da modificação, atualizado sempre que alguma modificação ocorre no arquivo ou no conteúdo do diretório. Esta é mais interessante que a `ctime` principalmente quando temos hardlinks.
- `ctime` - Data/Hora da última modificação do inodo do arquivo.

Em partições onde a gravação é freqüente (como na própria `/var/spool`) a desativação do atributo `atime` além de melhorar o desempenho do disco, não fará muita falta.

Periféricos SATA

Hardwares SATA (Serial ATA) representam a próxima geração em tecnologia usada para a transferência de dados em alta velocidade a baixo custo. Hoje está se tornando o padrão de indústria a utilização de dispositivos SATA em micros em substituição a dispositivos IDE. Dispositivos IDE tradicionais são chamados de PATA (parallel ATA, ou ATA paralelo).

Estes dispositivos são classificados em 2 tipos:

- **SATA I** - Esta se tornando alternativa a discos IDE (PATA). Possui taxa de transferência de até 150Mb/s
- **SATA II** - Esta se tornando alternativa a discos IDE (PATA). Possui taxa de transferência de até 300Mb/s

Um cabo SATA tende a ter o mesmo comprimento de um cabo IDE, raramente excedendo 50 centímetros.

Periféricos SCSI

Hardwares SCSI (Small Computer System Interfaces) representam a tecnologia ideal para a transferência de dados em alta velocidade e ligação de vários periféricos. A taxa de transferência especificada para dispositivos SCSI é sempre a padrão se comparada a dispositivos IDE (quando uma taxa de 66Mb/s quase nunca é atingida).

Estes dispositivos são classificados em 3 categorias:

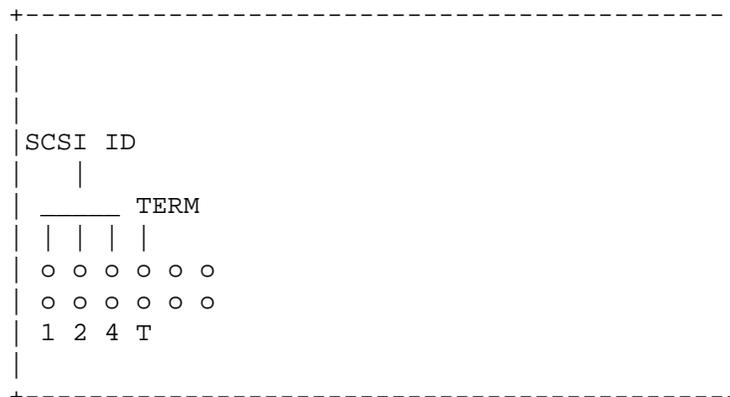
- SCSI I - Usa um cabo de 25 condutores para a ligação de periféricos. Normalmente usado em scanners, impressoras e outros dispositivos. A taxa de transferência não é muito alta se comparado aos outros tipos SCSI.
- SCSI II - Também chamado de *Fast SCSI*. Usa um cabo de 50 condutores para a ligação de periféricos. Permite que sejam ligados até 7 periféricos em uma mesma controladora (veja “Configurando uma SCSI ID e terminação”). É o mais comum encontrado hoje em dia, mas vem perdendo espaço para a tecnologia SCSI III.
- SCSI III - Também chamado de *Fast SCSI SE* ou *LVD*. Usa um cabo de 68 condutores para ligação de periféricos (veja “Configurando uma SCSI ID e terminação”). Permite que sejam ligados até 16 periféricos em uma mesma controladora.
- SATA I - Esta se tornando alternativa a discos IDE (PATA). Possui taxa de transferência de até 150Mb/s
- SATA II - Esta se tornando alternativa a discos IDE (PATA). Possui taxa de transferência de até 300Mb/s

Um cabo SCSI pode ter o comprimento de até 5 metros de extensão. Os periféricos SCSI são identificados através de números chamados de identificador SCSI ou SCSI ID. Estes números vão de 0 a 6 para o padrão SCSI 2 e de 0 a 15 para o padrão SCSI 3.

Placas SCSI como a *Adaptec UV 19160* permitem a ligação de periféricos SCSI 2 e SCSI 3 na mesma placa com a taxa de transmissão de 160 MB/s por periférico, além de possuir um "setup" próprio para configurar as opções dos dispositivos da placa e a operação da própria. A tecnologia SCSI é algo realmente rápido para a transferência de dados e cara também, seu uso é muito recomendado em servidores críticos. Os próprios dispositivos SCSI como discos rígidos, gravadores de CD, cd-rom, etc. são construídos de tal forma que tem a durabilidade maior que periféricos comuns, garantindo a máxima confiança para operação/armazenamento de dados em longos períodos de operação.

Configurando uma SCSI ID e terminação

Uma SCSI ID é configurada independentemente por dispositivo e consiste em 3 jumpers (ou dip switches) que possuem os valores 1, 2 e 4. Veja o exemplo abaixo de uma unidade de CD SCSI 2:



Se você deixar os 3 jumpers da SCSI ID abertos, o dispositivo usará a SCSI ID 0. Colocando o jumper na posição 1, a unidade terá a SCSI ID 1. Se você colocar um jumper na posição 1 e outro na 4, a unidade será identificada pela SCSI ID 5 (quando mais de um jumper é ligado, os números serão somados).

A terminação SCSI funciona de forma semelhante a de uma rede BNC, o último periférico do cabo SCSI deve ter o jumper de terminação colocado para indicar que é o último periférico do cabo e evitar deflexão de dados. Algumas placas SCSI modernas ajustam automaticamente a terminação de periféricos sem necessidade de ajustar manualmente.

Capítulo 4. Rede

Este capítulo descreve o que é uma rede, os principais dispositivos de rede no **GNU/Linux**, a identificação de cada um, como configurar os dispositivos, escolha de endereços IP, roteamento.

Parte deste capítulo, uns 70% pelo menos, é baseado no documento NET3-4-HOWTO. (seria perda de tempo reescrever este assunto pois existe um material desta qualidade já disponível).

O que é uma rede

Rede é a conexão de duas ou mais máquinas com o objetivo de compartilhar recursos entre uma máquina e outra. Os recursos podem ser:

- Compartilhamento do conteúdo de seu disco rígido (ou parte dele) com outros usuários. Os outros usuários poderão acessar o disco como se estivesse instalado na própria máquina). Também chamado de servidor de arquivos.
- Compartilhamento de uma impressora com outros usuários. Os outros usuários poderão enviar seus trabalhos para uma impressora da rede. Também chamado de servidor de impressão.
- Compartilhamento de acesso a Internet. Outros usuários poderão navegar na Internet, pegar seus e-mails, ler notícias, bate-papo no IRC, ICQ através do servidor de acesso Internet. Também chamado de servidor Proxy.
- Servidor de Internet/Intranet. Outros usuários poderão navegar nas páginas Internet localizadas em seu computador, pegar e-mails, usar um servidor de IRC para chat na rede, servidor de ICQ, etc

Com os ítems acima funcionando é possível criar permissões de acesso da rede, definindo quem terá ou não permissão para acessar cada compartilhamento ou serviço existente na máquina (www, ftp, irc, icq, etc), e registrando/avisando sobre eventuais tentativas de violar a segurança do sistema, firewalls, pontes, etc.

Entre outras ilimitadas possibilidades que dependem do conhecimento do indivíduo no ambiente **GNU/Linux**, já que ele permite muita flexibilidade para fazer qualquer coisa funcionar em rede.

A comunicação entre computadores em uma rede é feita através do *Protocolo de Rede*.

Protocolo de Rede

O protocolo de rede é a linguagem usada para a comunicação entre um computador e outro. Existem vários tipos de protocolos usados para a comunicação de dados, alguns são projetados para pequenas redes (como é o caso do NetBios) outros para redes mundiais (TCP/IP que possui características de roteamento).

Dentre os protocolos, o que mais se destaca atualmente é o TCP/IP devido ao seu projeto, velocidade e capacidade de roteamento.

Endereço IP

O *endereço IP* são números que identificam seu computador em uma rede. Inicialmente você pode imaginar o IP como um número de telefone. O IP é composto por quatro bytes e a convenção de escrita dos números é chamada de "notação decimal pontuada". Por convenção, cada interface (placa usada p/ rede) do computador ou roteador tem um endereço IP. Também é permitido que o mesmo endereço IP

seja usado em mais de uma interface de uma mesma máquina mas normalmente cada interface tem seu próprio endereço IP.

As Redes do Protocolo Internet são seqüências contínuas de endereços IP's. Todos os endereços dentro da rede tem um número de dígitos dentro dos endereços em comum. A porção dos endereços que são comuns entre todos os endereços de uma rede são chamados de *porção da rede*. Os dígitos restantes são chamados de *porção dos hosts*. O número de bits que são compartilhados por todos os endereços dentro da rede são chamados de *netmask* (máscara da rede) e o papel da *netmask* é determinar quais endereços pertencem ou não a rede. Por exemplo, considere o seguinte:

```

-----
Endereço do Host   192.168.110.23
Máscara da Rede   255.255.255.0
Porção da Rede    192.168.110.
Porção do Host    .23
-----
Endereço da Rede  192.168.110.0
Endereço Broadcast 192.168.110.255
-----

```

Qualquer endereço que é finalizado em zero em sua *netmask*, revelará o *endereço da rede* que pertence. O endereço e rede é então sempre o menor endereço numérico dentro da escalas de endereços da rede e sempre possui a *porção host* dos endereços codificada como zeros.

O endereço de *broadcast* é um endereço especial que cada computador em uma rede "escuta" em adição a seu próprio endereço. Este é um endereço onde os datagramas enviados são recebidos por todos os computadores da rede. Certos tipos de dados como informações de roteamento e mensagens de alerta são transmitidos para o endereço *broadcast*, assim todo computador na rede pode recebe-las simultaneamente.

Existe dois padrões normalmente usados para especificar o endereço de *broadcast*. O mais amplamente aceito é para usar o endereço mais alto da rede como endereço broadcast. No exemplo acima este seria 192.168.110.255. Por algumas razões outros sites tem adotado a convenção de usar o endereço de rede como o endereço broadcast. Na prática não importa muito se usar este endereço, mas você deve ter certeza que todo computador na rede esteja configurado para escutar o mesmo *endereço broadcast*.

Classes de Rede IP

Por razões administrativas após algum pouco tempo no desenvolvimento do protocolo IP alguns grupos arbitrários de endereços foram formados em redes e estas redes foram agrupadas no que foram chamadas de *classes*. Estas classes armazenam um tamanho padrão de redes que podem ser usadas. As faixas alocadas são:

Classe	Máscara de Rede	Endereço da Rede
A	255.0.0.0	0.0.0.0 - 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

O tipo de endereço que você deve utilizar depende exatamente do que estiver fazendo.

Referência rápida de máscara de redes

A tabela abaixo faz referência as máscaras de rede mais comuns e a quantidade de máquinas máximas que ela atinge. Note que a especificação da máscara tem influência direta na classe de rede usada:

Máscara (Forma octal)	Máscara (Forma 32 bits)	Número Máximo de Máquinas
Classe A:		
/8	/255.0.0.0	16,777,215
Classe B:		
/16	/255.255.0.0	65,535
/17	/255.255.128.0	32,767
/18	/255.255.192.0	16,383
/19	/255.255.224.0	8,191
/20	/255.255.240.0	4,095
/21	/255.255.248.0	2,047
/22	/255.255.252.0	1,023
/23	/255.255.254.0	511
Classe C		
/24	/255.255.255.0	255
/25	/255.255.255.128	127
/26	/255.255.255.192	63
/27	/255.255.255.224	31
/28	/255.255.255.240	15
/29	/255.255.255.248	7
/30	/255.255.255.252	3
/32	/255.255.255.255	1

Qualquer outra máscara fora desta tabela (principalmente para a classe A), deverá ser redimensionada com uma calculadora de IP para chegar a um número aproximado de redes/máquinas aproximados que deseja.

Para instalar uma máquina usando o Linux em uma rede existente

Se você quiser instalar uma máquina **GNU/Linux** em uma rede TCP/IP existente então você deve contactar qualquer um dos administradores da sua rede e perguntar o seguinte:

- Endereço IP de sua máquina
- Endereço IP da rede
- Endereço IP de broadcast
- Máscara da Rede IP
- Endereço do Roteador
- Endereço do Servidor de Nomes (DNS)

Você deve então configurar seu dispositivo de rede **GNU/Linux** com estes detalhes. Você não pode simplesmente escolhê-los e esperar que sua configuração funcione.

Endereços reservados para uso em uma rede Privada

Se você estiver construindo uma rede privada que nunca será conectada a Internet, então você pode escolher qualquer endereço que quiser. No entanto, para sua segurança e padronização, existem alguns endereços IP's que foram reservados especificamente para este propósito. Eles estão especificados no RFC1597 e são os seguintes:

ENDEREÇOS RESERVADOS PARA REDES PRIVADAS		
Classe de Rede	Máscara de Rede	Endereço da Rede
A	255.0.0.0	10.0.0.0 - 10.255.255.255
B	255.255.0.0	172.16.0.0 - 172.31.255.255
C	255.255.255.0	192.168.0.0 - 192.168.255.255

Você deve decidir primeiro qual será a largura de sua rede e então escolher a classe de rede que será usada.

Interface de rede

As interfaces de rede no **GNU/Linux** estão localizadas no diretório `/dev` e a maioria é criada dinamicamente pelos softwares quando são requisitadas. Este é o caso das interfaces `ppp` e `plip` que são criadas dinamicamente pelos softwares.

Abaixo a identificação de algumas interfaces de rede no Linux (a ? significa um número que identifica as interfaces seqüencialmente, iniciando em 0):

- `eth?` - Placa de rede Ethernet e WaveLan.
- `ppp?` - Interface de rede PPP (protocolo ponto a ponto).
- `slip?` - Interface de rede serial
- `eql` - Balanceador de tráfego para múltiplas linhas
- `plip?` - Interface de porta paralela
- `arc?e`, `arc?s` - Interfaces Arcnet
- `sl?`, `ax?` - Interfaces de rede AX25 (respectivamente para kernels 2.0.xx e 2.2.xx).
- `fddi?` - Interfaces de rede FDDI.
- `dlci??`, `sdl?` - Interfaces Frame Relay, respectivamente para para dispositivos de encapsulamento DLCI e FRAD.
- `nr?` - Interface Net Rom
- `rs?` - Interfaces Rose
- `st?` - Interfaces Strip (Starmode Radio IP)

- `tr?` - Token Ring

Para maiores detalhes sobre as interfaces acima, consulte o documento *NET3-4-HOWTO*.

A interface loopback

A interface *loopback* é um tipo especial de interface que permite fazer conexões com você mesmo. Todos os computadores que usam o protocolo TCP/IP utilizam esta interface e existem várias razões porque precisa fazer isto, por exemplo, você pode testar vários programas de rede sem interferir com ninguém em sua rede. Por convenção, o endereço IP 127.0.0.1 foi escolhido especificamente para a loopback, assim se abrir uma conexão telnet para 127.0.0.1, abrirá uma conexão para o próprio computador local.

A configuração da interface loopback é simples e você deve ter certeza que fez isto (mas note que esta tarefa é normalmente feita pelos scripts padrões de inicialização existentes em sua distribuição).

```
ifconfig lo 127.0.0.1
```

Caso a interface loopback não esteja configurada, você poderá ter problemas quando tentar qualquer tipo de conexão com as interfaces locais, tendo problemas até mesmo com o comando **ping**.

Atribuindo um endereço de rede a uma interface (ifconfig)

Após configurada fisicamente, a interface precisa receber um endereço IP para ser identificada na rede e se comunicar com outros computadores, além de outros parâmetros como o endereço de *broadcast* e a *máscara de rede*. O comando usado para fazer isso é o **ifconfig** (interface configure).

Para configurar a interface de rede Ethernet (`eth0`) com o endereço 192.168.1.1, máscara de rede 255.255.255.0, podemos usar o comando:

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
```

O comando acima ativa a interface de rede. A palavra `up` pode ser omitida, pois a ativação da interface de rede é o padrão. Para desativar a mesma interface de rede, basta usar o comando:

```
ifconfig eth0 down
```

Digitando `ifconfig` são mostradas todas as interfaces ativas no momento, pacotes enviados, recebidos e colisões de datagramas. Para mostrar a configuração somente da interface `eth0`, use o comando: `ifconfig eth0` Em sistemas **Debian**, o arquivo correto para especificar os dados das interfaces é o `/etc/network/interfaces` (veja ???).

Para mais detalhes, veja a página de manual do **ifconfig** ou o *NET3-4-HOWTO*.

Roteamento

Roteamento é quando uma máquina com múltiplas conexões de rede decide onde entregar os pacotes IP que recebeu, para que cheguem ao seu destino.

Pode ser útil ilustrar isto com um exemplo. Imagine um simples roteador de escritório, ele pode ter um link intermitente com a Internet, um número de segmentos ethernet alimentando as estações de trabalho e outro

link PPP intermitente fora de outro escritório. Quando o roteador recebe um datagrama de qualquer de suas conexões de rede, o mecanismo que usa determina qual a próxima interface deve enviar o datagrama. Computadores simples também precisam rotear, todos os computadores na Internet tem dois dispositivos de rede, um é a interface *loopback* (explicada acima) o outro é um usado para falar com o resto da rede, talvez uma ethernet, talvez uma interface serial PPP ou SLIP.

OK, viu como o roteamento funciona? cada computador mantém uma lista de regras especiais de roteamento, chamada *tabela de roteamento*. Esta tabela contém colunas que tipicamente contém no mínimo três campos, o primeiro é o *endereço de destino*, o segundo é o *nome da interface* que o datagrama deve ser roteado e o terceiro é opcionalmente o *endereço IP* da outra máquina que levará o datagrama em seu próximo passo através da rede. No **GNU/Linux** você pode ver a tabela de roteamento usando um dos seguintes comandos:

```
cat /proc/net/route
route -n
netstat -r
```

O processo de roteamento é muito simples: um datagrama (pacote IP) é recebido, o endereço de destino (para quem ele é) é examinado e comparado com cada item da tabela de roteamento. O item que mais corresponder com o endereço é selecionado e o datagrama é direcionado a interface especificada.

Se o campo *gateway* estiver preenchido, então o datagrama é direcionado para aquele computador pela interface especificada, caso contrário o endereço de destino é assumido sendo uma rede suportada pela interface.

Configurando uma rota no Linux

A configuração da rota é feita através da ferramenta **route**. Para adicionar uma rota para a rede 192.168.1.0 acessível através da interface eth0 basta digitar o comando:

```
route add -net 192.168.1.0 eth0
```

Para apagar a rota acima da *tabela de roteamento*, basta substituir a palavra `add` por `del`. A palavra `net` quer dizer que 192.168.1.0 é um endereço de rede (lembra-se das explicações em “Endereço IP”?) para especificar uma máquina de destino, basta usar a palavra `-host`. Endereços de máquina de destino são muito usadas em conexões de rede apenas entre dois pontos (como `ppp`, `plip`, `slip`). Por padrão, a interface é especificada como último argumento. Caso a interface precise especifica-la em outro lugar, ela deverá ser precedida da opção `-dev`.

Para adicionar uma rota padrão para um endereço que não se encontra na tabela de roteamento, utiliza-se o *gateway padrão da rede*. Através do gateway padrão é possível especificar um computador (normalmente outro gateway) que os pacotes de rede serão enviados caso o endereço não confira com os da tabela de roteamento. Para especificar o computador 192.168.1.1 como *gateway padrão* usamos:

```
route add default gw 192.168.1.1 eth0
```

O *gateway padrão* pode ser visualizado através do comando `route -n` e verificando o campo `gateway`. A opção `gw` acima, especifica que o próximo argumento é um endereço IP (de uma rede já acessível através das tabelas de roteamento).

O computador *gateway* está conectado a duas ou mais redes ao mesmo tempo. Quando seus dados precisam ser enviados para computadores fora da rede, eles são enviados através do computador *gateway* e o *gateway*

os encaminham ao endereço de destino. Desta forma, a resposta do servidor também é enviada através do *gateway* para seu computador (é o caso de uma típica conexão com a Internet).

A nossa configuração ficaria assim:

```
route add -net 192.168.1.0 eth0
route add default gw 192.168.1.1 eth0
```

Para mais detalhes, veja a página de manual do **route** ou o *NET3-4-HOWTO*.

Resolvedor de nomes (DNS)

DNS significa Domain Name System (sistema de nomes de domínio). O *DNS* converte os nomes de máquinas para endereços IPs que todas as máquinas da Internet possuem. Ele faz o mapeamento do nome para o endereço e do endereço para o nome e algumas outras coisas. Um mapeamento é simplesmente uma associação entre duas coisas, neste caso um nome de computador, como `www.cipsga.org.br`, e o endereço IP desta máquina (ou endereços) como `200.245.157.9`.

O *DNS* foi criado com o objetivo de tornar as coisas mais fáceis para o usuário, permitindo assim, a identificação de computadores na Internet ou redes locais através de nomes (é como se tivéssemos apenas que decorar o nome da pessoa ao invés de um número de telefone). A parte responsável por traduzir os nomes como `www.nome.com.br` em um endereço IP é chamada de *resolvedor de nomes*.

O *resolvedor de nomes* pode ser um banco de dados local (controlador por um arquivo ou programa) que converte automaticamente os nomes em endereços IP ou através de *servidores DNS* que fazem a busca em um banco de dados na Internet e retornam o endereço IP do computador desejado. Um servidor DNS mais difundido na Internet é o **bind**.

Através do DNS é necessário apenas decorar o endereço sem precisar se preocupar com o endereço IP (alguns usuários simplesmente não sabem que isto existe...). Se desejar mais detalhes sobre *DNS*, veja o documento *DNS-HOWTO*.

O que é um nome?

Você deve estar acostumado com o uso dos nomes de computadores na Internet, mas pode não entender como eles são organizados. Os nomes de domínio na Internet são uma estrutura hierárquica, ou seja, eles tem uma estrutura semelhante aos diretórios de seu sistema.

Um *domínio* é uma família ou grupo de nomes. Um domínio pode ser colocado em um *sub-domínio*. Um *domínio principal* é um domínio que não é um sub-domínio. Os domínios principais são especificados na RFC-920. Alguns exemplos de domínios principais comuns são:

- COM - Organizações Comerciais
- EDU - Organizações Educacionais
- GOV - Organizações Governamentais
- MIL - Organizações Militares
- ORG - Outras Organizações
- NET - Organizações relacionadas com a Internet

- **Identificador do País** - São duas letras que representam um país em particular.

Cada um dos domínios principais tem sub-domínios. Os domínios principais baseados no nome do país são freqüentemente divididos em sub-domínios baseado nos domínios `.com`, `.edu`, `.gov`, `.mil` e `.org`. Assim, por exemplo, você pode finalizá-lo com: `com.au` e `gov.au` para organizações comerciais e governamentais na Austrália; note que isto não é uma regra geral, as organizações de domínio atuais dependem da autoridade na escolha de nomes de cada domínio. Quando o endereço não especifica o domínio principal, como o endereço `www.unicamp.br`, isto quer dizer que é uma organização acadêmica.

O próximo nível da divisão representa o nome da organização. Subdomínios futuros variam em natureza, freqüentemente o próximo nível do sub-domínio é baseado na estrutura departamental da organização mas ela pode ser baseada em qualquer critério considerado razoável e significantes pelos administradores de rede para a organização.

A porção mais a esquerda do nome é sempre o nome único da máquina chamado *hostname*, a porção do nome a direita do hostname é chamado *nome de domínio* e o nome completo é chamado *nome do domínio completamente qualificado* (*Fully Qualified Domain Name*).

Usando o computador `www.debian.org.br` como exemplo:

- `br` - País onde o computador se encontra
- `org` - Domínio principal
- `debian` - Nome de Domínio
- `www` - Nome do computador

A localização do computador `www.debian.org.br` através de servidores DNS na Internet obedece exatamente a seqüência de procura acima. Os administradores do domínio `debian.org.br` podem cadastrar quantos sub-domínios e computadores quiserem (como `www.non-us.debian.org.br` ou `cvs.debian.org.br`).

Arquivos de configuração usados na resolução de nomes

Abaixo a descrição dos arquivos usados no processo de resolver um nome no sistema **GNU/Linux**.

/etc/resolv.conf

O `/etc/resolv.conf` é o arquivo de configuração principal do código do resolvidor de nomes. Seu formato é um arquivo texto simples com um parâmetro por linha e o endereço de servidores DNS externos são especificados nele. Existem três palavras chaves normalmente usadas que são:

<code>domain</code>	Especifica o nome do domínio local.
<code>search</code>	Especifica uma lista de nomes de domínio alternativos ao procurar por um computador, separados por espaços. A linha <code>search</code> pode conter no máximo 6 domínios ou 256 caracteres.
<code>nameserver</code>	Especifica o endereço IP de um servidor de nomes de domínio para resolução de nomes. Pode ser usado várias vezes.

Como exemplo, o `/etc/resolv.conf` se parece com isto:

```
domain maths.wu.edu.au
search maths.wu.edu.au wu.edu.au
nameserver 192.168.10.1
nameserver 192.168.12.1
```

Este exemplo especifica que o nome de domínio a adicionar ao nome não qualificado (i.e. hostnames sem o domínio) é `maths.wu.edu.au` e que se o computador não for encontrado naquele domínio então a procura segue para o domínio `wu.edu.au` diretamente. Duas linhas de nomes de servidores foram especificadas, cada uma pode ser chamada pelo código resolvidor de nomes para resolver o nome.

/etc/host.conf

O arquivo `/etc/host.conf` é o local onde é possível configurar alguns itens que gerenciam o código do resolvidor de nomes. O formato deste arquivo é descrito em detalhes na página de manual `resolv`. Em quase todas as situações, o exemplo seguinte funcionará:

```
order hosts,bind
multi on
```

Este arquivo de configuração diz ao resolvidor de nomes para checar o arquivo `/etc/hosts` (parâmetro `hosts`) antes de tentar verificar um *servidor de nomes* (parâmetro `bind`) e retornar um endereço IP válido para o computador procurado e `multi on` retornará todos os endereços IP resolvidos no arquivo `/etc/hosts` ao invés do primeiro.

Os seguintes parâmetros podem ser adicionados para evitar ataques de IP spoofing:

```
nospoof on
spoofalert on
```

O parâmetro `nospoof on` ativa a resolução reversa do nome da biblioteca `resolv` (para checar se o endereço pertence realmente àquele nome) e o `spoofalert on` registra falhas desta operação no **syslog**.

/etc/hosts

O arquivo `/etc/hosts` faz o relacionamento entre um nome de computador e endereço IP local. Recomendado para IPs constantemente acessados e para colocação de endereços de virtual hosts (quando deseja referir pelo nome ao invés de IP). A inclusão de um computador neste arquivo dispensa a consulta de um servidor de nomes para obter um endereço IP, sendo muito útil para máquinas que são acessadas frequentemente. A desvantagem de fazer isto é que você mesmo precisará manter este arquivo atualizado e se o endereço IP de algum computador for modificado, esta alteração deverá ser feita em cada um dos arquivos `hosts` das máquinas da rede. Em um sistema bem gerenciado, os únicos endereços de computadores que aparecerão neste arquivo serão da interface loopback e os nomes de computadores.

```
# /etc/hosts
127.0.0.1      localhost loopback
192.168.0.1   maquina.dominio.com.br
```

Você pode especificar mais que um nome de computador por linha como demonstrada pela primeira linha, a que identifica a interface loopback. Certifique-se de que a entrada do nome de domínio neste arquivo aponta para a interface de rede e não para a interface loopback, ou terá problema com o comportamento de alguns serviços.

OBS: Caso encontre problemas de lentidão para resolver nomes e até para executar os aplicativos (como o `mc`, etc), verifique se existem erros neste arquivo de configuração.

Estes sintomas se confundem com erros de memória ou outro erro qualquer de configuração de hardware, e somem quando a interface de rede é desativada (a com o IP não loopback). Isto é causados somente pela má configuração do arquivo `/etc/hosts`. O bom funcionamento do **Unix** depende da boa atenção do administrador de sistemas para configurar os detalhes de seu servidor.

/etc/networks

O arquivo `/etc/networks` tem uma função similar ao arquivo `/etc/hosts`. Ele contém um banco de dados simples de nomes de redes contra endereços de redes. Seu formato se difere por dois campos por linha e seus campos são identificados como:

```
Nome_da_Rede      Endereço_da_Rede
```

Abaixo um exemplo de como se parece este arquivo:

```
loopnet    127.0.0.0
localnet   192.168.1.0
amprnet    44.0.0.0
```

Quando usar comandos como `route`, se um destino é uma rede e esta rede se encontra no arquivo `/etc/networks`, então o comando `route` mostrará o *nome da rede* ao invés de seu endereço.

Executando um servidor de nomes

Se você planeja executar um servidor de nomes, você pode fazer isto facilmente. Por favor veja o documento DNS-HOWTO e quaisquer documentos incluídos em sua versão do BIND (Berkeley Internet Name Domain).

Serviços de Rede

Serviços de rede é o que está disponível para ser acessado pelo usuário. No TCP/IP, cada serviço é associado a um número chamado *porta* que é onde o servidor espera pelas conexões dos computadores clientes. Uma porta de rede pode se referenciada tanto pelo número como pelo nome do serviço.

Abaixo, alguns exemplos de portas padrões usadas em serviços TCP/IP:

- 21 - FTP (transferência de arquivos)
- 23 - Telnet (terminal virtual remoto)
- 25 - SmtP (envio de e-mails)
- 53 - DNS (resolvedor de nomes)
- 79 - Finger (detalhes sobre usuários do sistema)
- 80 - http (protocolo www - transferência de páginas Internet)
- 110 - Pop-3 (recebimento de mensagens)
- 119 - NNTP (usado por programas de notícias)

O arquivo padrão responsável pelo mapeamento do nome dos serviços e das portas mais utilizadas é o `/etc/services` (para detalhes sobre o seu formato, veja a “`/etc/services`”).

Serviços iniciados como Daemons de rede

Serviços de rede iniciados como *daemons* ficam residente o tempo todo na memória esperando que alguém se conecte (também chamado de *modo standalone*). Um exemplo de *daemon* é o servidor proxy **squid** e o servidor web **Apache** operando no modo *daemon*.

Alguns programas servidores oferecem a opção de serem executados como *daemons* ou através do *inetd*. É recomendável escolher *daemon* se o serviço for solicitado freqüentemente (como é o caso dos servidores web ou proxy).

Para verificar se um programa está rodando como *daemon*, basta digitar `ps ax` e procurar o nome do programa, em caso positivo ele é um *daemon*.

Normalmente os programas que são iniciados como *daemons* possuem seus próprios recursos de segurança/autenticação para decidir quem tem ou não permissão de se conectar.

Serviços iniciados através do inetd

Serviços iniciados pelo *inetd* são carregados para a memória somente quando são solicitados. O controle de quais serviços podem ser carregados e seus parâmetros, são feitos através do arquivo `/etc/inetd.conf`.

Um *daemon* chamado *inetd* lê as configurações deste arquivo e permanece residente na memória, esperando pela conexão dos clientes. Quando uma conexão é solicitada, o *daemon inetd* verifica as permissões de acesso nos arquivos `/etc/hosts.allow` e `/etc/hosts.deny` e carrega o programa servidor correspondente no arquivo `/etc/inetd.conf`. Um arquivo também importante neste processo é o `/etc/services` que faz o mapeamento das portas e nomes dos serviços.

Alguns programas servidores oferecem a opção de serem executados como *daemons* ou através do *inetd*. É recomendável escolher *inetd* se o serviço não for solicitado freqüentemente (como é o caso de servidores `ftp`, `telnet`, `talk`, etc).

`/etc/inetd.conf`

O arquivo `/etc/inetd.conf` é um arquivo de configuração para o *daemon* servidor *inetd*. Sua função é dizer ao **inetd** o que fazer quando receber uma requisição de conexão para um serviço em particular. Para cada serviço que deseja aceitar conexões, você precisa dizer ao *inetd* qual *daemon* servidor executar e como executa-lo.

Seu formato é também muito simples. É um arquivo texto com cada linha descrevendo um serviço que deseja oferecer. Qualquer texto em uma linha seguindo uma “#” é ignorada e considerada um comentário. Cada linha contém sete campos separados por qualquer número de espaços em branco (tab ou espaços). O formato geral é o seguinte:

```
serviço tipo_soquete proto opções.num usuário caminho_serv. opções_serv.
```

`serviço` É o serviço relevante a este arquivo de configuração pego do arquivo `/etc/services`.

`tipo_soquete` Este campo descreve o tipo do soquete que este item utilizará, valores permitidos são: `stream`, `dgram`, `raw`, `rdm`, ou `seqpacket`. Isto é um pouco técnico de natureza, mas como uma regra geral, todos os serviços

baseados em *tcp* usam *stream* e todos os protocolos baseados em *udp* usam *dgram*. Somente alguns tipos de daemons especiais de servidores usam os outros valores.

protocolo O protocolo é considerado válido para esta item. Isto deve bater com um item apropriado no arquivo */etc/services* e tipicamente será *tcp* ou *udp*. Servidores baseados no Sun RPC (*Remote Procedure Call*), utilizam *rpc/tcp* ou *rpc/udp*.

opções Existem somente duas configurações para este campo. A configuração deste campo diz ao *inetd* se o programa servidor de rede libera o soquete após ele ser iniciado e então se *inetd* pode iniciar outra cópia na próxima requisição de conexão, ou se o *inetd* deve aguardar e assumir que qualquer servidor já em execução pegará a nova requisição de conexão.

Este é um pequeno truque de trabalho, mas como uma regra, todos os servidores *tcp* devem ter este parâmetro ajustado para *nowait* e a maior parte dos servidores *udp* deve tê-lo ajustado para *wait*. Foi alertado que existem algumas excessões a isto, assim deixo isto como exemplo se não estiver seguro.

O *número* especificado após o "." é opcional e define a quantidade máxima de vezes que o serviço poderá ser executado durante 1 minuto. Se o serviço for executado mais vezes do que este valor, ele será automaticamente desativado pelo *inetd* e uma mensagem será mostrada no log do sistema avisando sobre o fato.

Para reativar o serviço interrompido, reinicie o **inetd** com: `killall -HUP inetd`. O valor padrão é 40.

usuário Este campo descreve que conta de usuário usuário no arquivo */etc/passwd* será escolhida como *dono* do daemon de rede quando este for iniciado. Isto é muito útil se você deseja diminuir os riscos de segurança. Você pode ajustar o usuário de qualquer item para o usuário *nobody*, assim se a segurança do servidor de redes é quebrada, a possibilidade de problemas é minimizada. Normalmente este campo é ajustado para *root*, porque muitos servidores requerem privilégios de usuário *root* para funcionarem corretamente.

caminho_servidor Este campo é o caminho para o programa servidor atual que será executado.

argumentos_servidor Este campo inclui o resto da linha e é opcional. Você pode colocar neste campo qualquer argumento da linha de comando que deseje passar para o daemon servidor quando for iniciado.

Uma dica que pode aumentar significativamente a segurança de seu sistema é comentar (colocar uma #no início da linha) os serviços que não serão utilizados.

Abaixo um modelo de arquivo */etc/inetd.conf* usado em sistemas **Debian**:

```
# /etc/inetd.conf: veja inetd(8) para mais detalhes.
#
# Banco de Dados de configurações do servidor Internet
#
#
```

```

# Linhas iniciando com "#:LABEL:" ou "#<off>#" não devem
# ser alteradas a não ser que saiba o que está fazendo!
#
#
# Os pacotes devem modificar este arquivo usando update-inetd(8)
#
# <nome_serviço> <tipo_soquete> <proto> <opções> <usuário> <caminho_servidor> <arg
#
#:INTERNO: Serviços internos
#echo          stream  tcp  nowait  root    internal
#echo          dgram   udp   wait   root    internal
#chargen      stream  tcp  nowait  root    internal
#chargen      dgram   udp   wait   root    internal
#discard      stream  tcp  nowait  root    internal
#discard      dgram   udp   wait   root    internal
#daytime      stream  tcp  nowait  root    internal
#daytime      dgram   udp   wait   root    internal
time          stream  tcp  nowait  root    internal
#time         dgram   udp   wait   root    internal

#:PADRÕES: Estes são serviços padrões.

#:BSD: Shell, login, exec e talk são protocolos BSD.
#shell        stream  tcp  nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rshd
#login        stream  tcp  nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogind
#exec         stream  tcp  nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
talk          dgram   udp   wait.10  nobody.tty /usr/sbin/tcpd  /usr/sbin/in.ta
ntalk         dgram   udp   wait.10  nobody.tty /usr/sbin/tcpd  /usr/sbin/in.nt

#:MAIL: Mail, news e serviços uucp.
smtp         stream  tcp  nowait.60 mail /usr/sbin/exim  exim -bs

#:INFO: Serviços informativos

#:BOOT: O serviço Tftp é oferecido primariamente para a inicialização. Alguns site
# o executam somente em máquinas atuando como "servidores de inicialização".

#:RPC: Serviços baseados em RPC

#:HAM-RADIO: serviços de rádio amador

#:OTHER: Outros serviços

```

Segurança da Rede e controle de Acesso

Deixe-me iniciar esta seção lhe alertando que a segurança da rede em sua máquina e ataques maliciosos são uma arte complexa. Uma regra importante é: "Não ofereça serviços de rede que não deseja utilizar".

Muitas distribuições vem configuradas com vários tipos de serviços que são iniciados automaticamente. Para melhorar, mesmo que insignificamente, o nível de segurança em seu sistema você deve editar se arquivo `/etc/inetd.conf` e comentar (colocar uma "#") as linhas que contém serviços que não utiliza.

Bons candidatos são serviços tais como: shell, login, exec, uucp, ftp e serviços de informação tais como finger, netstat e sysstat.

Existem todos os tipos de mecanismos de segurança e controle de acesso, eu descreverei os mais importantes deles.

/etc/ftpusers

O arquivo `/etc/ftpusers` é um mecanismo simples que lhe permite bloquear a conexão de certos usuários via *ftp*. O arquivo `/etc/ftpusers` é lido pelo programa daemon *ftp* (*ftpd*) quando um pedido de conexão é recebido. O arquivo é uma lista simples de usuários que não tem permissão de se conectar. Ele se parece com:

```
# /etc/ftpusers - login de usuários bloqueados via ftp
root
uucp
bin
mail
```

/etc/securetty

O arquivo `/etc/securetty` lhe permite especificar que dispositivos *tty* que o usuário *root* pode se conectar. O arquivo `/etc/securetty` é lido pelo programa *login* (normalmente `/bin/login`). Seu formato é uma lista de dispositivos *tty* onde a conexão é permitida, em todos os outros, a entrada do usuário *root* é bloqueada.

```
# /etc/securetty - terminais que o usuário root pode se conectar
tty1
tty2
tty3
tty4
```

O mecanismo de controle de acessos tcpd

O programa **tcpd** que você deve ter visto listado no mesmo arquivo `/etc/inetd.conf`, oferece mecanismos de registro e controle de acesso para os serviços que esta configurado para proteger. Ele é um tipo de firewall simples e fácil de configurar que pode evitar tipos indesejados de ataques e registrar possíveis tentativas de invasão.

Quando é executado pelo programa *inetd*, ele lê dos arquivos contendo regras de acesso e permite ou bloqueia o acesso ao servidor protegendo adequadamente.

Ele procura nos arquivos de regras até que uma regra confira. Se nenhuma regra conferir, então ele assume que o acesso deve ser permitido a qualquer um. Os arquivos que ele procura em seqüência são: `/etc/hosts.allow` e `/etc/hosts.deny`. Eu descreverei cada um destes arquivos separadamente.

Para uma descrição completa desta facilidade, você deve verificar a página de manual apropriada (`hosts_access` (5) é um bom ponto de partida).

/etc/hosts.allow

O arquivo `/etc/hosts.allow` é um arquivo de configuração do programa `/usr/sbin/tcpd`. O arquivo `hosts.allow` contém regras descrevendo que *hosts* tem permissão de acessar um serviço em sua máquina.

O formato do arquivo é muito simples:

```
# /etc/hosts.allow
#
# lista de serviços: lista de hosts : comando
```

lista de serviços É uma lista de nomes de serviços separados por vírgula que esta regra se aplica. Exemplos de nomes de serviços são: `ftpd`, `telnetd` e `fingerd`.

lista de hosts É uma lista de nomes de hosts separada por vírgula. Você também pode usar endereços IP's aqui. Adicionalmente, você pode especificar nomes de computadores ou endereço IP usando caracteres coringas para atingir grupos de hosts.

Exemplos incluem: `gw.vk2ktj.ampr.org` para conferir com um endereço de computador específico, `.uts.edu.au` para atingir qualquer endereço de computador finalizando com aquele string. Use `200.200.200.` para conferir com qualquer endereço IP iniciando com estes dígitos. Existem alguns parâmetros especiais para simplificar a configuração, alguns destes são: `ALL` atinge todos endereços, `LOCAL` atinge qualquer computador que não contém um "." (ie. está no mesmo domínio de sua máquina) e `PARANOID` atinge qualquer computador que o nome não confere com seu endereço (falsificação de nome). Existe também um último parâmetro que é também útil: o parâmetro `EXCEPT` lhe permite fazer uma lista de exceções. Isto será coberto em um exemplo adiante.

comando É um parâmetro opcional. Este parâmetro é o caminho completo de um comando que deverá ser executado toda a vez que esta regra conferir. Ele pode executar um comando para tentar identificar quem esta conectado pelo host remoto, ou gerar uma mensagem via E-Mail ou algum outro alerta para um administrador de rede que alguém está tentando se conectar.

Existem um número de expansões que podem ser incluídas, alguns exemplos comuns são: `%h` expande o endereço do computador que está conectado ou endereço se ele não possuir um nome, `%d` o nome do daemon sendo chamado.

Se o computador tiver permissão de acessar um serviço através do `/etc/hosts.allow`, então o `/etc/hosts.deny` não será consultado e o acesso será permitido.

Como exemplo:

```
# /etc/hosts.allow
#
# Permite que qualquer um envie e-mails
in.smtpd: ALL
# Permitir telnet e ftp somente para hosts locais e myhost.athome.org.au
in.telnetd, in.ftpd: LOCAL, myhost.athome.org.au
# Permitir finger para qualquer um mas manter um registro de quem é
in.fingerd: ALL: (finger %@h | mail -s "finger from %h" root)
```

Qualquer modificação no arquivo `/etc/hosts.allow` entrará em ação após reiniciar o daemon `inetd`. Isto pode ser feito com o comando `kill -HUP [pid do inetd]`, o `pid` do `inetd` pode ser obtido com o comando `ps ax|grep inetd`.

/etc/hosts.deny

O arquivo `/etc/hosts.deny` é um arquivo de configuração das regras descrevendo quais computadores não tem a permissão de acessar um serviço em sua máquina.

Um modelo simples deste arquivo se parece com isto:

```
# /etc/hosts.deny
#
# Bloqueia o acesso de computadores com endereços suspeitos
ALL: PARANOID
#
# Bloqueia todos os computadores
ALL: ALL
```

A entrada `PARANOID` é realmente redundante porque a outra entrada nega tudo. Qualquer uma destas linhas pode fazer uma segurança padrão dependendo de seu requerimento em particular.

Tendo um padrão `ALL: ALL` no arquivo `/etc/hosts.deny` e então ativando especificamente os serviços e permitindo computadores que você deseja no arquivo `/etc/hosts.allow` é a configuração mais segura.

Qualquer modificação no arquivo `/etc/hosts.deny` entrará em ação após reiniciar o daemon `inetd`. Isto pode ser feito com o comando `kill -HUP [pid do inetd]`, o `pid` do `inetd` pode ser obtido com o comando `ps ax|grep inetd`.

/etc/hosts.equiv e /etc/shosts.equiv

O arquivo `/etc/hosts.equiv` é usado para garantir/bloquear certos computadores e usuários o direito de acesso aos serviços "r*" (rsh, rexec, rcp, etc) sem precisar fornecer uma senha. O `/etc/shosts.equiv` é equivalente mas é lido somente pelo serviço ssh. Esta função é útil em um ambiente seguro onde você controla todas as máquinas, mesmo assim isto é um perigo de segurança (veja nas observações). O formato deste arquivo é o seguinte:

```
#Acesso  Máquina                               Usuário
-        maquina2.dominio.com.br             usuario2
-        maquina4.dominio.com.br             usuario2
+        maquina1.dominio.com.br             +@usuarios
```

O primeiro campo especifica se o acesso será permitido ou negado caso o segundo e terceiro campo confirmem. Por razões de segurança deve ser especificado o FQDN no caso de nomes de máquinas. Grupos de rede podem ser especificados usando a sintaxe "+@grupo".

Para aumentar a segurança, não use este mecanismo e encoraje seus usuários a também não usar o arquivo `.rhosts`.

ATENÇÃO O uso do sinal "+" sozinho significa permitir acesso livre a qualquer pessoa de qualquer lugar. Se este mecanismo for mesmo necessário, tenha muita atenção na especificação de seus campos.

Evita também A TODO CUSTO uso de nomes de usuários (a não ser para negar o acesso), pois é fácil forjar o login, entrar no sistema tomar conta de processos (como por exemplo do servidor **Apache** rodando sob o usuário `www-data` ou até mesmo o **root**), causando enormes estragos.

Verificando a segurança do TCPD e a sintaxe dos arquivos

O utilitário **tcpdchk** é útil para verificar problemas nos arquivos `hosts.allow` e `hosts.deny`. Quando é executado ele verifica a sintaxe destes arquivos e relata problemas, caso eles existam.

Outro utilitário útil é o **tcpdmatch**, o que ele faz é permitir que você simule a tentativa de conexões ao seu sistema e observar se ela será permitida ou bloqueada pelos arquivos `hosts.allow` e `hosts.deny`.

É importante mostrar na prática como o **tcpdmatch** funciona através de um exemplo simulando um teste simples em um sistema com a configuração padrão de acesso restrito:

- O arquivo `hosts.allow` contém as seguintes linhas:

```
ALL: 127.0.0.1
in.talkd, in.ntalkd: ALL
in.fingerd: 192.168.1. EXCEPT 192.168.1.30
```

A primeira linha permite o loopback (127.0.0.1) acessar qualquer serviço TCP/UDP em nosso computador, a segunda linha permite qualquer um acessar os servidor TALK (nós desejamos que o sistema nos avise quando alguém desejar conversar) e a terceira somente permite enviar dados do **finger** para computadores dentro de nossa rede privada (exceto para 192.168.1.30).

- O arquivo `hosts.deny` contém a seguinte linha:

```
ALL: ALL
```

Qualquer outra conexão será explicitamente derrubada.

Vamos aos testes, digitando: `"tcpdmatch in.fingerd 127.0.0.1"` (verificar se o endereço 127.0.0.1 tem acesso ao finger):

```
client: address 127.0.0.1
server: process in.fingerd
matched: /etc/hosts.allow line 1
access: granted
```

Ok, temos acesso garantido com especificado pela linha 1 do `hosts.allow` (a primeira linha que confere é usada). Agora `"tcpdmatch in.fingerd 192.168.1.29"`:

```
client: address 192.168.1.29
server: process in.fingerd
matched: /etc/hosts.allow line 3
access: granted
```

O acesso foi permitido através da linha 3 do `hosts.allow`. Agora `"tcpdmatch in.fingerd 192.168.1.29"`:

```
client: address 192.168.1.30
server: process in.fingerd
matched: /etc/hosts.deny line 1
access: denied
```

O que aconteceu? como a linha 2 do `hosts.allow` permite o acesso a todos os computadores 192.168.1.* exceto 192.168.1.30, ela não bateu, então o processamento partiu para o `hosts.deny` que nega todos os serviços para qualquer endereço. Agora um último exemplo: "tcpdmatch in.talkd www.debian.org"

```
client:   address  www.debian.org
server:   process  in.talkd
matched:  /etc/hosts.allow line 2
access:   granted
```

Ok, na linha 2 qualquer computador pode te chamar para conversar via talk na rede, mas para o endereço DNS conferir com um IP especificado, o **GNU/Linux** faz a resolução DNS, convertendo o endereço para IP e verificando se ele possui acesso.

No lugar do endereço também pode ser usado a forma `daemon@computador` ou `cliente@computador` para verificar respectivamente o acesso de daemons e cliente de determinados computadores aos serviços da rede.

Como pode ver o TCPD ajuda a aumentar a segurança do seu sistema, mas não confie nele além do uso em um sistema simples, é necessário o uso de um firewall verdadeiro para controlar minuciosamente a segurança do seu sistema e dos pacotes que atravessam os protocolos, roteamento e as interfaces de rede. Se este for o caso aprenda a trabalhar a fundo com firewalls e implemente a segurança da sua rede da forma que melhor planejar.

Firewall

Dentre todos os métodos de segurança, o *Firewall* é o mais seguro. A função do Firewall é bloquear determinados tipos de tráfego de um endereço ou para uma porta local ou permitir o acesso de determinados usuários mas bloquear outros, bloquear a falsificação de endereços, redirecionar tráfego da rede, ping da morte, etc.

A implementação de um bom firewall dependerá da experiência, conhecimentos de rede (protocolos, roteamento, interfaces, endereçamento, masquerade, etc), da rede local, e sistema em geral do Administrador de redes, a segurança de sua rede e seus dados dependem da escolha do profissional correto, que entenda a fundo o TCP/IP, roteamento, protocolos, serviços e outros assuntos ligados a rede.

Freqüentemente tem se ouvido falar de empresas que tiveram seus sistemas invadidos, em parte isto é devido a escolha do sistema operacional indevido mas na maioria das vezes o motivo é a falta de investimento da empresa em políticas de segurança, que algumas simplesmente consideram a segurança de seus dados e sigilo interno como uma *despesa a mais*.

Um bom firewall que recomendo é o **ipchains**, **Sinus** e o **TIS**. Particularmente gosto muito de usar o **ipchains** e o **Sinus** e é possível fazer coisas inimagináveis programando scripts para interagirem com estes programas...

Outros arquivos de configuração relacionados com a rede

/etc/services

O arquivo `/etc/services` é um banco de dados simples que associa um nome amigável a humanos a uma porta de serviço amigável a máquinas. É um arquivo texto de formato muito simples, cada linha

representa um item no banco de dados. Cada item é dividido em três campos separados por qualquer número de espaços em branco (tab ou espaços). Os campos são:

nome	porta/protocolo	apelido	# comentário
name	Uma palavra simples que representa o nome do serviço sendo descrito.		
porta/protocolo	Este campo é dividido em dois sub-campos.		
	<ul style="list-style-type: none"> • <i>porta</i> - Um número que especifica o número da porta em que o serviço estará disponível. Muitos dos serviços comuns tem designados um número de serviço. Estes estão descritos no RFC-1340. • <i>protocolo</i> - Este sub-campo pode ser ajustado para <i>tcp</i> ou <i>udp</i>. É importante notar que o item <i>18/tcp</i> é muito diferente do item <i>18/udp</i> e que não existe razão técnica porque o mesmo serviço precisa existir em ambos. Normalmente o senso comum prevalece e que somente se um serviço esta disponível em ambos os protocolos <i>tcp</i> e <i>udp</i>, você precisará especificar ambos. 		
apelidos	Outros nomes podem ser usados para se referir a entrada deste serviço.		
comentário	Qualquer texto aparecendo em uma linha após um caracter "#" é ignorado e tratado como comentário.		

/etc/protocols

O arquivo `/etc/protocols` é um banco de dados que mapeia números de identificação de protocolos novamente em nomes de protocolos. Isto é usado por programadores para permiti-los especificar protocolos por nomes em seus programas e também por alguns programas tal como `tcpdump` permitindo-os mostrar *nomes* ao invés de *números* em sua saída. A sintaxe geral deste arquivo é:

```
nomeprotocolo número apelidos
```

Camadas de Rede

São organizações do protocolo TCP/IP que visam organizar e simplificar seu padrão e implementação pelos desenvolvedores.

- Um *padrão* TCP é o conjunto de regras que devem ser seguidas para garantir a homogeneidade da comunicação entre diversos sistemas de diversos fabricantes (por exemplo, *Mac* com *Windows*, *Windows* com *Linux*, etc.).
- A *implementação* é o código escrito por cada desenvolvedor para integração ao sistema operacional seguindo as regras do padrão para garantir a comunicação entre as máquinas, portanto, a *implementação* do protocolo TCP varia de fabricante para fabricante.

Existem dois tipos de padrões TCP: *Darpa* e *OSI*. O padrão *Darpa* é dividido em 4 camadas e ainda é o padrão atualmente utilizado. O padrão *OSI* é mais recente, dividido em 7 camadas, mas ainda não se tornou um padrão como o *Darpa*.

Segue abaixo os padrões e a descrição de cada uma das camadas:

Darpa • Aplicação - *www*, *ftp*, *dns*, etc. Fazem interface com as aplicações do sistema.

- Transporte - Protocolo *tcp* e *udp*. Cuidam da parte de transporte dos dados do sistema.
- Rede - *IP, icmp, igmp, arp*. Cuida de levar o pacote para seu destino (rotas) e condições de transmissão.
- Interface de Rede - *Ethernet, FDDI, Token Ring*. Define qual o método que a mensagem transmitida será encapsulada para envio ao seu destino.

Apesar dos padrões *Darpa* e *OSI*, o protocolo *TCP/IP* é oficialmente independente destas camadas.

RFCs de referência sobre protocolos de rede

Como referência de pesquisa, segue abaixo a listagem de números de RFCs para protocolos de rede mais utilizados:

IP <http://www.rfc-editor.org/rfc/rfc791.txt>

ICMP <http://www.rfc-editor.org/rfc/rfc792.txt>

TCP <http://www.rfc-editor.org/rfc/rfc793.txt>

UDP <http://www.rfc-editor.org/rfc/rfc768.txt>

Capítulo 5. Configurações especiais de Rede

Este capítulo descreve alguns tipos de configurações que podem ser feitas em rede utilizando os recursos disponíveis do **Linux**. Aqui não estão todas as aplicações, pois o sistema é bastante flexível e o próprio time de desenvolvimento do kernel não demonstrou limitações quanto as formas de se construir uma rede :-)

IP Alias

Este recurso permite configurar uma interface de rede para responder por um ou mais IPs, que não precisam pertencer a mesma faixa. Para usuários externos, a impressão é que a rede tem "muitas" máquinas, quando na realidade apenas uma responde por todos estes endereços virtuais. Podemos citar algumas utilizações úteis deste recurso:

- Simular uma rede com diversas máquinas
- Construir virtual hosts baseados em IP
- Definir endereçamentos secundários para fins de análise e depuração de pacotes (principalmente como armadilhas para trojans)
- Colocação de serviços com operação restritas a interfaces em funcionamento através de faixas específicas usando as configurações da interface virtual
- Transição de IP de servidores de forma transparente
- Entre muitas outras. A idéia aqui é mostrar a simplicidade de se configurar este recurso e entender o processo, que é bastante simples.

Para configurar o recurso de *IP Alias* é necessário apenas que a opção *IP Aliasing Support* seja habilitada no kernel (como módulo ou embutida). Em nosso exemplo abaixo, temos uma rede com a interface `eth0` configurada com o IP `192.168.1.1` (classe C privada) e queremos adicionar uma interface virtual que atenda pelo IP `172.16.0.1` (classe B privada) e depois seguir os seguintes passos:

1. Ative a interface de rede com **ifconfig** ou **ifup** (caso esteja usando a **Debian**).
2. Crie uma interface virtual usando o comando `ifconfig eth0:0 172.16.0.1`. Isto criará uma nova interface chamada `eth0:0` que passará a responder pelo IP `172.6.0.1`. É permitido o uso de nomes para especificar a interface virtual, como: `eth0:redel`, `eth0:rede2`, `eth0:escritório`.
3. Digite `ifconfig` para ver as configurações de sua nova interface de rede. Use o `ping` também para vê-la: `ping 172.16.0.1`.

```
eth0      Encapsulamento do Link: Ethernet  Endereço de HW 00:80:AE:B3:AA:AA
inet end.: 192.168.1.1  Bcast:192.168.1.255  Masc:255.255.255.0
UP BROADCASTRUNNING MULTICAST MTU:1500 Métrica:1
RX packets:979 errors:0 dropped:0 overruns:0 frame:0
TX packets:1228 errors:0 dropped:0 overruns:0 carrier:0
colisões:1 txqueuelen:100
RX bytes:71516 (69.8 Kb)  TX bytes:1146031 (1.0 Mb)
IRQ:10  Endereço de E/S:0x300
```

```
eth0:0 Encapsulamento do Link: Ethernet Endereço de HW 00:80:AE:B3:AA:AA
inet end.: 192.168.1.10 Bcast:192.168.1.255 Masc:255.255.255.0
UP BROADCASTRUNNING MULTICAST MTU:1500 Métrica:1
IRQ:10 Endereço de E/S:0x300
```

Note que o MAC Address da placa eth0 e eth0:0 são o mesmo, indicando que a mesma interface atende ambos os IPs.

4. Se necessário ajuste as rotas ou gateway com o comando **route** (veja “Configurando uma rota no Linux”).

Para desativar uma interface de rede virtual, utilize a sintaxe: `ifconfig eth0:0 down` ou `ifdown eth0:0` (caso esteja usando a **Debian**).

Se o teste com o **ping** não funcionar, verifique se possui o suporte a *IP Alias* no kernel, se o módulo precisa ser carregado manualmente (caso seu kernel não esteja compilado com o **kmod**) ou se existe um firewall restritivo bloqueando seu IP.

Na distribuição **Debian** a configuração de uma interface virtual pode ser feita de forma idêntica a interfaces estáticas padrão:

```
auto eth0
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255

auto eth0:0
iface eth0:0 inet static
address 172.16.0.1
netmask 255.255.0.0
network 172.16.0.1
broadcast 172.16.255.255
```

OBS1: Quando você desativa uma interface física (eth0), todas as interfaces virtuais também são desativadas.

OBS2: Caso utilize um firewall (principalmente com a política padrão permissiva), esteja atento as modificações que precisa realizar para não comprometer a segurança de sua máquina. Caso tenha dados considerados seguros em sua máquina e esteja em dúvida sobre as implicações de segurança do *IP Alias* em sua máquina, consulte seu administrador de redes.

OBS3: Note que somente os 4 primeiros caracteres serão mostrados na saída do **ifconfig**, desta forma procure utilizar no máximo esta quantidade de caracteres para evitar problemas durante uma futura administração do servidor, no caso de esquecimento do nome completo da interface virtual).

Bridge

Uma *bridge* é uma interface de rede lógica composta por uma ou mais interfaces de rede física operando em nível 2 (enviando pacotes através de *MAC addresses*, veja “Camadas de Rede”).

Sua operação é transparente na rede, podendo ser usada como um switch/firewall, estação de monitoração, etc. Aqui descreverei como montar uma bridge simples e uma aplicação de firewall simples. As

possibilidades são diversas e uma configuração bem feita pode detectar ataques, protocolos desconhecidos até vírus complexos de rede.

Requerimentos para a Instalação

É necessário um dos seguintes requerimentos para se montar uma bridge:

- Kernel com suporte a bridge ativado (na configuração de rede)
- O pacote instalado.
- patch `bridge-nf` se desejar usar o netfilter com as interfaces de entrada e saída (como antes de usar a bridge) ao invés de controlar o tráfego apenas pela interface criada pela bridge.

Ative a opção `802.1d Ethernet Bridging` na seção `Networking Options`, recompile e instale seu novo kernel. Caso tenha aplicado o patch `bridge nf`, aparecerá uma sub opção chamada `netfilter (firewalling) support` que permitirá que o firewall trabalhe com as interfaces físicas ao invés de somente através da interface virtual criada pela bridge.

OBS: O patch `bridge nf` viola a RFC de bridges. Mesmo assim ela é a única opção em muitas aplicações, principalmente quando se deseja controlar o tráfego que atravessam as interfaces. Após isto instale o pacote , ele possui os utilitários necessários para ativar, configurar e monitorar o funcionamento de sua bridge.

Não é necessária ativação do `ip_forward` para o funcionamento da bridge, uma vez que ela funcionará como uma interface lógica que reúne interfaces de rede físicas.

Configuração da bridge

Nos exemplos abaixo, eu assumirei a utilização do nome de dispositivo `br0` para se referir a bridge no sistema. Siga estes passos para configurar uma bridge em sistemas **Debian**:

- Primeiro, desative os blocos no arquivo `/etc/network/interfaces` que configuram as interfaces que serão usadas na bridge (por exemplo, `eth0` e `eth1`). Elas podem ser comentadas, removidas, ou você poderá comentar a linha `auto eth0` e `auto eth1` para que ele não ative automaticamente estas interfaces com o `ifup -a` (executado durante a inicialização). Desta forma, a inicialização destas interfaces poderá somente ser feita manualmente.

```
auto br0
iface br0 inet static
    address 192.168.1.2
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    bridge_ports eth0 eth1
```

Note que a interface virtual da bridge (`br0`) deve ser configurada com parâmetros válidos de interfaces (assim com uma interface de rede padrão). Note a adição da linha `bridge_ports` que indica que interfaces de rede serão usadas para fazer a bridge. Caso seja usado o parâmetro `all`, todas as interfaces físicas de rede serão usadas para fazer bridge (excluindo a `lo`).

- Execute o `ifdown -a` (para desativar as interfaces antigas).

- Execute o `ifup br0` para levantar as interface `br0`. O sistema poder demorar um pouco para levantar a bridge (as vezes até 40 segundos) mas isto é normal.

Pronto, você terá uma bridge simples já configurada e funcionando em seu sistema! As interfaces físicas serão configuradas com o IP 0.0.0.0 e estarão operando em modo promíscuo.

Configurações mais avançadas de bridge

A bridge permite ainda definir prioridade para utilização de interfaces, além de outras funcionalidades que lhe permitem ajustar a performance da máquina de acordo com sua rede. Um bom exemplo, é quando você deseja criar 2 bridges em uma mesma máquina envolvendo interfaces de rede específicas, uma atendendo a rede 192.168.0.x e outra a rede 192.168.1.x:

```
auto br0
iface br0 inet static
    address 192.168.0.2
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    bridge_ports eth0 eth1

auto br1
iface br1 inet static
    address 192.168.1.2
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.0.1
    bridge_ports eth2 eth3 eth4
```

No exemplo acima, as interfaces `eth0` e `eth1` fazem parte da bridge `br0` e as demais (`eth2`, `eth3` e `eth4`) da bridge `br1`.

```
bridge_ports eth2 eth3
bridge_bridgeprio 16385
bridge_portprio eth1 100
bridge_fd 5
```

Configuração manual da bridge

Internamente, o que o `ifup` faz é interpretar os parâmetros no arquivo de configuração e executar os comandos do pacote para ativar a interface da bridge. O utilitário responsável por este processo é o **brctl**. Será documentado aqui como ativar uma bridge através deste programa (que servirá para fazer uma bridge em qualquer sistema **Linux**).

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1

ifconfig eth0 0.0.0.0
```

```
ifconfig eth1 0.0.0.0
ifconfig br0 192.168.0.4
```

O comando acima ativa uma bridge simples, como o primeiro exemplo. Tenha certeza que as interfaces físicas de rede estão desativadas antes de executar este comando.

Outros parâmetros que podem ser usados com o **brctl**:

<code>setbridgeprio [bridge] [prioridade]</code>	Define a prioridade da bridge, o valor deve estar entre 0 e 65536 (16 bits). Valores menores definem uma prioridade maior.
<code>setfd [bridge] [tempo]</code>	Ajusta o delay da bridge especificada em [tempo] segundos.
<code>setmaxage [bridge] [tempo]</code>	Ajusta o tempo máximo de vida da bridge para [tempo] segundos.
<code>setportprio [bridge] [interface] [prioridade]</code>	Ajusta a prioridade da [interface] especificada na [bridge]. O valor de prioridade deve estar entre 0 e 255 (8 bits). Quanto menor o valor maior a prioridade. Isto é útil para otimizações o volume de tráfego em máquinas que possuem diversas interfaces configuradas fazendo parte da bridge.

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
brctl setportprio br0 eth0 50
brctl setportprio br0 eth1 80
brctl setfd br0 2
```

```
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
ifconfig br0 192.168.0.4
```

Usando o iptables para construir um firewall na máquina da bridge

A construção de um firewall em uma bridge não tem maiores segredos, basta referir-se a interface lógica da bridge para construir suas regras (tendo em mente como uma bridge funciona e como os pacotes atravessarão as interfaces).

Caso aplique o patch *bridge nf*, será possível referir-se as interfaces locais de rede e também a da bridge. Neste caso a interface da bridge será identificada como interface *IN* ou *OUT PHYSIN* e as interfaces físicas como *PHYSOUT*:

```
Oct 22 09:19:24 router kernel: IN=br0 PHYSIN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:d4
```

Mesmo que a bridge não necessite de *ip_forward* ativado para redirecionar os pacotes através das interfaces, isto será necessário para habilitar o uso do firewall para controlar o tráfego que atravessa as interfaces.

Filtrando pacotes não IP na bridge

Para fazer esta tarefa, utilize a ferramenta **ebtables** disponível em (<http://users.pandora.be/bart.de.schuymer/ebtables/>).

Conectando dois computadores usando a porta paralela

O **Linux** é bastante poderoso quando se trata de métodos para se conectar duas ou mais máquinas em rede. Uma brincadeira que é levada a sério é que qualquer coisa que ligue uma máquina a outra possui um controlador desenvolvido por alguém para fazer uma rede :)

Usando o `plip` (*Parallel Line Internet Protocol*) permite criar uma interface de rede para a porta paralela que utiliza todos os recursos de uma rede normal. Esta interface será identificada por `plip?`, onde ? é o número da porta paralela, recém configurada.

A rede via porta paralela pode atingir até 1Mb/s e mesmo esta velocidade parecer aparentemente baixa apresenta diversas vantagens por sua escalabilidade e pode lhe salvar em muitas situações de problemas. Algumas características deste tipo de rede:

- Pode ser configurado em qualquer máquina, pois sempre haverá uma porta paralela.
- É útil para fazer instalação de **Linux** em máquinas sem CD-ROM. No momento da instalação é preciso somente alternar para um console, executar os passos descritos aqui e continuar com o processo de instalação normal :)
- É uma boa solução quando as duas máquinas estão próximas
- O custo para montagem desta rede é extremamente baixo, bastando um cabo Lap Link Paralelo que custa no máximo R\$20,00 o de 1,5M ou se gosta de eletrônica, montar seu próprio cabo usando o esquema que descrevo em “Construindo um cabo LapLink Paralelo”.
- Você poderá fazer qualquer coisa que faria em uma rede normal (incluindo MASQUERADING, roteamento entre redes, etc) sendo bastante interessante para testes práticos dos exemplos do Foca Linux Avançado ;-)
- Ficará admirado com as capacidade de rede existente no **Linux** e feliz por ter colocado mais uma configuração em funcionamento :)

Agora, os contras da conexão via porta paralela:

- A porta paralela não estará disponível para ser usada em impressoras, conexão de câmeras.
- O cabo não pode ter mais de 4,5 metros. Acima dessa comprimento, você pode colocar sua controladora em risco além da perda de sinal. Por segurança, o tamanho recomendável é 2,5 metros.
- Quando toda a banda do cabo é utilizada, algumas CPUs se tornam extremamente lentas.

Para configurar uma conexão via cabo paralelo (`plip`) entre duas máquinas, vamos assumir que a primeira máquina terá o IP 192.168.1.1 e a segunda máquina 192.168.1.2:

1. Conecte o cabo Lap Link em cada uma das portas de impressora. Caso saiba fazer conexões eletrônicas ou goste do assunto, veja “Construindo um cabo LapLink Paralelo”.
2. Verifique se o seu kernel está compilado com o suporte a rede `plip`. Caso não esteja, a configuração da interface `plip` falhará no passo do **ifconfig**.
3. Se o sistema executa algum daemon de impressão, interrompa antes de usar a porta paralela. Alguns tipos de serviços de impressão interferem no funcionamento do `plip`.

4. Configure o módulo `parport_pc` passando o parâmetro `irq=7` (a IRQ que sua porta de impressora utiliza). Esta configuração é necessária pois em algumas máquinas isso faz que o `plip` não funcione ou aconteçam somente timeouts de transmissão.
5. Execute o comando `ifconfig plip0 192.168.1.1`. Verifique se a interface foi ativada com o comando `ifconfig plip0`.
6. Nesse ponto a interface está ativa, mas a nossa máquina não conhece nada sobre a rede ou como alcançar a máquina 192.168.1.2. Como a conexão é ponto a ponto, precisamos adicionar uma rota direta para esta máquina com o comando: `route add -host 192.168.1.2 plip0`.

Este comando diz para criar uma rota com o destino 192.168.1.2 usando a interface `plip0`.

7. Configure a outra máquina seguindo os passos acima, apenas invertendo os 2 endereços IPs usados.

Pronto, agora verifique se cada uma das máquinas se comunica com a outra usando o comando `ping 192.168.1.x`. Se ocorrer um erro de timeout na transmissão, leia atentamente os passos acima e refaça a configuração em ambas as máquinas. Ainda não funcionando, verifique se existe um firewall bloqueando os pacotes da nova interface e se o cabo Lap Link está em bom estado, o problema pode estar aí.

O número máximo de interfaces `plip?` está limitado ao número máximo suportado pela máquina. O padrão em sistemas padrão IBM/PC é de 3 (`plip0`, `plip1`, `plip2`).

Para desativar uma rede `plip`, utilize o comando `ifconfig plip0 down`, remova o módulo `plip` (`rmmmod plip`). Após isto, a porta paralela será liberada para uso por outros aplicativos.

Construindo um cabo LapLink Paralelo

Se você tem experiência com eletrônica, poderá construir seu próprio cabo LapLink Paralelo para fazer os testes desta seção. Os materiais necessários são:

- 2 Conectores DB25 macho
- 2 Capas para os conectores acima.
- Fio para ligação dos conectores (15 ligações). No meu caso utilizei 2 metros de um rolo de cabo SCSI de 50 vias para fazer as ligações, que é uma boa alternativa para manter o cabo bonito e os fios juntos.

Este é o conector macho DB25 (a tomada que liga no computador) visto por trás (minha namorada já disse que não sou bom em arte ASCII). Bom, não custa tentar de novo:

```

-----
13  \ o o o o o o o o o o o o o / 1
25  \ o o o o o o o o o o o o o / 14
-----

```

A figura acima mostra a posição dos pinos como referência para a soldagem dos terminais. A tabela abaixo mostra a ligação dos fios nos cabos das 2 pontas do cabo:

```

+-----+-----+
| Ponta 1 | Ponta 2 |
+-----+-----+
|     1   |     1   |

```

2	15
3	13
4	12
5	10
6	11
10	5
11	6
12	4
13	3
14	14
15	2
16	16
17	17
25	25

Conectando dois computadores usando a porta serial

Este método permite criar uma rede ponto a ponto usando a porta serial da máquina, que funcionará de forma semelhante a mostrada em “Conectando dois computadores usando a porta paralela”.

O método que irei descrever é bastante simples e utiliza o **slattach** e o protocolo *slip* para comunicação entre as duas máquinas, mas nada impede que seja usado o *ppp* para comunicação, apenas acrescentará um pouco mais de complexibilidade para esta configuração para obter o mesmo resultado.

Usando o método descrito, será criada uma interface chamada `s1?` (interface SLIP, onde ? é o número da interface recém configurada).

A rede via porta serial pode atingir em média 115.200kbps/s mas é prático quando não tem outras opções para fazer uma rede ponto a ponto. Segue algumas características deste tipo de rede:

- Pode ser configurado em qualquer máquina, pois sempre haverá uma porta serial disponível.
- É possível fazer a instalação de **Linux** em máquinas sem CD-ROM e acesso a rede, onde não é possível gerar disquetes para instalar o resto dos pacotes necessários, embora seja limitado a 11Kb/s. No momento da instalação é preciso somente alternar para um console, executar os passos descritos aqui e continuar com o processo de instalação normal :)
- É uma boa solução quando as duas máquinas até em ambientes próximos.
- O custo para montagem desta rede é extremamente baixo, bastando um cabo Lap Link Serial custa em média R\$20,00 o cabo de 4 metros. Se você também é um amante da eletrônica, estou descrevendo o esquema de montagem do cabo em “Construindo um cabo LapLink Serial”.
- Você poderá fazer qualquer coisa que faria em uma rede normal (incluindo roteamento entre redes, MASQUERADING, etc)
- É mais uma prova das capacidades de rede que é possível usando o **Linux**.

Agora, os contras da conexão via porta serial:

- A porta serial não estará disponível para ser usada para conexão de mouses, impressoras seriais, dispositivos eletrônicos e inteligentes, etc.

- O comprimento máximo do cabo é de 15 metros. Acima dessa comprimento, você pode colocar sua controladora em risco além da perda de sinal. Por segurança, o tamanho máximo recomendável é 13 metros

Para configurar uma conexão via cabo serial entre duas máquinas, vamos assumir que a primeira máquina terá o IP 192.168.2.1 e a segunda máquina 192.168.2.2:

1. Conecte o cabo Lap Link serial em cada uma das portas seriais.
2. Verifique se o seu kernel está compilado com o suporte a rede `slip` e também com suporte a `cslip` (`slip` compactado, que melhora a taxa de transferência dependendo dos dados sendo transmitidos). Caso não tenha o suporte a `slip`, você poderá usar o `ppp` nas duas pontas do link fazendo algumas adaptações para usar a interface `ppp?`, como é simples não será descrito neste guia :) (veja o manual do **slattach**)
3. Interrompa qualquer programa que esteja usando a porta serial.
4. Execute o comando `slattach -s 115200 /dev/ttyS1 &`. A função do **slattach** é associar uma interface de rede a um dispositivo, neste caso associamos o dispositivo `/dev/ttyS1` (segunda porta serial) a interface `s10` (verifique se a interface foi criada usando o comando `ifconfig s10`.

A opção `-p` especifica um protocolo alternativo para o **slattach**, o padrão é o `cslip`. Outros tipos disponíveis são `slip`, `adaptive ppp` e `kiss` (usado em conexões de rádio AX.25). Recomendo ver a página de manual do **slattach**.

5. Nesse ponto a interface está ativa, mas a nossa máquina não conhece nada sobre a rede ou como alcançar a máquina 192.168.2.2. Como a conexão é ponto a ponto, precisamos adicionar uma rota direta para esta máquina com o comando: `route add -host 192.168.2.2 s10`.

Este comando diz para criar uma rota com o destino 192.168.2.2 usando a interface `s10`.

6. Configure a outra máquina seguindo os passos acima, apenas invertendo os 2 endereços IPs usados.

Pronto, agora verifique se cada uma das máquinas se comunica com a outra usando o comando `ping 192.168.2.x`. Se ocorrer um erro, verifique os seguintes itens:

- Se as velocidade e o protocolo especificado em ambos os lados do link estão iguais.
- Se já existe um processo **slattach** rodando em segundo plano.
- Se existe um firewall bloqueando os pacotes da nova interface
- Se o cabo Lap Link serial está em bom estado.

O número máximo de interfaces `s1?` depende da quantidade de portas seriais da sua máquina. Caso utilize uma placa multi serial, o número máximo de conexões de rede se torna grande (mas isto é apenas para curiosidade, pois não compensa uma multi serial para ligar uma quantidade grande de máquinas a baixa velocidade).

Para derrubar a conexão, basta derrubar a interface serial com o `ifconfig s10 down`, dar um kill no daemon do **slattach** e remover o módulo `slip` e `cslip` com o comando `rmmmod`. Assim sua porta serial será liberada e poderá ser usada por outros aplicativos.

Construindo um cabo LapLink Serial

Se você é uma pessoa que sabe mexer com eletrônica, poderá construir seu próprio cabo LapLink serial para fazer os testes desta seção. Os materiais necessários são:

- 2 - Conectores seriais DB9 fêmea
- 2 - Capas para os conectores acima.
- Fios para ligação dos conectores. Uma forma que utilizei para montar este cabo foi aproveitar um carretel de cabo SCSI aproveitando 10 metros desfiando somente 9 dos 50 fios que acompanha o cabo (deixei um fio extra no caso de algum outro se romper).
- Ferro de solda e solda para as ligações.
- Concentração e paciência para a confecção correta dos cabos.

Este é o conector fêmea DB9 (tomada que liga na máquina) visto por trás (hora de mostrar novamente meu talento com arte ASCII :))

```

-----
1 \ o o o o o / 5
6 \ o o o o o / 9
-----

```

A figura acima mostra a posição dos pinos como referência para a soldagem dos terminais. A tabela abaixo mostra a ligação dos fios nos cabos das 2 pontas. Note que cada ponta pode ter a opção da serial de 9 ou 25 pinos (ou as duas):

Ponta 1		Ponta 2	
9	25	25	9
5	7	7	5
3	2	3	2
7	4	5	8
6	6	20	4
2	3	2	3
8	5	4	7
4	20	6	6

Capítulo 6. Arquivos e daemons de Log

A atividade dos programas são registradas em arquivos localizados em `/var/log`. Estes arquivos de registros são chamados de *logs* e contém a data, hora e a mensagem emitida pelo programa (violações do sistema, mensagens de erro, alerta e outros eventos) entre outros campos. Enfim, muitos detalhes úteis ao administrador tanto para acompanhar o funcionamento do seu sistema, comportamento dos programas ou ajudar na solução e prevenção de problemas.

Alguns programas como o **Apache**, **exim**, **ircd** e **squid** criam diversos arquivos de log e por este motivo estes são organizados em sub-diretórios (a mesma técnica é usada nos arquivos de configuração em `/etc`, conforme a padrão FHS atual).

Formato do arquivo de log

Um arquivo de log é normalmente composto pelos seguintes campos:

Data | Hora | Máquina | daemon | mensagem

O campo *máquina* é o nome do computador que registrou a mensagem (a máquina pode atuar como um servidor de logs registrando mensagens de diversos computadores em sua rede). O campo *daemon* indica qual programa gravou a *mensagem*.

O uso dos utilitários do console pode ajudar muito na pesquisa e monitoração dos logs, por exemplo, para obter todas as mensagens do daemon **kernel** da estação de trabalho `wrk1`, eliminando os campos "wrk1" e "kernel":

```
cat /var/log/*|grep 'wrk1'|grep 'kernel'|awk '{print $1 $2 $3 $6 $7 $8 $9 $10 $11
```

Os parâmetros "\$1", "\$2" do comando **awk** indica que campos serão listados, (omitimos \$4 e \$5 que são respectivamente "wrk1" e "kernel"). Um bom utilitário para monitoração de logs está documentado em "logcheck".

Daemons de log do sistema

Os daemons de log do sistema registram as mensagens de saída do kernel (**klogd**) e sistema (**syslogd**) nos arquivos em `/var/log`.

A classificação de qual arquivo em `/var/log` receberá qual tipo de mensagem é controlado pelo arquivo de configuração `/etc/syslog.conf` através de *facilidades* e *níveis* (veja "Arquivo de configuração `syslog.conf`" para detalhes).

syslogd

Este daemon controla o registro de logs do sistema.

syslogd [opções]

opções, -f Especifica um arquivo de configuração alternativo ao `/etc/syslog.conf`.

-h	Permite redirecionar mensagens recebidas a outros servidores de logs especificados.
-l [computadores]	Especifica um ou mais computadores (separados por ":") que deverão ser registrados somente com o nome de máquina ao invés do FQDN (nome completo, incluindo domínio).
-m [minutos]	Intervalo em <i>minutos</i> que o syslog mostrará a mensagem --MARK--. O valor padrão padrão é 20 minutos, 0 desativa.
-n	Evita que o processo caia automaticamente em background. Necessário principalmente se o syslogd for controlado pelo init .
-p [soquete]	Especifica um soquete UNIX alternativo ao invés de usar o padrão /dev/log.
-r	Permite o recebimento de mensagens através da rede através da porta UDP 514. Esta opção é útil para criar um servidor de logs centralizado na rede. Por padrão, o servidor syslog rejeitará conexões externas.
-s [domínios]	Especifica a lista de domínios (separados por ":") que deverão ser retirados antes de enviados ao log.
-a [soquetes]	Especifica soquetes adicionais que serão monitorados. Esta opção será necessária se estiver usando um ambiente chroot . É possível usar até 19 soquetes adicionais
-d	Ativa o modo de depuração do syslog . O syslog permanecerá operando em primeiro plano e mostrará as mensagens no terminal atual.

Na distribuição **Debian**, o daemon **syslogd** é iniciado através do script /etc/init.d/syslogd.

Arquivo de configuração **syslog.conf**

O arquivo de configuração /etc/syslog.conf possui o seguinte formato:

```
facilidade.nível                                destino
```

A *facilidade* e *nível* são separadas por um "." e contém parâmetros que definem o que será registrado nos arquivos de log do sistema:

- *facilidade* - É usada para especificar que tipo de programa está enviando a mensagem. Os seguintes níveis são permitidos (em ordem alfabética):
 - *auth* - Mensagens de segurança/autorização (é recomendável usar *authpriv* ao invés deste).
 - *authpriv* - Mensagens de segurança/autorização (privativas).
 - *cron* - Daemons de agendamento (**cron** e **at**).
 - *daemon* - Outros daemons do sistema que não possuem facilidades específicas.
 - *ftp* - Daemon de ftp do sistema.
 - *kern* - Mensagens do kernel.
 - *lpr* - Subsistema de impressão.

- `local0` a `local7` - Reservados para uso local.
- `mail` - Subsistema de e-mail.
- `news` - Subsistema de notícias da USENET.
- `security` - Sinônimo para a facilidade `auth` (evite usa-la).
- `syslog` - Mensagens internas geradas pelo **syslogd**.
- `user` - Mensagens genéricas de nível do usuário.
- `uucp` - Subsistema de UUCP.
- `*` - Confere com todas as facilidades.

Mais de uma facilidade pode ser especificada na mesma linha do `syslog.conf` separando-as com ",".

- `nível` - Especifica a importância da mensagem. Os seguintes níveis são permitidos (em ordem de importância invertida; da mais para a menos importante):
 - `emerg` - O sistema está inutilizável.
 - `alert` - Uma ação deve ser tomada imediatamente para resolver o problema.
 - `crit` - Condições críticas.
 - `err` - Condições de erro.
 - `warning` - Condições de alerta.
 - `notice` - Condição normal, mas significativa.
 - `info` - Mensagens informativas.
 - `debug` - Mensagens de depuração.
 - `*` - Confere com todos os níveis.
 - `none` - Nenhuma prioridade.

Além destes níveis os seguintes sinônimos estão disponíveis:

- `error` - Sinônimo para o nível `err`.
- `panic` - Sinônimo para o nível `emerg`.
- `warn` - Sinônimo para o nível `warning`.
- `destino` - O destino das mensagens pode ser um arquivo, um pipe (se iniciado por um "|"), um computador remoto (se iniciado por uma "@"), determinados usuários do sistema (especificando os logins separados por vírgula) ou para todos os usuários logados via **wall** (usando "*").

Todas as mensagens com o nível especificado e superiores a esta especificadas no **syslog.conf** serão registradas, de acordo com as opções usadas. Conjuntos de *facilidades* e *níveis* podem ser agrupadas separando-as por ";".

OBS1: Sempre use TABS ao invés de espaços para separar os parâmetros do `syslog.conf`.

OBS2: Algumas facilidades como `security`, emitem um beep de alerta no sistema e enviam uma mensagem para o console, como forma de alerta ao administrador e usuários logados no sistema.

Existem ainda 4 caracteres que garantem funções especiais: "*", "=", "!" e "-":

- "*" - Todas as mensagens da *facilidade* especificada serão redirecionadas.
- "=" - Somente o *nível* especificado será registrado.
- "!" - Todos os *níveis* especificados e maiores NÃO serão registrados.
- "-" - Pode ser usado para desativar o sync imediato do arquivo após sua gravação.

Os caracteres especiais "=" e "!" podem ser combinados em uma mesma regra.

Exemplo: Veja abaixo um exemplo de um arquivo `/etc/syslog.conf` padrão de sistemas **Debian**

```
#
# Primeiro alguns arquivos de log padrões. Registrados por facilidade
#
auth,authpriv.*          /var/log/auth.log
*.*;auth,authpriv.none  -/var/log/syslog
cron.*                   /var/log/cron.log
daemon.*                 -/var/log/daemon.log
kern.*                   -/var/log/kern.log
lpr.*                    -/var/log/lpr.log
mail.*                   /var/log/mail.log
user.*                   -/var/log/user.log
uucp.*                   -/var/log/uucp.log

#
# Registro de logs do sistema de mensagens. Divididos para facilitar
# a criação de scripts para manipular estes arquivos.
#
mail.info                 -/var/log/mail.info
mail.warn                 -/var/log/mail.warn
mail.err                  /var/log/mail.err

# Registro para o sistema de news INN
#
news.crit                 /var/log/news/news.crit
news.err                  /var/log/news/news.err
news.notice               -/var/log/news/news.notice

#
# Alguns arquivos de registro "pega-tudo".
# São usadas ",," para especificar mais de uma prioridade (por
# exemplo, "auth,authpriv.none") e ";" para especificar mais de uma
# facilidade.nível que será gravada naquele arquivo.
# Isto permite deixar as regras consideravelmente menores e mais legíveis
#
*.=debug;\
    auth,authpriv.none;\
```

```

        news.none;mail.none      -/var/log/debug
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages

#
# Emergências são enviadas para qualquer um que estiver logado no sistema. Isto
# é feito através da especificação do "*" como destino das mensagens e são
# enviadas através do comando wall.
#
*.emerg                          *

#
# Eu gosto de ter mensagens mostradas no console, mas somente em consoles que
# não utilizo.
#
#daemon,mail.*;\
#     news.=crit;news.=err;news.=notice;\
#     *.=debug;*.=info;\
#     *.=notice;*.=warn          /dev/tty8

# O pipe /dev/xconsole é usado pelo utilitário "xconsole". Para usa-lo,
# você deve executar o "xconsole" com a opção "-file":
#
#     $ xconsole -file /dev/xconsole [...]
#
# NOTA: ajuste as regras abaixo, ou ficará maluco se tiver um um site
# muito movimentado...
#
daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
        *.=notice;*.=warn        |/dev/xconsole

# A linha baixo envia mensagens importantes para o console em que
# estamos trabalhando logados (principalmente para quem gosta de ter
# controle total sobre o que está acontecendo com seu sistema).
*.err;kern.debug;auth.notice;mail.crit /dev/console

```

klogd

Este daemon controla o registro de mensagens do kernel. Ele monitora as mensagens do kernel e as envia para o daemon de monitoramento **syslogd**, por padrão.

klogd [opções]

<i>opções</i> , -d	Ativa o modo de depuração do daemon
-f [arquivo]	Envia as mensagens do kernel para o arquivo especificado ao invés de enviar ao daemon do syslog
-i	Envia um sinal para o daemon recarregar os símbolos de módulos do kernel.
-I	Envia um sinal para o daemon recarregar os símbolos estáticos e de módulos do kernel.

-n	Evita a operação em segundo plano. Útil se iniciado pelo init
-k [arquivo]	Especifica o arquivo que contém os símbolos do kernel. Exemplos deste arquivo estão localizados em <code>/boot/System.map-xx.xx.xx</code> .
-o	Faz com que o daemon leia e registre todas as mensagens encontradas nos buffers do kernel, após isto o daemon é encerrado.
-p	Ativa o modo paranóia. Isto fará o klogd somente carregar detalhes sobre os módulos quando os caracteres <code>Oops</code> forem detectados nas mensagens do kernel. É recomendável ter sempre a última versão do klogd e evitar a utilização desta opção em ambientes críticos.
-s	Força a utilização da interface de chamadas do sistema para comunicação com o kernel.
-x	Esconde tradução EIP, assim ele não lê o arquivo <code>/boot/System.map-xx-xx-xx</code> .

A especificação de um arquivo com a opção `-k` é necessária se desejar que sejam mostradas a tabela de símbolos ao invés de endereços numéricos do kernel.

logger

Este comando permite enviar uma mensagem nos log do sistema. A mensagem é enviada aos logs via daemon **syslogd** ou via soquete do sistema, é possível especificar a prioridade, nível, um nome identificando o processo, etc. Seu uso é muito útil em shell scripts ou em outros eventos do sistema.

logger [*opções*] [*mensagem*]

Onde:

<i>mensagem</i>	Mensagem que será enviada ao daemon <i>syslog</i>
<i>opções</i> , -i	Registra o PID do processo
-s	Envia a mensagem ambos para a saída padrão (STDOUT) e syslog.
-f [arquivo]	Envia o conteúdo do arquivo especificado como <i>mensagem</i> ao syslog.
-t [nome]	Especifica o nome do processo responsável pelo log que será exibido antes do PID na mensagem do syslog.
-p [prioridade]	Especifica a prioridade da mensagem do syslog, especificada como <i>facilidade.nível</i> . Veja os tipos de prioridade/níveis em “Arquivo de configuração <i>syslog.conf</i> ”. O valor padrão <i>prioridade.nível</i> é <i>user.notice</i>
-u [soquete]	Envia a mensagem para o [soquete] especificado ao invés do syslog

Mais detalhes sobre o funcionamento sobre o daemon de log do sistema **syslogd**, pode ser encontrado em “*syslogd*”

Exemplos: `logger -i -t focalinux Teste teste teste`, `logger -i -f /tmp/mensagem -p security.emerg`

Programas úteis para monitoração e gerenciamento de arquivos de logs

logcheck

É um programa usado para enviar um e-mail periodicamente ao administrador do sistema (através do **cron** ou outro daemon com a mesma função) alertando sobre os eventos que ocorreram desde a última execução do programa. As mensagens do **logcheck** são tratadas por arquivos em `/etc/logcheck` e organizadas em categorias antes de ser enviada por e-mail, isto garante muita praticidade na interpretação dos eventos ocorridos no sistema.

As categorias são organizadas da mais importantes para a menos importante, e vão desde "Hacking em andamento" (providências devem ser tomadas imediatamente para resolver a situação) até "eventos anormais do sistema" (mensagens de inicialização, mensagens dos daemons do sistema, etc.).

O tipo de mensagem que será incluída/ignorada nos logs enviados podem ser personalizadas pelo administrador do sistema através dos arquivos/diretórios dentro de `/etc/logcheck`. Nomes de arquivos/diretórios contendo a palavra "ignore" são usados para armazenar expressões regulares que NÃO serão enviadas pelo **logcheck**. É permitido o uso de expressões regulares **perl/sed** para especificar as mensagens nos arquivos de log.

logrotate

Usado para fazer backups dos logs atuais do sistema (programado via **cron**, ou outro daemon com a mesma função) e criando novos arquivos de logs que serão usados pelo sistema. Opcionalmente os arquivos de logs antigos serão compactados para diminuir a utilização de espaço em disco ou enviados por e-mail ao administrador. A rotação dos arquivos de logs proporciona maior agilidade quando precisamos encontrar algum detalhe útil (que seria mais difícil de se achar em um arquivo de log de 10MB ou maior).

A rotação de logs é feita de acordo com o tamanho do arquivo de logs especificado, mas a opção `-f` pode ser usada para "forçar" a rotação de logs. A opção `-d` fornece mais detalhes sobre o que o **logrotate** está fazendo. Seu arquivo principal de configuração é o `/etc/logrotate.conf`. Um modelo deste tipo de arquivo é o seguinte:

```
#### Estas opções afetam globalmente o funcionamento do logrotate
# roda os arquivos de log semanalmente
weekly

# mantém as últimas 4 cópias de logs anteriores
rotate 4

# Erros de não existência dos logs são enviados para o usuário root
mail root

# Cria novos arquivos de log (vazios) após rodar os antigos
create

# Descomente isso se desejar seus arquivos de logs compactados. O parâmetro
# delaycompress é usado para que o primeiro log rodado seja mantido
# descompactado
compress
```

delaycompress

```
# Executam os scripts em prerotate e postrotate a cada vez que os logs
# forem rodados.
nosharedscripts
```

```
# Definimos um diretório que poderá conter definições individuais para
# diversos serviços no sistema, eles podem ir neste arquivo mas
# diversas configurações individuais podem deixar a interpretação
# deste arquivo confusa.
include /etc/logrotate.d
```

```
# Define opções específicas para a rotação mensal de /var/log/wtmp, o novo arquivo
# de log somente será rodados caso tenha mais de 5MB (size 5M), será criado
# com a permissão 0664 e pertencerá ao usuário root grupo utmp
# (create 0664 root utmp) e será mantida somente uma cópia do log anterior.
# (rotate 1)
/var/log/wtmp {
    monthly
    create 0664 root utmp
    size 5M
    rotate 1
}
```

```
# Define opções específicas para a rotação mensal de /var/log/btmp, se o arquivo
# não existir não será necessário gerar alertas (missingok) que serão enviados
# ao administrador. O novo arquivo criado deverá ter a permissão 0664 com o
# dono root e grupo utmp (create 0664 root utmp) e será
# mantida somente uma cópia do log anterior.
/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
```

```
# Define opções específicas para a rotação mensal de /var/log/lastlog, o novo
# arquivo será criado com a permissão 0664 com o dono root e grupo
# utmp e será mantida somente uma cópia do arquivo de log anterior
# (rotate 1).
/var/log/lastlog {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
```

```
# Define opções específicas para a rotação diária de /var/log/messages, o
# arquivo será rodado se atingir o tamanho de 1Mb, então o
# novo arquivo será criado com as mesmas permissões do arquivo anterior.
# O comando killall -1 syslogd será executado após a rotação
# para que o daemon syslogd funcione corretamente mas somente uma vez
# durante a rotação de vários arquivos de logs (sharedscripts).
```

```
# Serão mantidas as 10 últimas cópias do arquivo /var/log/messages
# compactadas (o parâmetro compress foi especificado na seção global deste
# arquivo de configuração).
/var/log/messages {
    daily
    size 1M
    sharedscripts
    postrotate
        /sbin/killall -1 syslogd
    endscript
    rotate 10
}

# Define opções específicas para a rotação mensal dos arquivos em /var/log/mirror/
# a falta desses arquivos não precisa ser notificada ao administrador (missingok),
# mesmo assim o parâmetro "nomail" evitará isto de qualquer forma. Os logs
# rodados não serão compactados (nocompress) e serão mantidas as últimas 7 cópias
# dos logs.
/var/log/mirror/* {
    montly
    nomail
    missingok
    nocompress
    rotate 7
}

# logs específicos do sistema podem ser configurados aqui. As opções padrões e
# definidas na seção global deste arquivo serão usadas para processar os
# arquivos de logs restantes.
```

Qualquer definição de parâmetro especificado no arquivo de configuração, substituirá as definições anteriores. Quando o número máximo de logs mantidos pela opção *rotate [num]* é atingida, os logs eliminados serão enviados para o usuário especificado na opção *mail [email]*. A utilização da diretiva *nomail* evita isso.

Quando for utilizar coringas para se referir a determinados arquivos dentro de um diretório, não utilize a sintaxe "log-xxx-*" porque isto forçaria a recompressão de arquivos ".gz" já feitas, gerando arquivos do tipo `.gz.gz...` e derrubando o processamento da sua máquina gerada por um loop de compactação e enchendo as entradas de diretório. Prefira usar a sintaxe `log-xxx-*.log` (ou outra, modificando a configuração do programa que gera os logs).

OBS: É importante enviar um sinal HUP ao programa que grava para aquele arquivo de log para que não ocorram problemas após a rotação, isto é feito usando o parâmetro *postrotate*.

Configurando um servidor de logs

As mensagens das máquinas de sua rede podem ser centralizadas em uma única máquina, isto facilita o gerenciamento, análise e solução de problemas que ocorrem nas máquinas da rede. Mais importante ainda é que qualquer invasão a estação de trabalho não será registrada localmente (podendo ser apagada posteriormente pelo invasor, isso é comum).

Configurando o servidor de logs Adicione a opção *-r* ao iniciar o daemon **syslogd** para aceitar logs enviados das máquinas clientes. Na distribuição **Debian** modifique o arquivo `/etc/init.d/syslogd` colocando a opção *-r* na

variável `SYSLOGD` e reinicie o serviço usando `./sysklogd restart`.

Adicionalmente poderão ser usadas as opções `-l máquina` (é um "L" minúsculo não uma letra "I") para registrar o nome FQDN da máquina e `-h` para redirecionar conexões a outros servidores de logs (veja "syslogd").

Configurando máquinas cliente

Modifique o arquivo `/etc/syslogd.conf` (veja "Arquivo de configuração `syslog.conf`" colocando o nome do computador seguido de "@" para redirecionar as mensagens dos logs:

```
auth,authpriv.*                @servlog
*.*;auth,authpriv.none        @servlog
cron.*                          @servlog
daemon.*                       @servlog
kern.*                          -/var/log/kern.log
kern.*          @servlog
lpr.*                           @servlog
mail.*                          /var/log/mail.log
user.*                          -/var/log/user.log
user.*          @servlog
uucp.*                          -/var/log/uucp.log
```

E reinicie o daemon **syslogd** da máquina cliente para re-ler o arquivo de configuração: `killall -HUP syslogd` ou `/etc/init.d/sysklogd restart`.

OBS1: Mantenha o relógio do servidor de logs sempre atualizado (use o **chrony** ou outro daemon de sincronismo NTP para automatizar esta tarefa).

OBS2: É interessante compilar um daemon **syslogd** personalizado modificando o nome e localização do arquivo `/etc/syslog.conf` para enganar possíveis invasores. Isto pode ser modificado no arquivo `syslogd.c` na linha:

```
#define _PATH_LOGCONF    "/etc/syslog.conf"
```

Use a imaginação para escolher um nome de arquivo e localização que dificulte a localização deste arquivo.

OBS3: Em uma grande rede, é recomendável configurar um computador dedicado como servidor de log (desativando qualquer outro serviço) e configurar o **iptables** para aceitar somente o tráfego indo para a porta UDP 514 (syslogd):

```
iptables -P INPUT DROP
iptables -A INPUT -p udp --dport 514 -j ACCEPT
```

Capítulo 7. A distribuição Debian GNU/Linux

Este capítulo traz algumas características sobre a distribuição **Debian GNU/Linux**, programas de configuração e particularidades. A maioria dos trechos aqui descritos, também se aplicam a distribuições baseadas na **Debian**, como o *Kurumin* e o *Ubuntu*.

Você deve estar se perguntando mas porque um capítulo falando sobre a distribuição Debian se eu uso outra?. Bem, a partir da versão *Intermediário* do *Foca Linux* existem algumas partes que são específicas de algumas distribuições Linux e que não se aplicam a outras, como a localização dos arquivos de configuração, nomes dos programas de configuração e outros detalhes específicos e esta versão é a baseada na **Debian**. Pegue na página do Foca Linux (<http://www.guiafoca.org>) uma versão Intermediário /Avançado do guia específico para sua distribuição.

Porque usar a Debian?

A **Debian** é a distribuição que mais cresce no mundo, cada versão é somente lançada após rigorosos testes de segurança e correção de falhas fazendo desta a mais segura e confiável dentre todas as outras distribuições Linux. É reconhecida como a mais segura, maior e atualizada mais frequentemente entre as outras distribuições **Linux**, além de ser a única sem fins comerciais.

É a única que adota o estilo de desenvolvimento aberto e não é mantida por uma empresa comercial (note que o endereço do WebSite da **Debian** termina com `.org`), ao invés disso é mantida por programadores, hackers e especialistas de segurança espalhados ao redor do mundo, seguindo o estilo de desenvolvimento do Linux. Possui suporte a mais de 12 arquiteturas e 15 sub-arquiteturas (entre elas, Intel x86, Alpha, VMS, Sparc, Macintosh (m68k), Power Pc, ARM, etc).

Suas atualizações são constantes e não é necessário adquirir um novo CD para fazer upgrades. Meu sistema é atualizado semanalmente e de forma segura através de 2 simples comandos. Veja ??? as instruções de como fazer isto.

Cada pacote da distribuição é mantida por uma pessoa, o que garante uma boa qualidade, implementações de novos recursos e rápida correção de falhas. Qualquer pessoa com bons conhecimentos no sistema e inglês pode se tornar um *Debian Developer*, para detalhes consulte a lista de discussão *debian-user-portuguese* (veja “Listas de discussão”) ou veja a página oficial da **Debian**: <http://www.debian.org/>.

A distribuição apresenta compatibilidade com outros sistemas a partir da instalação até a seleção de programas e execução do sistema, sua instalação está até mesmo disponível desde computadores 386 que utilizam unidades de disquetes de 5 1/4 polegadas até para computadores UDMA66, instalando através de DVD e pen drives. Com a Debian é possível iniciar a instalação usando um pen drive e continuar usando a internet.

É a distribuição mais indicada para uso em servidores devido ao seu desempenho, segurança e programas úteis de gerenciamento e monitoração da rede, recomendados por especialistas que participam de seu desenvolvimento.

Não existem versões separadas da Debian para servidores, uso pessoal, etc, ao invés disso a distribuição usa perfis de usuário (dependendo da função do usuário) e perfis de computador (dependendo do que deseja fazer), podendo ser selecionado mais de um perfil de usuário/computador.

Os perfis selecionam automaticamente os pacotes mais úteis para a instalação. Os pacotes existentes em cada perfil foram escolhidos através de debates entre usuários que trabalham ativamente naquela área, resultando em uma seleção de pacotes de alta produtividade.

Para os usuários avançados e exigentes, também é possível selecionar os pacotes individualmente via **dselect**, o que resultará em uma instalação somente com pacotes úteis e melhor configurada.

Pacotes existentes na Debian

O número de pacotes existentes na distribuição atual da **Debian** (*Buster* - 10) é de 59000.

A **Debian** (como a **Red Hat**) usa um formato próprio para armazenar os programas: o formato `.deb`. Este formato permite a declaração, resolução e checagem automática de dependências, pacotes sugeridos, opcionais e outras características que o torna atraente para o desenvolvimento, gerenciamento e manutenção do sistema.

Estes pacotes são gerenciados através do programa **dpkg** (*Debian Package*) ou através de front-ends como o **dselect** ou **apt** (para detalhes veja ???).

O que é sid/testing/frozen/stable?

Para o lançamento de uma nova distribuição **Debian**, o seguinte processo ocorre: `sid => testing => stable` (sendo a *stable* sempre o lançamento oficial e sem bugs da distribuição).

sid Durante o desenvolvimento de uma nova distribuição **Debian**, ela é chamada de *sid*. A *sid* é a versão *Unstable*, isto não significa instabilidade, mas sim que a distribuição esta sofrendo modificações para se tornar uma versão estável, recebendo novos pacotes, etc.

Quando os pacotes não são modificados após um determinado período, os scripts da Debian copiam estes pacotes (novos ou atualizados) para a *testing*.

Não use a distribuição *sid* (*unstable*) ao menos que tenha experiência no **Linux** para corrigir problemas, que certamente aparecerão.

testing A *testing* recebe os pacotes que não são modificados durante algum tempo da *unstable*, isto significa que eles possuem alguma estabilidade.

A *testing* é uma espécie de congelamento permanente (freeze) durante o desenvolvimento da *Unstable*.

Os novos pacotes que entram na *unstable* também caem na *testing* após certo tempo.

Mesmo assim, podem existir falhas graves na *testing*, se você precisa de um ambiente realmente livre de falhas, use a *stable*.

frozen (congelada) Na data programada pela equipe de lançamento da **Debian**, a distribuição *testing* é congelada: nenhum pacote novo da *unstable* cai na *testing* e começa a procura de falhas na distribuição *testing*. Nenhuma nova característica é implementada nos pacotes (a não ser que seja extremamente necessário) e os developers se dedicam a correção de erros nos pacotes.

A distribuição *testing* congelada se tornará a futura *stable* após todas as falhas serem corrigidas. É considerado seguro usar a *frozen* após 1 mês de "congelamento".

Quando a *testing* é congelada, o ciclo de desenvolvimento da *unstable* continua para que a próxima distribuição da **Debian** seja lançada.

stable

Quando todos os bugs da *testing* congelada são eliminados, ela é lançada como *stable*, a nova *versão Oficial* da **Debian**.

A *stable* é o resultado final do desenvolvimento, das correções de falhas/segurança e que passou por todos os ciclos de testes para ser lançada. Resumindo é a distribuição pronta para ser usada com toda a segurança.

Como obter a Debian

A instalação da distribuição pode ser obtida através de Download de <ftp://ftp.debian.org/debian/dists/stable/main/disks-i386> (para Intel x86), seus programas diversos estão disponíveis em <ftp://ftp.debian.org/debian/dists/stable/main/binary-i386>.

Programas de configuração

- **aptitude** - Seleciona pacote para instalação/desinstalação
- **pppconfig** - Configura o computador para se conectar a Internet usando conexão discada. Após isto, use `pon` para se conectar a Internet, `poff` para se desconectar e `plog` para monitorar a conexão.
- **pppoeconf** - Configura o computador para conectar a internet usando ADSL
- **modconf** - Permite selecionar os módulos que serão automaticamente carregados na inicialização do sistema. Se requerido pelos módulos os parâmetros I/O, IRQ e DMA também podem ser especificados.
- **shadowconfig** - Permite ativar ou desativar o suporte a senhas ocultas (shadow password). Com as senhas ocultas ativadas, as senhas criptografadas dos usuários e grupos são armazenadas nos arquivos `shadow` e `gshadow` respectivamente, que somente podem ser acessadas pelo usuário root.

Isto aumenta consideravelmente a segurança do sistema pois os arquivos `passwd` e `group` contém dados de usuários que devem ter permissão de leitura de todos os usuários do sistema.

- **tasksel** - Permite selecionar/modificar de forma fácil a instalação de pacotes em seu sistema através da função que sua máquina terá ou do seu perfil de usuário.
- **tzconfig** - Permite modificar/selecionar o fuso-horário usado na distribuição.

Além destes, a Debian conta com o sistema de configuração baseado no `dpkg-reconfigure` que permite configurar de forma fácil e rápida aspecto de pacotes: `dpkg-reconfigure xserver-xorg`.

Arquivos de inicialização

Os arquivos de inicialização da distribuição **Debian** (e baseadas nela) estão localizados no diretório `/etc/init.d`. Cada daemon (programa residente na memória) ou configuração específica possui um arquivo de onde pode ser ativado/desativado. Os sistemas residentes neste diretório não são ativados diretamente, mas sim através de links existentes nos diretórios `/etc/rc?.d` onde cada diretório consiste em um nível de execução do sistema (veja também a “Níveis de Execução”).

Por padrão, você pode usar as seguintes palavras chaves com os arquivos de configuração:

- `start` - Inicia o daemon ou executa a configuração
- `stop` - Interrompe a execução de um daemon ou desfaz a configuração feita anteriormente (se possível).

- `restart` - Reinicia a execução de um daemon. É equivalente ao uso de `stop` e `start` mas se aplicam somente a alguns daemons e configurações, que permitem a interrupção de execução e reinício.

Por exemplo, para reconfigurar as interfaces de rede do computador, podemos utilizar os seguintes comandos:

```
cd /etc/init.d
./networking restart
```

Níveis de Execução

Os *Níveis de execução* (run levels) são diferentes modos de funcionamento do **GNU/Linux** com programas, daemons e recursos específicos. Em geral, os sistemas **GNU/Linux** possuem sete níveis de execução numerados de 0 a 6. O daemon **init** é o primeiro programa executado no **GNU/Linux** (veja através do `ps ax|grep init`) e responsável pela carga de todos daemons de inicialização e configuração do sistema.

O nível de execução padrão em uma distribuição **GNU/Linux** é definido através do arquivo de configuração do `/etc/inittab` (???) através da linha

```
id:2:initdefault:
```

Entendendo o funcionamento dos níveis de execução do sistema (runlevels)

Os nível de execução atual do sistema pode ser visualizado através do comando **runlevel** e modificado através dos programas **init** ou **telinit**. Quando é executado, o **runlevel** lê o arquivo `/var/run/utmp` e adicionalmente lista o nível de execução anterior ou a letra N em seu lugar (caso ainda não tenha ocorrido a mudança do nível de execução do sistema).

Na **Debian**, os diretórios `/etc/rc0.d` a `/etc/rc6.d` contém os links simbólicos para arquivos em `/etc/init.d` que são acionados pelo nível de execução correspondente.

Por exemplo, o arquivo `S10sysklogd` em `/etc/rc2.d`, é um link simbólico para `/etc/init.d/sysklogd`.

O que aconteceria se você removesse o arquivo `/etc/rc2.d/S10sysklogd`? Simplesmente o daemon **sysklogd** deixaria de ser executado no nível de execução 2 do sistema (que é o padrão da **Debian**).

A **Debian** segue o seguinte padrão para definir se um link simbólico em `/etc/rc[0-6].d` iniciará ou interromperá a execução de um serviço em `/etc/init.d`, que é o seguinte:

- Se um link é iniciado com a letra K (kill), quer dizer que o serviço será interrompido naquele nível de execução. O que ele faz é executar o daemon em `/etc/init.d` seguido de `stop`.
- Se um link é iniciado com a letra S (start), quer dizer que o serviço será iniciado naquele nível de execução (é equivalente a executar o daemon seguido de `start`).

Primeiro os links com a letra K são executado e depois os S. A ordem que os links são executados dependem do valor numérico que acompanha o link, por exemplo, os seguintes arquivos são executados em seqüência:

```
S10sysklogd
S12kerneld
S20inetd
S20linuxlogo
S20logoutd
S20lprng
S89cron
S99xdm
```

Note que os arquivos que iniciam com o mesmo número (S20*) são executados alfabeticamente. O nível de execução do sistema pode ser modificado usando-se o comando **init** ou **telinit**. Os seguinte níveis de execução estão disponíveis na **Debian**:

- 0 - Interrompe a execução do sistema. todos os programas e daemons finalizados. É acionado pelo comando `shutdown -h`
- 1 - Modo monousuário, útil para manutenção dos sistema.
- 2 - Modo multiusuário (padrão da Debian)
- 3 - Modo multiusuário
- 4 - Modo multiusuário
- 5 - Modo multiusuário com login gráfico
- 6 - Reinicialização do sistema. Todos os programas e daemons são encerrados e o sistema é reiniciado. É acionado pelo comando `shutdown -r` e o pressionamento de CTRL+ALT+DEL.

Por exemplo, para listar o nível de execução atual do sistema digite: `runlevel`. O **runlevel** deverá listar algo como:

```
N 2
```

Agora para mudar para o nível de execução 1, digite: `init 3`. Agora confira a mudança digitando: `runlevel`. Você deverá ver este resultado:

```
2 3
```

Isto quer dizer que o nível de execução anterior era o 2 e o atual é o 3.

Rede no sistema Debian

O local que contém as configurações de rede em um sistema **Debian** é o `/etc/network/interfaces`. O formato deste arquivo é descrito em ???.

Bug tracking system

É o sistema para relatar bugs e enviar sugestões sobre a distribuição. Para relatar um bug primeiro você deve saber inglês (é a língua universal entendida pelos desenvolvedores) e verificar se o bug já foi relatado. O Debian *Bug tracking system* pode ser acessado pelo endereço: <http://bugs.debian.org/>.

Para relatar uma falha/sugestão, envie um e-mail para: <submit@bugs.debian.org>, com o assunto referente a falha/sugestão que deseja fazer e no corpo da mensagem:

Package: pacote
Severity: normal/grave/wishlist
Version: versão do pacote

E o relato do problema

O bug será encaminhado diretamente ao mantenedor do pacote que verificará o problema relatado. Os campos `Package` e `Severity` são obrigatórios para definir o nome do pacote (para endereçar o bug para a pessoa correta) e versão do pacote (esta falha pode ter sido relatada e corrigida em uma nova versão).

Onde encontrar a Debian para Download?

No endereço <ftp://ftp.debian.org/>. Outros endereços podem ser obtidos na página oficial da **Debian** (<http://www.debian.org/>) clicando no link `Download e mirrors`.

A distribuição Etch (4.0) completa, com 18830 pacotes ocupa em torno de 10 GB. Você também pode optar por fazer a instalação dos pacotes opcionais via Internet através do método `apt`. Para detalhes veja o guia do `dselect` ou envie uma mensagem para a lista de discussão <debian-user-portuguese@lists.debian.org> (veja “Listas de discussão” para detalhes).

Lista de pacotes para uma instalação rápida e manual

Esta seção contém uma lista de pacotes necessários que atendem a maioria dos usuários normais da **Debian** em um **sistema padrão** sem desperdício de espaço e sabendo exatamente o que está instalando.

Estou assumindo que você concluiu a instalação da **Debian 10.0** (Buster) mas preferiu pular o passo de seleção de pacotes do `dselect` e fazer uma instalação manual.

A lista de pacotes está dividida por categorias e você precisa ter o programa `apt` configurado corretamente para que os comandos funcionem (veja ??? para detalhes).

Se pretende usar a lista de pacotes para fazer a instalação da **Debian** em muitos computadores, você tem duas opções:

1. Copiar o conteúdo das seções que seguem e fazer um script de instalação personalizado para automatizar a instalação de pacotes da **Debian** em outras máquinas
2. Após a instalação dos pacotes no computador, utilize o comando `dpkg --get-selections >Lista-Pacotes.txt` para gerar o arquivo `Lista-Pacotes.txt` contendo a lista de pacotes instalados.

Então no computador que pretende fazer a instalação de pacotes, use o comando `dpkg --set-selections <Lista-Pacotes.txt` e então digitar `apt-get -f install` ou escolher a opção `Install` no `dselect`.

Para mais detalhes veja ??? e a ???. É importante usar o comando `apt-get clean` após a instalação de pacotes para remover os pacotes baixados pelo `apt` de `/var/cache/apt/archives` (exceto na instalação de pacotes através do disco rígido local).

Pacotes Básicos (Altamente Recomendado)

```
apt-get install cpio info libident libncurses4 man-db manpages whois vim
hdparm mc postfix linuxlogo less kbd mutt bzip2
cron gpm
```

Compilação do Kernel e programas em linguagem C

```
apt-get install perl, gcc libc6-dev bin86 make
```

Se pretender utilizar o para compilar o kernel mais facilmente, então você precisará dos seguintes pacotes:

```
apt-get install kernel-package dpkg-dev
```

Veja ??? para entender como compilar seu próprio kernel.

X11 (básico)

```
apt-get install xbase-clients xserver-xorg xfonts-75dpi xfonts-base
xserver-common xterm xfstt xdm
```

Caso suas fontes sejam mostradas em tamanho exagerado, remova o pacotes ou ajuste a seção Files do arquivo `/etc/X11/xorg.conf` apropriadamente.

Window Managers para o X

```
apt-get install wmaker wmakerconf wmaker-data wmapload
eterm enlightenment enlightenment-theme-bluesteel asclock
afterstep
```

OBS: Existem também gerenciadores de seção como o **gnome**, **kde**, ocupam bastante espaço em disco

Impressão (texto e gráfico com sistema de spool)

```
apt-get install lprng magicfilter gs gsfonds
```

Som (mixer, mp3, Midi, wav, CD-Player)

```
xmms playmidi cam aumix alsa-base alsa-oss alsamixer xgcd sox
```

Programas de Internet (clientes)

```
apt-get install xchat gaim firefox fetchmail procmail mime-support
```

Acessórios

```
apt-get install gimp gimp-nonfree gnotepad openoffice freefont
```

Rede

```
apt-get install finger, talk, talkd, telnet
```

Capítulo 8. Personalização do Sistema

Este capítulo ensina como personalizar algumas características de seu sistema **GNU/Linux**.

Variáveis de Ambientes

É um método simples e prático que permite a especificação de opções de configuração de programas sem precisar mexer com arquivos no disco ou opções. Algumas variáveis do **GNU/Linux** afetam o comportamento de todo o Sistema Operacional, como o idioma utilizado e o path (veja ???) . Variáveis de ambientes são nomes que contém algum valor e tem a forma `Nome=Valor`. As variáveis de ambiente são individuais para cada usuário do sistema ou consoles virtuais e permanecem residentes na memória RAM até que o usuário saia do sistema (logo-off) ou até que o sistema seja desligado.

As variáveis de ambiente são visualizadas/criadas através do comando `set` ou `echo $NOME` (apenas visualiza) e exportadas para o sistemas com o comando `export NOME=VALOR`.

Nos sistemas **Debian**, o local usado para especificar variáveis de ambiente é o `/etc/environment` (veja “Arquivo `/etc/environment`”). Todas as variáveis especificadas neste arquivos serão inicializadas e automaticamente exportadas na inicialização do sistema.

Exemplo: Para criar uma variável chamada `TESTE` que contenha o valor `123456` digite: `export TESTE=123456`. Agora para ver o resultado digite: `echo $TESTE` ou `set | grep TESTE`. Note que o `$` que antecede o nome `TESTE` serve para identificar que se trata de uma variável e não de um arquivo comum.

Modificando o Idioma usado em seu sistema

O idioma usado em seu sistema pode ser modificado facilmente através das variáveis de ambiente. Atualmente a maioria dos programas estão sendo *localizados*. A localização é um recurso que especifica arquivos que contém as mensagens do programas em outros idiomas. Você pode usar o comando **locale** para listar as variáveis de localização do sistema e seus respectivos valores. As principais variáveis usadas para determinar qual idioma os programas *localizados* utilizarão são:

- `LANG` - Especifica o idioma_`PAIS` local. Podem ser especificados mais de um idioma na mesma variável separando-os com `:`, desta forma caso o primeiro não esteja disponível para o programa o segundo será verificado e assim por diante. A língua Inglesa é identificada pelo código `C` e usada como padrão caso nenhum locale seja especificado.

Por exemplo: `export LANG=pt_BR`, `export LANG=pt_BR:pt_PT:C`

- `LC_MESSAGES` - Especifica o idioma que serão mostradas as mensagens dos programas. Seu formato é o mesmo de `LANG`.
- `LC_ALL` - Configura todas as variáveis de localização de uma só vez. Seu formato é o mesmo de `LANG`.

As mensagens de localização estão localizadas em arquivos individuais de cada programa em `/usr/share/locale/[Idioma]/LC_MESSAGES` . Elas são geradas através de arquivos `potfiles` (arquivos com a extensão `.po` ou `.pot` e são gerados catálogos de mensagens `.mo`. As variáveis de ambiente podem ser especificadas no arquivo `/etc/environment` desta forma as variáveis serão carregadas toda a vez que seu sistema for iniciado. Você também pode especificar as variáveis de localização em seu arquivos de inicialização `.bash_profile`, `.bashrc` ou `.profile` assim toda a vez que entrar no sistema, as variáveis de localização personalizadas serão carregadas.

Siga as instruções a seguir de acordo com a versão de sua distribuição **Debian**:

Debian 4.0 Acrescente a linha `pt_BR ISO-8859-1` no arquivo `/etc/locale.gen`, rode o utilitário **locale-gen** para gerar os locales. Agora acrescente as variáveis de localização no arquivo `/etc/locale.def` seguindo a forma:

```
export LANG=pt_BR
export LC_ALL=pt_BR
export LC_MESSAGES=pt_BR
```

Note que o arquivo `/etc/environment` também pode ser usado para tal tarefa, mas o `locales.def` foi criado especialmente para lidar com variáveis de localização no **Debian 4.0**.

Para as mensagens e programas do X-Window usarem em seu idioma local, é preciso colocar as variáveis no arquivo `~/.xserverrc` do diretório home de cada usuário e dar a permissão de execução neste arquivo (`chmod 755 .xserverrc`). Lembre-se de incluir o caminho completo do arquivo executável do seu gerenciador de janelas na última linha deste arquivo (sem o `&` no final), caso contrário o Xserver será finalizado logo após ler este arquivo.

Abaixo exemplos de localização com as explicações:

- `export LANG=pt_BR` - Usa o idioma `pt_BR` como língua padrão do sistema. Caso o idioma Português do Brasil não esteja disponível, `C` é usado (Inglês).
- `export LANG=C` - Usa o idioma Inglês como padrão (é a mesma coisa de não especificar `LANG`, pois o idioma Inglês é usado como padrão).
- `export LANG=pt_BR:pt_PT:es_ES:C` - Usa o idioma Português do Brasil como padrão, caso não esteja disponível usa o Português de Portugal, se não estiver disponível usa o Espanhol e por fim o Inglês.
- `LANG=es_ES ls --help` - Executa apenas o comando `ls --help` usando o idioma `es_ES` (sem alterar o locale do sistema).

É recomendável usar a variável `LC_ALL` para especificar o idioma, desta forma todas as outras variáveis (`LANG`, `MESSAGES`, `LC_MONETARY`, `LC_NUMERIC`, `LC_COLLATE`, `LC_CTYPE` e `LC_TIME`) serão configuradas automaticamente.

alias

Permite criar um apelido a um comando ou programa. Por exemplo, se você gosta de digitar (como eu) o comando `ls --color=auto` para ver uma listagem longa e colorida, você pode usar o comando **alias** para facilitar as coisas digitando: `alias ls='ls --color=auto'` (não se esqueça da meia aspa 'para identificar o comando'). Agora quando você digitar `ls`, a listagem será mostrada com cores.

Se você digitar `ls -la`, a opção `-la` será adicionada no final da linha de comando do alias: `ls --color=auto -la`, e a listagem também será mostrada em cores.

Se quiser utilizar isto toda vez que entrar no sistema, veja “Arquivo `.bash_profile`” e “Arquivo `.bashrc`”.

Arquivo `/etc/profile`

Este arquivo contém comandos que são executados para *todos* os usuários do sistema no momento do login. Somente o usuário `root` pode ter permissão para modificar este arquivo.

Este arquivo é lido antes do arquivo de configuração pessoal de cada usuário (`.profile`(root) e `.bash_profile`).

Quando é carregado através de um shell que requer login (nome e senha), o **bash** procura estes arquivos em seqüência e executa os comandos contidos, caso existam:

1. `/etc/profile`
2. `~/.bash_profile`
3. `~/.bash_login`
4. `~/.profile`

Ele *interrompe* a pesquisa assim que localiza o primeiro arquivo no diretório do usuário (usando a seqüência acima). Por exemplo, se você tem o arquivo `~/.bash_login` e `~/.bash_profile` em seu diretório de usuário, ele processará o `/etc/profile` e após isto o `~/.bash_profile`, mas nunca processará o `~/.bash_login` (a menos que o `~/.bash_profile` seja apagado ou renomeado).

Caso o **bash** seja carregado através de um shell que não requer login (um terminal no X, por exemplo), o seguinte arquivo é executado: `~/.bashrc`.

Observação: Nos sistemas Debian, o `profile` do usuário `root` está configurado no arquivo `/root/.profile`. A razão disto é porque se o **bash** for carregado através do comando **sh**, ele fará a inicialização clássica deste shell lendo primeiro o arquivo `/etc/profile` e após o `~/.profile` e ignorando o `.bash_profile` e `.bashrc` que são arquivos de configuração usados somente pelo **Bash**. Exemplo, inserindo a linha `mesg y` no arquivo `/etc/profile` permite que todos os usuários do sistema recebam pedidos de `talk` de outros usuários. Caso um usuário não quiser receber pedidos de **talk**, basta somente adicionar a linha `mesg n` no arquivo pessoal `.bash_profile`.

Arquivo `.bash_profile`

Este arquivo reside no diretório pessoal de cada usuário. É executado por shells que usam autenticação (nome e senha). `.bash_profile` contém comandos que são executados para o usuário no momento do login no sistema após o `/etc/profile`. Note que este é um arquivo oculto pois tem um "." no início do nome.

Por exemplo colocando a linha: `alias ls='ls --colors=auto'` no `.bash_profile`, cria um apelido para o comando `ls --colors=auto` usando `ls`, assim toda vez que você digitar `ls` será mostrada a listagem colorida.

Arquivo `.bashrc`

Possui as mesmas características do `.bash_profile` mas é executado por shells que não requerem autenticação (como uma seção de terminal no X).

Os comandos deste arquivo são executados no momento que o usuário inicia um shell com as características acima. Note que este é um arquivo oculto pois tem um "." no início do nome.

Arquivo `.hushlogin`

Deve ser colocado no diretório pessoal do usuário. Este arquivo faz o **bash** pular as mensagens do `/etc/motd`, número de e-mails, etc. Exibindo imediatamente o aviso de comando após a digitação da senha.

Arquivo `/etc/environment`

Armazena as variáveis de ambiente que são exportadas para todo o sistema. Uma variável de ambiente controla o comportamento de um programa, registram detalhes úteis durante a sessão do usuário no sistema, especificam o idioma das mensagens do sistema, etc.

Exemplo do conteúdo de um arquivo `/etc/environment`:

```
LANG=pt_BR
LC_ALL=pt_BR
LC_MESSAGES=pt_BR
```

Diretório `/etc/skel`

Este diretório contém os modelos de arquivos `.bash_profile` e `.bashrc` que serão copiados para o diretório pessoal dos usuários no momento que for criada uma conta no sistema. Desta forma você não precisará configurar estes arquivos separadamente para cada usuário.

Capítulo 9. Impressão

Este capítulo descreve como imprimir em seu sistema **GNU/Linux** e as formas de impressão via spool, rede, gráfica, etc.

Antes de seguir os passos descritos neste capítulo, tenha certeza que seu kernel foi compilado com o suporte a impressora USB e/ou paralela ativado, caso contrário até mesmo a impressão direta para a porta de impressora falhará. Para detalhes veja ???.

Portas de impressora

Uma porta de impressora é o local do sistema usado para se comunicar com a impressora. Em sistemas **GNU/Linux**, a porta de impressora paralela é identificada como `lp0`, `lp1`, `lp2` no diretório `/dev`, caso a impressora seja USB, o dispositivo será o mesmo, mas estará disponível no diretório `/dev/usb`. Os dispositivos `lp0`, `lp1` e `lp2` correspondem respectivamente a LPT1, LPT2 e LPT3 no **DOS** e **Windows**. Recomendo que o suporte a porta paralela esteja compilado como módulo no kernel.

Imprimindo diretamente para a porta de impressora

Isto é feito direcionando a saída ou o texto com `>` diretamente para a porta de impressora no diretório `/dev`.

Supondo que você quer imprimir o texto contido do arquivo `trabalho.txt` e a porta de impressora em seu sistema é `/dev/usb/lp0`, você pode usar os seguintes comandos:

- `cat trabalho.txt >/dev/usb/lp0` - Direciona a saída do comando **cat** para a impressora USB conectada em `lp0`.
- `cat <trabalho.txt >/dev/usb/lp0`. Faz a mesma coisa que o acima.
- `cat -n trabalho.txt >/dev/usb/lp0` - Numera as linhas durante a impressão.
- `head -n 30 trabalho.txt >/dev/usb/lp0` - Imprime as 30 linhas iniciais do arquivo.
- `cat trabalho.txt | tee /dev/usb/lp0` - Mostra o conteúdo do **cat** na tela e envia também para a impressora USB.

Os métodos acima servem somente para imprimir em modo texto (letras, números e caracteres semi-gráficos).

OBS: Note que a impressora somente imprimirá diretamente a partir da porta, caso ela seja uma impressora com firmware interna (impressora inteligente). Algumas impressoras mais recentes (principalmente os modelos mais baratos) somente imprimem caso estejam configuradas com o respectivo driver (Win Printers ou impressoras via software), e nunca aceitarão o comando diretamente para a porta de impressão. Para *Win Printers*, a melhor alternativa de configuração de funcionamento será através do CUPS (Common Unix Print System).

Imprimindo via spool

A impressão via spool (fila de impressão) tem por objetivo liberar logo o programa do serviço que está fazendo a impressão deixando um outro programa específico tomar conta.

Este programa é chamado de *daemon de impressão*, normalmente é o **lpr** ou o **lprng** (recomendado) em sistemas **GNU/Linux**.

Logo após receber o arquivo que será impresso, o programa de spool gera um arquivo temporário (normalmente localizado em `/var/spool/lpd`) que será colocado em fila para a impressão (um trabalho será impresso após o outro, em seqüência). O arquivo temporário gerado pelo programa de spool é apagado logo após concluir a impressão.

Antes de se imprimir qualquer coisa usando os daemons de impressão, é preciso configurar os parâmetros de sua impressora no arquivo `/etc/printcap`. Um arquivo `/etc/printcap` para uma impressora local padrão se parece com o seguinte:

```
lp|Impressora compatível com Linux
:lp=/dev/lp0
:sd=/var/spool/lpd/lp
:af=/var/log/lp-acct
:lf=/var/log/lp-errs
:pl#66
:pw#80
:pc#150
:mx#0
:sh
```

É possível também compartilhar a impressora para a impressão em sistemas remotos, isto será visto em uma seção separada neste guia.

Usando os exemplos anteriores da seção Imprimindo diretamente para uma porta de impressora, vamos acelerar as coisas:

- `cat trabalho.txt |lpr` - Direciona a saída do comando **cat** para o programa de spool **lpr**.
- `cat <trabalho.txt |lpr`. Faz a mesma coisa que o acima.
- `cat -n trabalho.txt |lpr` - Numera as linhas durante a impressão.
- `head -n 30 trabalho.txt |lpr` - Imprime as 30 linhas iniciais do arquivo.

A fila de impressão pode ser controlada com os comandos:

- `lpq` - Mostra os trabalhos de impressão atuais
- `lprm` - Remove um trabalho de impressão

Ou usado o programa de administração **lpc** para gerenciar a fila de impressão (veja a página de manual do **lpc** ou digite `?` ao iniciar o programa para detalhes).

OBS1: Se a impressora não imprimir ou não for possível compartilhar a porta de impressora paralela com outros dispositivos (tal como o *plip*), verifique se o módulo *parport_pc* foi carregado e com os valores de `irq` e `I/O` corretos (por exemplo, `modprobe parport_pc io=0x378 irq=7`). Muitas vezes sua porta paralela pode funcionar sem problemas durante a impressão, mas se ao utilizar *plip* ocorrerem erros, a causa pode ser essa. Na distribuição **Debian**, use o programa **modconf** para configurar os valores permanentemente para o módulo *parport_pc*.

OBS2: Se tiver mais de uma impressora instalada na máquina, será necessário especificar a opção `"-P impressora"` para especificar qual impressora deseja imprimir/controlar.

Impressão em modo gráfico

A impressão em modo gráfico requer que conheça a marca e modelo de sua impressora e os métodos usados para imprimir seus documentos. Este guia abordará somente a segunda recomendação :-)

Ghost Script

O método mais usados pelos aplicativos do **GNU/Linux** para a impressão de gráficos do *Ghost Script*. O Ghost Script (chamado de *gs*) é um interpretador do formato *Pos Script* (arquivos *.ps*) e pode enviar o resultado de processamento tanto para a tela como impressora. Ele está disponível para diversas plataformas e sistemas operacionais além do **GNU/Linux**, inclusive o **DOS**, **Windows**, **OS/2**, etc.

O formato *.ps* esta se tornando uma padronização para a impressão de gráficos em **GNU/Linux** devido a boa qualidade da impressão, liberdade de configuração, gerenciamento de impressão feito pelo *gs* e por ser um formato universal, compatíveis com outros sistemas operacionais.

Para imprimir um documento via Ghost Script, você precisará do pacote `gs`, (para a distribuição **Debian** e distribuições baseadas, ou outros de acordo com sua distribuição Linux) e suas dependências. A distribuição **Debian** vem com vários exemplos Pos Script no diretório `/usr/share/doc/gs/example` que são úteis para o aprendizado e testes com o Ghost Script.

Hora da diversão:

- Copie os arquivos `tiger.ps.gz` e `alphabet.ps.gz` do diretório `/usr/share/doc/gs/examples` (sistemas **Debian**) para `/tmp` e descompacte-os com o comando **`gzip -d tiger.ps.gz e gzip -d alphabet.ps.gz`**. Se a sua distribuição não possui arquivos de exemplo ou você não encontra nenhuma referência de onde se localizam, mande um e-mail que os envio os 2 arquivos acima (são 32Kb).
- O Ghost Script requer um monitor EGA, VGA ou superior para a visualização dos seus arquivos (não tenho certeza se ele funciona com monitores CGA ou Hércules Monocromático).

Para visualizar os arquivos na tela digite:

```
gs tiger.ps
gs alphabet.ps
```

Para sair do **Ghost Script** pressione CTRL+C. Neste ponto você deve ter visto um desenho de um tigre e (talvez) letras do alfabeto.

Se o comando `gs alphabet.ps` mostrou somente uma tela em branco, você se esqueceu de instalar as fontes do Ghost Script (estão localizadas no pacote na distribuição Debian).

- Para imprimir o arquivo `alphabet.ps` use o comando:

```
gs -q -dSAFER -dNOPAUSE -sDEVICE=epson -r240x72 -sPAPERSIZE=legal -sOutputFile=/
alphabet.ps
```

O arquivo `alphabet.ps` deve ser impresso. Caso aparecerem mensagens como `Error: /invalidfont in findfont` no lugar das letras, você se esqueceu de instalar ou configurar as fontes do Ghost Script. Instale o pacote de fontes (na **Debian**) ou verifique a documentação sobre como configurar as fontes.

Cada uma das opções acima descrevem o seguinte:

- `-q, -dQUIET` - Não mostra mensagens de inicialização do Ghost Script.
- `-dSAFER` - É uma opção para ambientes seguros, pois desativa a operação de mudança de nome e deleção de arquivo e permite somente a abertura dos arquivos no modo somente leitura.
- `-dNOPAUSE` - Desativa a pausa no final de cada página processada.
- `-sDEVICE=dispositivo` - Dispositivo que receberá a saída do Ghost Script. Neste local pode ser especificada a marca o modelo de sua impressora ou um formato de arquivo diferente (como `pcxmono`, `bmp256`) para que o arquivo `.ps` seja convertido para o formato designado.

Para detalhes sobre os dispositivos disponíveis em seu Ghost Script, digite `gs --help|less` ou veja a página de manual. Normalmente os nomes de impressoras e modelos são concatenados, por exemplo, `bjc600` para a impressora *Canon BJC 600*, `epson` para impressoras padrão `epson`, `stcolor` para *Epson Stylus color*, etc.

O Hardware-HOWTO contém referências sobre hardware suportados pelo **GNU/Linux**, tal como impressoras e sua leitura pode ser útil.

- `-r<ResH>x<ResV>` - Define a resolução de impressão (em dpi) Horizontal e Vertical. Os valores dependem de sua impressora.
- `-sPAPERSIZE=tamanho` - Tamanho do papel. Podem ser usados `a4`, `legal`, `letter`, etc. Veja a página de manual do `gs` para ver os outros tipos suportados e suas medidas.
- `-sOutputFile=dispositivo` - Dispositivo que receberá a saída de processamento do `gs`. Você pode especificar
 - `arquivo.epson` - Nome do arquivo que receberá todo o resultado do processamento. O `arquivo.epson` terá toda a impressão codificada no formato entendido por impressoras `epson` e poderá ser impresso com o comando `cat arquivo.epson >/dev/lp0`.

Uma curiosidade útil: É possível imprimir este arquivo em outros sistemas operacionais, tal como o **DOS** digitando: `copy /b arquivo.eps prn` (lembre-se que o **DOS** tem um limite de 8 letras no nome do arquivo e 3 na extensão. Você deve estar compreendendo a flexibilidade que o **GNU/Linux** e suas ferramentas permitem, isso é só o começo.

- `impressao%d.epson` - Nome do arquivo que receberá o resultado do processamento. Cada página será gravada em arquivos separados como `impressao1.epson`, `impressao2.epson`.

Os arquivos podem ser impressos usando os mesmos métodos acima.

- `/dev/lp0` para uma impressora em `/dev/lp0`
- `-` para redirecionar a saída de processamento do `gs` para a saída padrão. É útil para usar o `gs` com pipes `|`.
- `\|lpr` - Envia a saída do Ghost Script para o daemon de impressão. O objetivo é deixar a impressão mais rápida.

Se você é curioso ou não está satisfeito com as opções mostradas acima, veja a página de manual do `gs`.

Magic Filter

O *Magic Filter* é um filtro de impressão inteligente. Ele funciona acionado pelo spool de impressão (mais especificamente o arquivo `/etc/printcap`) e permite identificar e imprimir arquivos de diversos tipos diretamente através do comando `lpr arquivo`.

É um ótimo programa e **ALTAMENTE RECOMENDADO** se você deseja apenas clicar no botão imprimir e deixar os programas fazerem o resto :-). A intenção do programa é justamente automatizar os trabalhos de impressão e spool.

A maioria dos programas para ambiente gráfico X11, incluindo o Netscape, Word Perfect, Gimp e Star Office trabalham nativamente com o **magicfilter**.

Instalação e configuração do Magic Filter

O Magic Filter é encontrado no pacote da distribuição **Debian** e baseadas.

Sua configuração pode ser feita com o programa **magicfilterconfig** que torna o processo de configuração rápido e fácil para quem não conhece a sintaxe do arquivo `/etc/printcap` ou não tem muitas exigências sobre a configuração detalhada da impressora.

Após instalar o **magicfilter** reinicie o daemon de impressão (se estiver usando a **Debian**, entre no diretório `/etc/init.d` e como usuário `root` digite `./lpr restart` ou `./lprng restart`).

Para testar o funcionamento do **magicfilter**, digite `lpr alphabet.ps` e `lpr tiger.ps`, os arquivos serão enviados para o **magicfilter** que identificará o arquivo como *Pos Script*, executará o Ghost Script e retornará o resultado do processamento para o daemon de impressão. O resultado será visto na impressora.

Se tiver problemas, verifique se a configuração feita com o **magicfilterconfig** está correta. Caso precise re-configurar o **magicfilter**, digite `magicfilterconfig --force` (lembre-se que a opção `--force` substitui qualquer configuração personalizada que tenha adicionado ao arquivo `/etc/printcap`).

Outros detalhes técnicos sobre o Magic Filter

Durante a configuração do **magicfilter**, a seguinte linha é adicionada ao arquivo `/etc/printcap`:

```
:if=/etc/magicfilter/epson9-filter
```

Não tenho nenhum contrato de divulgação com a *epson* :-). Estou usando esta marca de impressora porque é a mais tradicional e facilmente encontrada. A linha que começa com `:if` no **magicfilter** identifica um arquivo de filtro de impressão.

O arquivo `/etc/magicfilter/epson9-filter` é criado usando o formato do `magicfilter`, e não é difícil entender seu conteúdo e fazer algumas modificações:

```
#!/usr/sbin/magicfilter
#
# Magic filter setup file for 9-pin Epson (or compatible) printers
#
# This file is in the public domain.
#
# This file has been automatically adapted to your system.
```

```

#
# wild guess: native control codes start with ESC
0      \033          cat

# PostScript
0 %! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -sDEVICE=epson -sOutputFile=
0 \004%! filter /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -sDEVICE=epson -sOutputF

# PDF
0 %PDF fpipe /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -sDEVICE=epson -sOutputFile

# TeX DVI
0 \367\002 fpipe /usr/bin/dvips -X 120 -Y 72 -R -q -f

# compress'd data
0 \037\235 pipe /bin/gzip -cdq

# packed, gzipped, frozen and SCO LZH data
0 \037\036 pipe /bin/gzip -cdq
0 \037\213 pipe /bin/gzip -cdq
0 \037\236 pipe /bin/gzip -cdq
0 \037\240 pipe /bin/gzip -cdq

0 BZh pipe /usr/bin/bzip2 -cdq

# troff documents
0 .\?\?\040 fpipe `/usr/bin/grog -Tps $FILE`
0 .\\\\" fpipe `/usr/bin/grog -Tps $FILE`
0 '\\\\" fpipe `/usr/bin/grog -Tps $FILE`
0 '.\\\\" fpipe `/usr/bin/grog -Tps $FILE`
0 \\\\" fpipe `/usr/bin/grog -Tps $FILE`

```

Você deve ter notado que para cada tipo de arquivo existe o respectivo programa que é executado, basta você modificar as opções usadas nos programas neste arquivo (como faria na linha de comando) para afetar o comportamento da impressão.

Por exemplo, modificando a resolução para -r240x72 no processamento de arquivos Pos Script (gs), a impressora passará a usar esta resolução.

Impressão remota

Aqui será explicado como fazer seu sistema **Linux** atuar como um servidor de impressão para outras máquinas de sua rede.

Dando permissão para impressão remota via lpd/lprng

As máquinas autorizadas a usar a impressora local deverão ter seus nomes incluídos no arquivo `/etc/hosts.lpd` (para o daemon **lpd** padrão) ou `/etc/lprng/lpd.perms` (para o daemon **lpd** do pacote).

O arquivo `/etc/lprng/lpd.perms` do **lprng** é mais configurável (e complexo), uma linha como:

```
ACCEPT HOST=estaca01.dominio.org SERVICE=X,R,P,Q,M,C
```

aceitará os serviços (SERVICE) de conexão (X), lpr (R), impressão de trabalhos (P), lpq (Q), lprm (M) e lpc (C) da máquina `estacao1.dominio.org`. Veja os comentários neste arquivo para entender o funcionamento de suas opções ou a página de manual do `lpd.perms`.

Impressão via rlpd

O **rlpd** redireciona a impressão diretamente ao servidor de impressão. Sua vantagem é que a impressão é feita diretamente sem a necessidade de configurar um arquivo `/etc/printcap` e dispensar trabalhos adicionais de administração. Ele envia o trabalho de impressão diretamente ao daemon **lpd** na porta 515 (a máquina deve estar configurada para aceitar conexões, veja “Dando permissão para impressão remota via lpd/lprng”).

Para enviar o arquivo `listagem.txt` para a impressora `hp` no servidor `impr.meudominio.org`:

```
rlpd -Himpr.meudominio.org -Php listagem.txt
```

A opção `-H` especifica o nome do servidor de impressão e `-P` o nome da impressora. Caso não tenha permissões para imprimir na impressora remota, uma mensagem será mostrada.

Impressão via printcap

Através deste método, a impressão será tratada através do spool remoto (**lpd** ou **lprng**) e enviada ao servidor de impressão. Para que isto funcione, utilize a seguinte configuração no seu arquivo `/etc/printcap`:

```
lp:Impressora remota:\
:sd=/var/spool/lpd/lp:\
:rm=impr.meudominio.org:\
:rp=hp:\
:sh:
```

Então quando for executado o comando **lpr** na máquina remota, o **lprng** enviará a impressão para a impressora `hp` (`rp=hp`) na máquina `impr.meudominio.org` (`rm=impr.meudominio.org`).

Caso você tenha a opção de imprimir tanto para uma impressora local quando para uma remota, você poderá usar uma configuração como a seguinte:

```
lp|hp|Impressora Local:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/hp:\
:sh:\
:pw#80:\
:pl#66: \
:px#1440:\
:mx#0:\
:if=/etc/magicfilter/dj930c-filter:\
:af=/var/log/lp-acct:\
:lf=/var/log/lp-errs:
```

```
hp-r|Impressora Remota:\
:sd=/var/spool/lpd/lp:\
```

```
:rm=impr.meudominio.org:\  
:rp=hp:\  
:sh:
```

Para selecionar qual impressora será usada, adicione a opção *-Pimpressora* na linha de comando dos utilitários **lpr**, **lpq**, **lprm** (por exemplo, `lpr -Php-r relatorio.txt`. Quando a opção *-P* é especificada, a impressora *lp* será usada por padrão.

OBS Lembre-se de reiniciar seu daemon de impressão toda vez que modificar o arquivo `/etc/printcap`.

Capítulo 10. Firewall iptables

Este capítulo documenta o funcionamento do firewall **iptables** que acompanha a série do kernel 2.4, opções usadas, e aponta alguns pontos fundamentais para iniciar a configuração e construção de bons sistemas de firewall.

Introdução

O *Firewall* é um programa que como objetivo proteger a máquina contra acessos indesejados, tráfego indesejado, proteger serviços que estejam rodando na máquina e bloquear a passagem de coisas que você não deseja receber (como conexões vindas da Internet para sua segura rede local, evitando acesso aos dados corporativos de uma empresa ou a seus dados pessoais). No kernel do Linux 2.4, foi introduzido o firewall **iptables** (também chamado de netfilter) que substituiu o **ipchains** dos kernels da série 2.2. Este novo firewall tem como vantagem ser muito estável (assim como o **ipchains** e **ipfwadm**), confiável, permitir muita flexibilidade na programação de regras pelo administrador do sistema, mais opções disponíveis ao administrador para controle de tráfego, controle independente do tráfego da rede local/entre redes/interfaces devido a nova organização das etapas de roteamento de pacotes.

O **iptables** é um firewall em nível de pacotes e funciona baseado no endereço/porta de origem/destino do pacote, prioridade, etc. Ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar. Em firewalls mais restritivos, o pacote é bloqueado e registrado para que o administrador do sistema tenha conhecimento sobre o que está acontecendo em seu sistema.

Ele também pode ser usado para modificar e monitorar o tráfego da rede, fazer NAT (masquerading, source nat, destination nat), redirecionamento de pacotes, marcação de pacotes, modificar a prioridade de pacotes que chegam/saem do seu sistema, contagem de bytes, dividir tráfego entre máquinas, criar proteções anti-spoofing, contra syn flood, DoS, etc. O tráfego vindo de máquinas desconhecidas da rede pode também ser bloqueado/registrado através do uso de simples regras. As possibilidades oferecidas pelos recursos de filtragem **iptables** como todas as ferramentas UNIX maduras dependem de sua imaginação, pois ele garante uma grande flexibilidade na manipulação das regras de acesso ao sistema, precisando apenas conhecer quais interfaces o sistema possui, o que deseja bloquear, o que tem acesso garantido, quais serviços devem estar acessíveis para cada rede, e iniciar a construção de seu firewall.

O **iptables** ainda tem a vantagem de ser modularizável, funções podem ser adicionadas ao firewall ampliando as possibilidades oferecidas. Usei por 2 anos o **ipchains** e afirmo que este é um firewall que tem possibilidades de gerenciar tanto a segurança em máquinas isoladas como roteamento em grandes organizações, onde a passagem de tráfego entre redes deve ser minuciosamente controlada.

Um firewall não funciona de forma automática (instalando e esperar que ele faça as coisas por você), é necessário pelo menos conhecimentos básicos de rede tcp/ip, roteamento e portas para criar as regras que farão a segurança de seu sistema. A segurança do sistema depende do controle das regras que serão criadas por você, as falhas humanas são garantia de mais de 95% de sucesso nas invasões.

Enfim o **iptables** é um firewall que agrada tanto a pessoas que desejam uma segurança básica em seu sistema, quando administradores de grandes redes que querem ter um controle minucioso sobre o tráfego que passam entre suas interfaces de rede (controlando tudo o que pode passar de uma rede a outra), controlar o uso de tráfego, monitoração, etc.

Versão

É assumido que esteja usando a versão 1.2.3 do **iptables** e baseadas nas opções do kernel 2.4.16 (sem o uso de módulos experimentais). As explicações contidas aqui podem funcionar para versões posteriores,

mas é recomendável que leia a documentação sobre modificações no programa (changelog) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

Um resumo da história do iptables

O **iptables** é um código de firewall das versões 2.4 do kernel, que substituiu o **ipchains** (presente nas séries 2.2 do kernel). Ele foi incluído no kernel da série 2.4 em meados de Junho/Julho de 1999.

A história do desenvolvimento (desde o porte do **ipfw** do **BSD** para o **Linux** até o **iptables** (que é a quarta geração de firewalls do kernel) está disponível no documento, `Netfilter-howto`.

Características do firewall iptables

- Especificação de portas/endereço de origem/destino
- Suporte a protocolos TCP/UDP/ICMP (incluindo tipos de mensagens icmp)
- Suporte a interfaces de origem/destino de pacotes
- Manipula serviços de proxy na rede
- Tratamento de tráfego dividido em chains (para melhor controle do tráfego que entra/sai da máquina e tráfego redirecionado).
- Permite um número ilimitado de regras por chain
- Muito rápido, estável e seguro
- Possui mecanismos internos para rejeitar automaticamente pacotes duvidosos ou mal formados.
- Suporte a módulos externos para expansão das funcionalidades padrões oferecidas pelo código de firewall
- Suporte completo a roteamento de pacotes, tratadas em uma área diferente de tráfegos padrões.
- Suporte a especificação de tipo de serviço para priorizar o tráfego de determinados tipos de pacotes.
- Permite especificar exceções para as regras ou parte das regras
- Suporte a detecção de fragmentos
- Permite enviar alertas personalizados ao **syslog** sobre o tráfego aceito/bloqueado.
- Redirecionamento de portas
- Masquerading
- Suporte a SNAT (modificação do endereço de origem das máquinas para um único IP ou faixa de IP's).
- Suporte a DNAT (modificação do endereço de destino das máquinas para um único IP ou faixa de IP's)
- Contagem de pacotes que atravessaram uma interface/regra
- Limitação de passagem de pacotes/conferência de regra (muito útil para criar proteções contra, syn flood, ping flood, DoS, etc).

Ficha técnica

Pacote:

- `iptables` - Sistema de controle principal para protocolos ipv4
- `ip6tables` - Sistema de controle principal para protocolos ipv6
- `iptables-save` - Salva as regras atuais em um arquivo especificado como argumento. Este utilitário pode ser dispensado por um shell script contendo as regras executado na inicialização da máquina.
- `iptables-restore` - Restaura regras salvas pelo utilitário **iptables-save**.

Requerimentos

É necessário que o seu kernel tenha sido compilado com suporte ao **iptables** (veja “Habilitando o suporte ao iptables no kernel”. O requerimento mínimo de memória necessária para a execução do **iptables** é o mesmo do kernel 2.4 (4MB). Dependendo do tráfego que será manipulado pela(s) interface(s) do firewall ele poderá ser executado com folga em uma máquina 386 SX com 4MB de RAM.

Como as configurações residem no kernel não é necessário espaço extra em disco rígido para a execução deste utilitário.

Arquivos de logs criados pelo iptables

Todo tráfego que for registrado pelo **iptables** é registrado por padrão no arquivo `/var/log/kern.log`.

Instalação

```
apt-get install iptables
```

O pacote contém o utilitário **iptables** (e **ip6tables** para redes ipv6) necessários para inserir suas regras no kernel. Se você não sabe o que é ipv6, não precisará se preocupar com o utilitário **ip6tables** por enquanto.

Enviando Correções/Contribuindo com o projeto

A página principal do projeto é <http://netfilter.filewatcher.org>. Sugestões podem ser enviadas para a lista de desenvolvimento oficial do **iptables**: <http://lists.samba.org>.

O que aconteceu com o ipchains e ipfwadm?

O **iptables** faz parte da nova geração de firewalls que acompanha o kernel 2.4, mas o suporte ao **ipchains** e **ipfwadm** ainda será mantido através de módulos de compatibilidade do kernel até 2004. Seria uma grande falta de consideração retirar o suporte a estes firewalls do kernel como forma de obrigar a "aprenderem" o **iptables** (mesmo o suporte sendo removido após este período, acredito que criarão patches "externos" para futuros kernels que não trarão mais este suporte). Se precisa do suporte a estes firewalls antes de passar em definitivo para o **iptables** leia “Habilitando o suporte ao iptables no kernel”.

Se você é um administrador que gosta de explorar todos os recursos de um firewall, usa todos os recursos que ele oferece ou mantém uma complexa rede corporativa, tenho certeza que gostará do **iptables**.

Tipos de firewalls

Existem basicamente dois tipos de firewalls:

- nível de aplicação - Este tipo de firewall analisam o conteúdo do pacote para tomar suas decisões de filtragem. Firewalls deste tipo são mais intrusivos (pois analisam o conteúdo de tudo que passa por ele) e permitem um controle relacionado com o conteúdo do tráfego. Alguns firewalls em

nível de aplicação combinam recursos básicos existentes em firewalls em nível de pacotes combinando as funcionalidade de controle de tráfego/controle de acesso em uma só ferramenta. Servidores proxy, como o **squid**, são um exemplo deste tipo de firewall.

- nível de pacotes - Este tipo de firewall toma as decisões baseadas nos parâmetros do pacote, como porta/endereço de origem/destino, estado da conexão, e outros parâmetros do pacote. O firewall então pode negar o pacote (DROP) ou deixar o pacote passar (ACCEPT). O **iptables** é um excelente firewall que se encaixa nesta categoria.

Firewall em nível de pacotes é o assunto explicado nesta seção do guia mas será apresentada uma explicação breve sobre o funcionamento de análise de strings do **iptables**.

Os dois tipos de firewalls podem ser usados em conjunto para fornecer uma camada dupla de segurança no acesso as suas máquinas/máquinas clientes.

O que proteger?

Antes de iniciar a construção do firewall é bom pensar nos seguintes pontos:

- Quais serviços precisa proteger. Serviços que devem ter acesso garantido a usuários externos e quais serão bloqueados a todas/determinadas máquinas. É recomendável bloquear o acesso a todas portas menores que 1024 por executarem serviços que rodam com privilégio de usuário `root`, e autorizar somente o acesso as portas que realmente deseja (configuração restritiva nesta faixa de portas).
- Que tipo de conexões eu posso deixar passar e quais bloquear. Serviços com autenticação em texto plano e potencialmente inseguros como `rlogin`, `telnet`, `ftp`, `NFS`, `DNS`, `LDAP`, `SMTP RCP`, `X-Window` são serviços que devem ser ter acesso garantido somente para máquinas/redes que você confia. Estes serviços podem não ser só usados para tentativa de acesso ao seu sistema, mas também como forma de atacar outras pessoas aproveitando-se de problemas de configuração.

A configuração do firewall ajuda a prevenir isso, mesmo se um serviço estiver mal configurado e tentando enviar seus pacotes para fora, será impedido. Da mesma forma se uma máquina Windows de sua rede for infectada por um trojan não haverá pânico: o firewall poderá estar configurado para bloquear qualquer tentativa de conexão vinda da internet (cracker) para as máquinas de sua rede.

Para cópia de arquivos via rede insegura (como através da Internet), é recomendado o uso de serviços que utilizam criptografia para login e transferência de arquivos (veja ???) ou a configuração de uma VPN.

- Que máquinas terão acesso livre e quais serão restritas.
- Que serviços deverão ter prioridade no processamento.
- Que máquinas/redes NUNCA deverão ter acesso a certas/todas máquinas.
- O volume de tráfego que o servidor manipulará. Através disso você pode ter que balancear o tráfego entre outras máquinas, configurar proteções contra DoS, syn flood, etc.
- O que tem permissão de passar de uma rede para outra (em máquinas que atuam como roteadores/gateways de uma rede interna).
- Etc.

A análise destes pontos pode determinar a complexidade do firewall, custos de implementação, prazo de desenvolvimento e tempo de maturidade do código para implementação. Existem muitos outros pontos que podem entrar na questão de desenvolvimento de um sistema de firewall, eles dependem do tipo de firewall que está desenvolvendo e das políticas de segurança de sua rede.

O que são regras?

As regras são como comandos passados ao **iptables** para que ele realize uma determinada ação (como bloquear ou deixar passar um pacote) de acordo com o endereço/porta de origem/destino, interface de origem/destino, etc. As regras são armazenadas dentro dos chains e processadas na ordem que são inseridas.

As regras são armazenadas no kernel, o que significa que quando o computador for reiniciado tudo o que fez será perdido. Por este motivo elas deverão ser gravadas em um arquivo para serem carregadas a cada inicialização.

Um exemplo de regra: `iptables -A INPUT -s 123.123.123.1 -j DROP`.

O que são chains?

Os *Chains* são locais onde as regras do firewall definidas pelo usuário são armazenadas para operação do firewall. Existem dois tipos de chains: os embutidos (como os chains *INPUT*, *OUTPUT* e *FORWARD*) e os criados pelo usuário. Os nomes dos chains embutidos devem ser especificados sempre em maiúsculas (note que os nomes dos chains são case-sensitive, ou seja, o chain `input` é completamente diferente de `INPUT`).

O que são tabelas?

Tabelas são os locais usados para armazenar os chains e conjunto de regras com uma determinada característica em comum. As tabelas podem ser referenciadas com a opção *-t tabela* e existem 3 tabelas disponíveis no **iptables**:

- `filter` - Esta é a tabela padrão, contém 3 chains padrões:
 - `INPUT` - Consultado para dados que chegam a máquina
 - `OUTPUT` - Consultado para dados que saem da máquina
 - `FORWARD` - Consultado para dados que são redirecionados para outra interface de rede ou outra máquina.

Os chains *INPUT* e *OUTPUT* somente são atravessados por conexões indo/se originando de localhost.

OBS: Para conexões locais, somente os chains *INPUT* e *OUTPUT* são consultados na tabela `filter`.

- `nat` - Usada para dados que gera outra conexão (masquerading, source nat, destination nat, port forwarding, proxy transparente são alguns exemplos). Possui 3 chains padrões:
 - `PREROUTING` - Consultado quando os pacotes precisam ser modificados logo que chegam. É o chain ideal para realização de DNAT e redirecionamento de portas (“Fazendo DNAT”).
 - `OUTPUT` - Consultado quando os pacotes gerados localmente precisam ser modificados antes de serem roteados. Este chain somente é consultado para conexões que se originam de IPs de interfaces locais.
 - `POSTROUTING` - Consultado quando os pacotes precisam ser modificados após o tratamento de roteamento. É o chain ideal para realização de SNAT e IP Masquerading (“Fazendo SNAT”).
- `mangle` - Utilizada para alterações especiais de pacotes (como modificar o tipo de serviço (TOS) ou outros detalhes que serão explicados no decorrer do capítulo. Possui 2 chains padrões:

- INPUT - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain *INPUT* da tabela *filter*.
- FORWARD - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain *FORWARD* da tabela *filter*.
- PREROUTING - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain *PREROUTING* da tabela *nat*.
- POSTROUTING - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain *POSTROUTING* da tabela *nat*.
- OUTPUT - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain *OUTPUT* da tabela *nat*.

Veja “A tabela mangle” para mais detalhes sobre a tabela mangle.

Habilitando o suporte ao iptables no kernel

Para usar toda a funcionalidade do firewall **iptables**, permitindo fazer o controle do que tem ou não permissão de acessar sua máquina, fazer Masquerading/NAT em sua rede, etc., você precisará dos seguintes componentes compilados em seu kernel (os módulos experimentais fora ignorados intencionalmente):

```
*
* Network Options:
*
```

```
Network packet filtering (replaces ipchains) [Y/m/n/?]
Network packet filtering debugging [Y/m/n/?]
```

e na Subseção:

```
*
* IP: Netfilter Configuration
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [M/n/y/?]
  FTP protocol support (CONFIG_IP_NF_FTP) [M/n/?]
  IRC protocol support (CONFIG_IP_NF_IRC) [M/n/?]
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [Y/m/n/?]
  limit match support (CONFIG_IP_NF_MATCH_LIMIT) [Y/m/n/?]
  MAC address match support (CONFIG_IP_NF_MATCH_MAC) [M/n/y/?]
  netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [M/n/y/?]
  Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [M/n/y/?]
  TOS match support (CONFIG_IP_NF_MATCH_TOS) [M/n/y/?]
  LENGTH match support (CONFIG_IP_NF_MATCH_LENGTH) [M/n/y/?]
  TTL match support (CONFIG_IP_NF_MATCH_TTL) [M/n/y/?]
  tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) [M/n/y/?]
  Connection state match support (CONFIG_IP_NF_MATCH_STATE) [M/n/?]
  Packet filtering (CONFIG_IP_NF_FILTER) [M/n/y/?]
    REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [M/n/?]
  Full NAT (CONFIG_IP_NF_NAT) [M/n/?]
    MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE) [M/n/?]
```

```
REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT) [M/n/?]
Packet mangling (CONFIG_IP_NF_MANGLE) [M/n/y/?]
TOS target support (CONFIG_IP_NF_TARGET_TOS) [M/n/?]
MARK target support (CONFIG_IP_NF_TARGET_MARK) [M/n/?]
LOG target support (CONFIG_IP_NF_TARGET_LOG) [M/n/y/?]
TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) [M/n/y/?]
```

Esta configuração permite que você não tenha problemas para iniciar o uso e configuração do seu firewall iptables, ela ativa os módulos necessários para utilização de todos os recursos do firewall **iptables**. Quando conhecer a função de cada um dos parâmetros acima (durante o decorrer do texto), você poderá eliminar muitas das opções desnecessárias para seu estilo de firewall ou continuar fazendo uso de todas ;-)

OBS1: A configuração acima leva em consideração que você **NÃO** executará os códigos antigos de firewall **ipfwadm** e **ipchains**. Caso deseje utilizar o **ipchains** ou o **ipfwadm**, será preciso responder com "M" a questão "IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES)". Será necessário carregar manualmente o módulo correspondente ao firewall que deseja utilizar (`modprobe iptables_filter.o` no caso do **iptables**).

Não execute mais de um tipo de firewall ao mesmo tempo!!!

OBS2: É recomendável ativar o daemon **kmod** para carga automática de módulos, caso contrário será necessário compilar todas as partes necessárias embutidas no kernel, carregar os módulos necessários manualmente ou pelo **iptables** (através da opção `--modprobe=módulo`).

Ligando sua rede interna a Internet

Se a sua intenção (como da maioria dos usuários) é conectar sua rede interna a Internet de forma rápida e simples, leia "Fazendo IP masquerading (para os apressados)" ou "Fazendo SNAT". Um exemplo prático de configuração de Masquerading deste tipo é encontrado em "Conectando sua rede interna a Internet".

Após configurar o masquerading, você só precisará especificar o endereço IP da máquina masquerading (servidor) como *Gateway* da rede. No **Windows 9x/NT/2000** isto é feito no Painel de Controle/Rede/Propriedades de Tcp/IP. No **Linux** pode ser feito com `route add default gw IP_do_Servidor`.

Manipulando chains

O **iptables** trabalha com uma tabela de regras que é analisada uma a uma até que a última seja processada. Por padrão, se uma regra tiver qualquer erro, uma mensagem será mostrada e ela descartada. O pacote não conferirá e a ação final (se ele vai ser aceito ou rejeitado) dependerá das regras seguintes.

As opções passadas ao **iptables** usadas para manipular os chains são **SEMPRE** em maiúsculas. As seguintes operações podem ser realizadas:

Adicionando regras - A

Como exemplo vamos criar uma regra que bloqueia o acesso a nossa própria máquina (127.0.0.1 - loopback). Primeiro daremos um ping para verificar seu funcionamento:

```
#ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=0.6 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.5 ms
```

```
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.5/0.6 ms
```

Ok, a máquina responde, agora vamos incluir uma regra no chain INPUT (*-A INPUT*) que bloqueie (*-j DROP*) qualquer acesso indo ao endereço 127.0.0.1 (*-d 127.0.0.1*):

```
iptables -t filter -A INPUT -d 127.0.0.1 -j DROP
```

Agora verificamos um novo ping:

```
#ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

```
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
```

Desta vez a máquina 127.0.0.1 não respondeu, pois todos os pacotes com o destino 127.0.0.1 (*-d 127.0.0.1*) são rejeitados (*-j DROP*). A opção *-A* é usada para adicionar novas regras no final do chain. Além de *-j DROP* que serve para rejeitar os pacotes, podemos também usar *-j ACCEPT* para aceitar pacotes. A opção *-j* é chamada de *alvo da regra* ou somente *alvo* pois define o destino do pacote que atravessa a regra (veja “Especificando um alvo”). Bem vindo a base de um sistema de firewall :-)

OBS1: - O acesso a interface loopback não deve ser de forma alguma bloqueado, pois muitos aplicativos utilizam soquetes tcp para realizarem conexões, mesmo que você não possua uma rede interna.

OBS2: - A tabela *filter* será usada como padrão caso nenhuma tabela seja especificada através da opção *-t*.

Listando regras - L

A seguinte sintaxe é usada para listar as regras criadas:

```
iptables [-t tabela] -L [chain] [opções]
```

Onde:

tabela É uma das tabelas usadas pelo **iptables**. Se a tabela não for especificada, a tabela *filter* será usada como padrão. Veja “O que são tabelas?” para detalhes.

chain Um dos chains disponíveis na tabela acima (veja “O que são tabelas?”) ou criado pelo usuário (“Criando um novo chain - N”). Caso o chain não seja especificado, todos os chains da tabela serão mostrados.

opções As seguintes opções podem ser usadas para listar o conteúdo de chains:

- *-v* - Exibe mais detalhes sobre as regras criadas nos chains.
- *-n* - Exibe endereços de máquinas/portas como números ao invés de tentar a resolução DNS e consulta ao */etc/services*. A resolução de nomes pode tomar muito tempo dependendo da quantidade de regras que suas tabelas possuem e velocidade de sua conexão.
- *-x* - Exibe números exatos ao invés de números redondos. Também mostra a faixa de portas de uma regra de firewall.
- *--line-numbers* - Exibe o número da posição da regra na primeira coluna da listagem.

Para listar a regra criada anteriormente usamos o comando:

```
#iptables -t filter -L INPUT
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      all  --  anywhere              localhost
```

O comando `iptables -L INPUT -n` tem o mesmo efeito, a diferença é que são mostrados números ao invés de nomes:

```
#iptables -L INPUT -n
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      all  --  0.0.0.0/0            127.0.0.1
```

```
#iptables -L INPUT -n --line-numbers
```

```
Chain INPUT (policy ACCEPT)
num target     prot opt source                destination
1  DROP      all  --  0.0.0.0/0            127.0.0.1
```

```
#iptables -L INPUT -n -v
```

```
Chain INPUT (policy ACCEPT 78 packets, 5820 bytes)
  pkts bytes target     prot opt in     out     source                destination
    2  194 DROP      icmp  --  *      *      0.0.0.0/0            127.0.0.1
```

Os campos assim possuem o seguinte significado:

Chain INPUT	Nome do chain listado
(policy ACCEPT 78 packets, 5820 bytes)	política padrão do chain (veja “Especificando a política padrão de um chain - P”).
pkts	Quantidade de pacotes que atravessaram a regra (veja “Zerando contador de bytes dos chains - Z”).
bytes	Quantidade de bytes que atravessaram a regra. Pode ser referenciado com K (Kilobytes), M (Megabytes), G (Gigabytes).
target	O alvo da regra, o destino do pacote. Pode ser ACCEPT, DROP ou outro chain. Veja “Especificando um alvo” para detalhes sobre a especificação de um alvo.
prot	Protocolo especificado pela regra. Pode ser udp, tcp, icmp ou all. Veja “Especificando um protocolo” para detalhes.
opt	Opções extras passadas a regra. Normalmente “!” (veja “Especificando uma exceção”) ou “f” (veja “Especificando fragmentos”).
in	Interface de entrada (de onde os dados chegam). Veja “Especificando a interface de origem/destino”.

out	Interface de saída (para onde os dados vão). Veja “Especificando a interface de origem/destino”.
source	Endereço de origem. Veja “Especificando um endereço de origem/destino”.
destination	Endereço de destino. Veja “Especificando um endereço de origem/destino”.
outras opções	Estas opções normalmente aparecem quando são usadas a opção -x: <ul style="list-style-type: none">• <code>dpt</code> ou <code>dpts</code> - Especifica a porta ou faixa de portas de destino.• <code>reject-with icmp-port-unreachable</code> - Significa que foi usado o alvo REJECT naquela regra (veja “Alvo REJECT”).

Apagando uma regra - D

Para apagar um chain, existem duas alternativas:

1. Quando sabemos qual é o número da regra no chain (listado com a opção `-L`) podemos referenciar o número diretamente. Por exemplo, para apagar a regra criada acima:

```
iptables -t filter -D INPUT 1
```

Esta opção não é boa quando temos um firewall complexo com um grande número de regras por chains, neste caso a segunda opção é a mais apropriada.

2. Usamos a mesma sintaxe para criar a regra no chain, mas trocamos `-A` por `-D`:

```
iptables -t filter -D INPUT -d 127.0.0.1 -j DROP
```

Então a regra correspondentes no chain INPUT será automaticamente apagada (confira listando o chain com a opção `-L`). Caso o chain possua várias regras semelhantes, somente a primeira será apagada.

OBS: Não é possível apagar os chains defaults do **iptables** (*INPUT, OUTPUT...*).

Inserindo uma regra - I

Precisamos que o tráfego vindo de `192.168.1.15` não seja rejeitado pelo nosso firewall. Não podemos adicionar uma nova regra (`-A`) pois esta seria incluída no final do chain e o tráfego seria rejeitado pela primeira regra (nunca atingindo a segunda). A solução é inserir a nova regra antes da regra que bloqueia todo o tráfego ao endereço `127.0.0.1` na posição 1:

```
iptables -t filter -I INPUT 1 -s 192.168.1.15 -d 127.0.0.1 -j ACCEPT
```

Após este comando, temos a regra inserida na primeira posição do chain (repare no número 1 após INPUT) e a antiga regra número 1 passa a ser a número 2. Desta forma a regra acima será consultada, se a máquina de origem for `192.168.1.15` então o tráfego estará garantido, caso contrário o tráfego com o destino `127.0.0.1` será bloqueado na regra seguinte.

Substituindo uma regra - R

Após criar nossa regra, percebemos que a nossa intenção era somente bloquear os pings com o destino `127.0.0.1` (pacotes ICMP) e não havia necessidade de bloquear todo o tráfego da máquina. Existem

duas alternativas: apagar a regra e inserir uma nova no lugar ou modificar diretamente a regra já criada sem afetar outras regras existentes e mantendo a sua ordem no chain (isso é muito importante). Use o seguinte comando:

```
iptables -R INPUT 2 -d 127.0.0.1 -p icmp -j DROP
```

O número 2 é o número da regra que será substituída no chain INPUT, e deve ser especificado. O comando acima substituirá a regra 2 do chain INPUT (-R INPUT 2) bloqueando (-j DROP) qualquer pacote icmp (-p icmp) com o destino 127.0.0.1 (-d 127.0.0.1).

Criando um novo chain - N

Em firewalls organizados com um grande número de regras, é interessante criar chains individuais para organizar regras de um mesmo tipo ou que tenha por objetivo analisar um tráfego de uma mesma categoria (interface, endereço de origem, destino, protocolo, etc) pois podem consumir muitas linhas e tornar o gerenciamento do firewall confuso (e conseqüentemente causar sérios riscos de segurança). O tamanho máximo de um nome de chain é de 31 caracteres e podem conter tanto letras maiúsculas quanto minúsculas.

```
iptables [-t tabela] [-N novochain]
```

Para criar o chain *internet* (que pode ser usado para agrupar as regras de internet) usamos o seguinte comando:

```
iptables -t filter -N internet
```

Para inserir regras no chain *internet* basta especifica-lo após a opção -A:

```
iptables -t filter -A internet -s 200.200.200.200 -j DROP
```

E então criamos um pulo (-j) do chain *INPUT* para o chain *internet*:

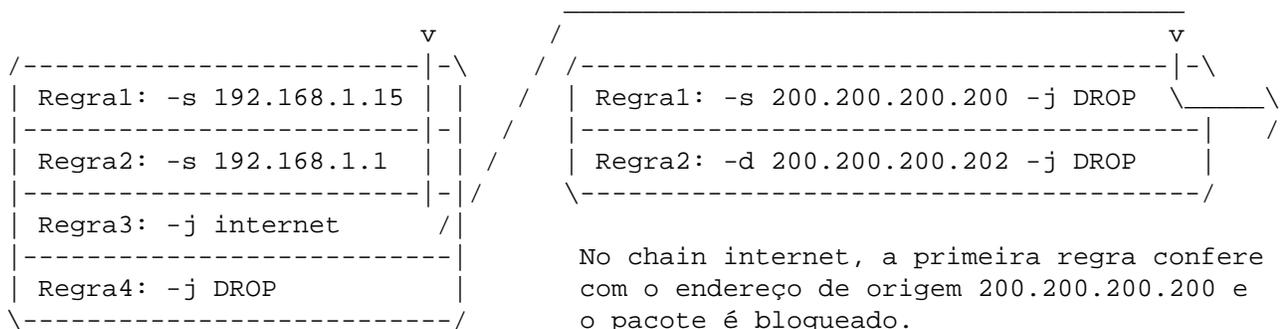
```
iptables -t filter -A INPUT -j internet
```

OBS: O chain criando pelo usuário pode ter seu nome tanto em maiúsculas como minúsculas.

Se uma máquina do endereço 200.200.200.200 tentar acessar sua máquina, o **iptables** consultará as seguintes regras:

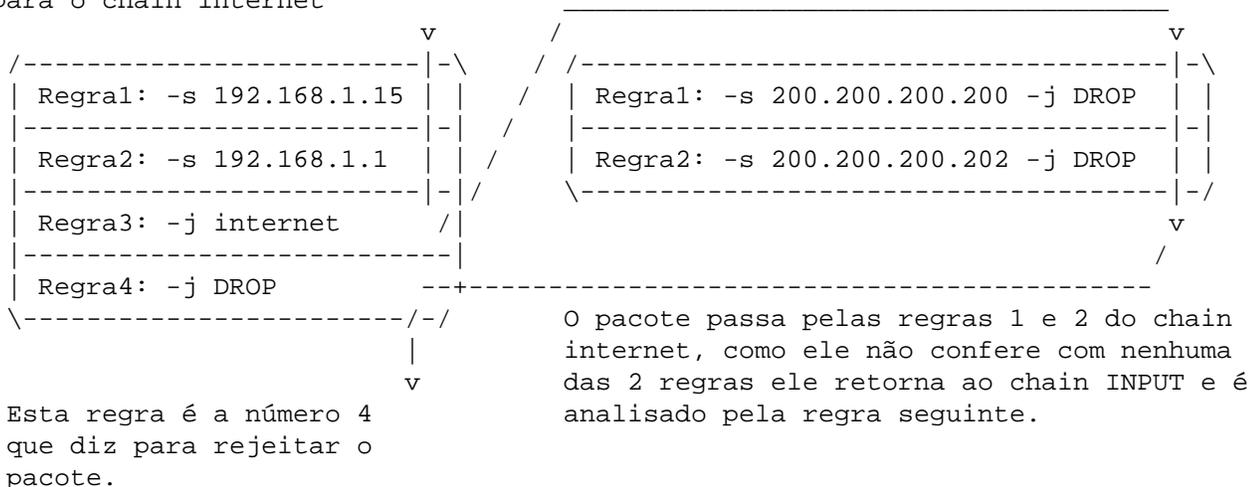
`INPUT'	`internet'
Regra1: -s 192.168.1.15	Regra1: -s 200.200.200.200
Regra2: -s 192.168.1.1	Regra2: -d 192.168.1.1
Regra3: -j DROP	

O pacote tem o endereço de origem 200.200.200.200, ele passa pela primeira e segunda regras do chain INPUT, a terceira regra direciona para o chain internet



Se uma máquina com o endereço de origem `200.200.200.201` tentar acessar a máquina, então as regra consultadas serão as seguintes:

O pacote tem o endereço de origem `200.200.200.201`, ele passa pela primeira e segunda regras do chain `INPUT`, a terceira regra direciona para o chain `internet`



Renomeando um chain criado pelo usuário - E

Se por algum motivo precisar renomear um chain criado por você na tabela `filter`, `nat` ou `mangle`, isto poderá ser feito usando a opção `-E` do `iptables`:

```
iptables -t filter -E chain-antigo novo-chain
```

Note que não é possível renomear os chains defaults do `iptables`.

Listando os nomes de todas as tabelas atuais

Use o comando `cat /proc/net/ip_tables_names` para fazer isto. É interessante dar uma olhada nos arquivos dentro do diretório `/proc/net`, pois os arquivos existentes podem lhe interessar para outras finalidades.

Limpendo as regras de um chain - F

Para limpar todas as regras de um chain, use a seguinte sintaxe:

```
iptables [-t tabela] [-F chain]
```

Onde:

tabela Tabela que contém o chain que desejamos zerar.

chain Chain que desejamos limpar. Caso um chain não seja especificado, todos os chains da tabela serão limpos.

```
iptables -t filter -F INPUT
iptables -t filter -F
```

Apagando um chain criado pelo usuário - X

Para apagarmos um chain criado pelo usuário, usamos a seguinte sintaxe:

```
iptables [-t tabela] [-X chain]
```

Onde:

tabela Nome da tabela que contém o chain que desejamos excluir.

chain Nome do chain que desejamos apagar. Caso não seja especificado, todos os chains definidos pelo usuário na tabela especificada serão excluídos.

OBS: - Chains embutidos nas tabelas não podem ser apagados pelo usuário. Veja os nomes destes chains em “O que são tabelas?”.

```
iptables -t filter -X internet
iptables -X
```

Zerando contador de bytes dos chains - Z

Este comando zera o campo *pkts* e *bytes* de uma regra do **iptables**. Estes campos podem ser visualizados com o comando **iptables -L -v**. A seguinte sintaxe é usada:

```
iptables [-t tabela] [-Z chain] [-L]
```

Onde:

tabela Nome da tabela que contém o chain que queremos zerar os contadores de bytes e pacotes.

chain Chain que deve ter os contadores zerados. Caso não seja especificado, todos os chains da tabela terão os contadores zerados. Note que as opções *-Z* e *-L* podem ser usadas juntas, assim o chain será listado e imediatamente zerado. Isto evita a passagem de pacotes durante a listagem de um chain.

```
iptables -t filter -Z INPUT
```

Especificando a política padrão de um chain - P

A política padrão determina o que acontecerá com um pacote quando ele chegar ao final das regras contidas em um chain. A política padrão do **iptables** é "ACCEPT" mas isto pode ser alterado com o comando:

```
iptables [-t tabela] [-P chain] [ACCEPT/DROP]
```

Onde:

tabela	Tabela que contém o chain que desejamos modificar a política padrão.
chain	Define o chain que terá a política modificada. O chain deve ser especificado.
ACCEPT/ DROP	ACCEPT aceita os pacotes caso nenhuma regra do chain conferir (usado em regras permissivas). DROP rejeita os pacotes caso nenhuma regra do chain conferir (usado em regras restritivas).

A política padrão de um chain é mostrada com o comando **iptables -L**:

```
# iptables -L INPUT

Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      icmp -- anywhere             localhost
```

No exemplo acima, a política padrão de INPUT é ACCEPT (policy ACCEPT), o que significa que qualquer pacote que não seja rejeitado pela regra do chain, será aceito. Para alterar a política padrão deste chain usamos o comando:

```
iptables -t filter -P INPUT DROP
```

NOTA: As políticas de acesso PERMISSIVASS (ACCEPT) normalmente são usadas em conjunto com regras restritivas no chain correspondentes (tudo é bloqueado e o que sobrar é liberado) e políticas RESTRITIVAS (DROP) são usadas em conjunto com regras permissivas no chain correspondente (tudo é liberado e o que sobrar é bloqueado pela política padrão).

Outras opções do iptables

Especificando um endereço de origem/destino

As opções `-s` (ou `--src/--source`) e `-d` (ou `--dst/--destination`) servem para especificar endereços de *origem* e *destino* respectivamente. É permitido usar um endereço IP completo (como 192.168.1.1), um hostname (debian), um endereço fqdn (www.debian.org) ou um par *rede/máscara* (como 200.200.200.0/255.255.255.0 ou 200.200.200.0/24).

Caso um endereço/máscara não sejam especificados, é assumido 0/0 como padrão (todas as máquinas de todas as redes). A interpretação dos endereços de origem/destino dependem do chain que está sendo especificado (como INPUT e OUTPUT por exemplo).

OBS: Caso seja especificado um endereço fqdn e este resolver mais de um endereço IP, serão criadas várias regras, cada uma se aplicando a este endereço IP específico. É recomendável sempre que possível a especificação de endereços IP's nas regras, pois além de serem muito rápidos (pois não precisar de

resolução DNS) são mais seguros para evitar que nosso firewall seja enganado por um ataque de IP spoofing.

```
# Bloqueia o tráfego vindo da rede 200.200.200.*:
iptables -A INPUT -s 200.200.200.0/24 -j DROP

# Bloqueia conexões com o destino 10.1.2.3:
iptables -A OUTPUT -d 10.1.2.3 -j DROP

# Bloqueia o tráfego da máquina www.dominio.teste.org a rede 210.21.1.3
# nossa máquina possui o endereço 210.21.1.3
iptables -A INPUT -s www.dominio.teste.org -d 210.21.1.3 -j DROP
```

Especificando a interface de origem/destino

As opções *-i* (ou *--in-interface*) e *-o* (ou *--out-interface*) especificam as interfaces de origem/destino de pacotes. Nem todas as chains aceitam as interfaces de origem/destino simultaneamente, a interface de entrada (*-i*) nunca poderá ser especificada em um chain OUTPUT e a interface de saída (*-o*) nunca poderá ser especificada em um chain INPUT. Abaixo uma rápida referência:

TABELA	CHAIN	INTERFACE	
		ENTRADA (-i)	SAÍDA (-o)
filter	INPUT	SIM	NÃO
	OUTPUT	NÃO	SIM
	FORWARD	SIM	SIM
nat	PREROUTING	SIM	NÃO
	OUTPUT	NÃO	SIM
	POSTROUTING	NÃO	SIM
mangle	PREROUTING	SIM	NÃO
	OUTPUT	NÃO	SIM

O caminho do pacote na interface será determinado pelo tipo da interface e pela posição dos chains nas etapas de seu roteamento. O chain OUTPUT da tabela filter somente poderá conter a interface de saída (veja a tabela acima). O chain FORWARD da tabela filter é o único que aceita a especificação de ambas as interfaces, este é um ótimo chain para controlar o tráfego que passa entre interfaces do firewall.

Por exemplo para bloquear o acesso do tráfego de qualquer máquina com o endereço 200.123.123.10 vinda da interface ppp0 (uma placa de fax-modem):

```
iptables -A INPUT -s 200.123.123.10 -i ppp0 -j DROP
```

A mesma regra pode ser especificada como

```
iptables -A INPUT -s 200.123.123.10 -i ppp+ -j DROP
```

O sinal de "+" funciona como um coringa, assim a regra terá efeito em qualquer interface de ppp0 a ppp9. As interfaces ativas no momento podem ser listadas com o comando **ifconfig**, mas é permitido especificar uma regra que faz referência a uma interface que ainda não existe, isto é interessante para conexões intermitentes como o PPP. Para bloquear qualquer tráfego local para a Internet:

```
iptables -A OUTPUT -o ppp+ -j DROP
```

Para bloquear a passagem de tráfego da interface ppp0 para a interface eth1 (de uma de nossas redes internas):

```
iptables -A FORWARD -i ppp0 -o eth1 -j DROP
```

Especificando um protocolo

A opção `-p` (ou `--protocol`) é usada para especificar protocolos no **iptables**. Podem ser especificados os protocolos *tcp*, *udp* e *icmp*. Por exemplo, para rejeitar todos os pacotes UDP vindos de 200.200.200.200:

```
iptables -A INPUT -s 200.200.200.200 -p udp -j DROP
```

OBS1: Tanto faz especificar os nomes de protocolos em maiúsculas ou minúsculas.

Especificando portas de origem/destino

As portas de origem/destino devem ser especificadas após o protocolo e podem ser precedidas por uma das seguintes opções:

- `--source-port` ou `--sport` - Especifica uma porta ou faixa de portas de origem.
- `--destination-port` ou `--dport` - Especifica uma porta ou faixa de portas de destino.

Uma faixa de portas pode ser especificada através de `PortaOrigem:PortaDestino`:

```
# Bloqueia qualquer pacote indo para 200.200.200.200 na faixa de
# portas 0 a 1023
iptables -A OUTPUT -d 200.200.200.200 -p tcp --dport :1023 -j DROP
```

Caso a *PortaOrigem* de uma faixa de portas não seja especificada, 0 é assumida como padrão, caso a *PortaDestino* não seja especificada, 65535 é assumida como padrão. Caso precise especificar diversas regras que envolvam o tratamento de portas diferentes, recomendo da uma olhada em “Especificando múltiplas portas de origem/destino”, antes de criar um grande número de regras.

Especificando mensagens do protocolo ICMP

O protocolo ICMP não possui portas, mas é possível fazer um controle maior sobre o tráfego ICMP que entra/sai da rede através da especificação dos tipos de mensagens ICMP. Os tipos de mensagens devem ser especificados com a opção `--icmp-type CódigoICMP` logo após a especificação do protocolo `icmp`:

```
iptables -A INPUT -s 200.123.123.10 -p icmp --icmp-type time-exceeded -i ppp+ -j D
```

A regra acima rejeitará mensagens ICMP do tipo "time-exceeded" (tempo de requisição excedido) que venham do endereço 200.123.123.10 através da interface *ppp+*.

Alguns tipos de mensagens ICMP são classificados por categoria (como o próprio "time-exceeded"), caso a categoria "time-exceeded" seja especificada, todas as mensagens daquela categoria (como "ttl-zero-during-transit", "ttl-zero-during-reassembly") conferirão na regra especificada. Os tipos de mensagens ICMP podem ser obtidos com o comando `iptables -p icmp -h`:

```
echo-reply (pong)
destination-unreachable
  network-unreachable
  host-unreachable
  protocol-unreachable
  port-unreachable
  fragmentation-needed
  source-route-failed
  network-unknown
  host-unknown
  network-prohibited
  host-prohibited
  TOS-network-unreachable
  TOS-host-unreachable
  communication-prohibited
  host-precedence-violation
  precedence-cutoff
source-quench
redirect
  network-redirect
  host-redirect
  TOS-network-redirect
  TOS-host-redirect
echo-request (ping)
router-advertisement
router-solicitation
time-exceeded (ttl-exceeded)
  ttl-zero-during-transit
  ttl-zero-during-reassembly
parameter-problem
  ip-header-bad
  required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply
```

OBS1: Não bloqueie mensagens do tipo "host-unreachable" e "source-quench", pois terá sérios problemas no controle de suas conexões. A primeira diz que o destino está inalcançável e a segunda que o host está sobrecarregado, assim os pacotes devem ser enviados mais lentamente.

Especificando pacotes syn

Pacotes syn são usados para iniciarem uma conexão, o uso da opção `--syn` serve para especificar estes tipos de pacotes. Desta maneira é possível bloquear somente os pacotes que iniciam uma conexão, sem afetar os

pacotes restantes. Para que uma conexão ocorra é necessário que a máquina obtenha a resposta a pacotes syn enviados, caso ele seja bloqueado a resposta nunca será retornada e a conexão não será estabelecida.

```
iptables -A INPUT -p tcp --syn --dport 23 -i ppp+ -j DROP
```

A regra acima bloqueia (-j DROP) qualquer tentativa de conexão (--syn) vindas da interface ppp+ ao telnet (--dport 23) da máquina local, conexões já efetuadas não são afetadas por esta regra. A opção --syn somente pode ser especificada para o protocolo tcp.

ATENÇÃO: - A situação de passagem de pacotes durante deve ser levada em conta durante a inicialização do firewall, bloqueando a passagem de pacotes durante o processo de configuração, criando regras que bloqueiam a passagem de pacotes (exceto para a interface loopback) até que a configuração do firewall esteja completa, pode ser uma solução eficiente.

Outra alternativa segura é configurar as regras de firewall antes das interfaces de rede se tornarem ativas (usando a opção "pre-up comando_firewall" no arquivo de configuração /etc/network/interfaces em sistemas **Debian**).

Especificando fragmentos

A opção "-f" (ou --fragment) permite especificar regras que confirmam com fragmentos. Fragmentos são simplesmente um pacote maior dividido em pedaços para poder ser transmitido via rede TCP/IP para remontagem do pacote pela máquina de destino.

Somente o primeiro fragmento possui detalhes de cabeçalho para ser processado, os segundos e seguintes somente possuem alguns cabeçalhos necessários para dar continuidade ao processo de remontagem do pacote no destino.

Uma regra como

```
iptables -A INPUT -s 200.200.200.1 -f -j DROP
```

derrubará os fragmentos de 200.200.200.1 derrubará o segundo pacote e pacotes seguintes enviados por 200.200.200.1 até nós.

OBS1: Note que se o cabeçalho do pacote não tiver detalhes suficientes para checagem de regras no **iptables**, a regra simplesmente não ira conferir.

OBS2: Não é preciso especificar a opção "-f" para conexões NAT, pois os pacotes são remontados antes de entrarem no código de filtragem.

OBS3: A opção "-f" também pode ser usada para evitar o flood por fragmentos (bomba de fragmentos) que, dependendo da intensidade, podem até travar a máquina.

Especificando uma exceção

Muitos parâmetros (como o endereço de origem/destino, protocolo, porta, mensagens ICMP, fragmentos, etc) podem ser precedidos pelo sinal "!" que significa exceção. Por exemplo:

```
iptables -t filter -A INPUT ! -s 200.200.200.10 -j DROP
```

Diz para rejeitar todos os pacotes EXCETO os que vem do endereço 200.200.200.10.

```
iptables -A INPUT -p tcp ! --syn -s 200.200.200.10 ! -i eth0 -j DROP
```

Diz para bloquear todos os pacotes EXCETO os que iniciam conexões (! --syn), EXCETO para pacotes vindos pela interface eth0 (! -i eth0).

```
iptables -A INPUT -s 200.200.200.10 ! -p tcp -j DROP
```

Bloqueia todos os pacotes vindos de 200.200.200.10, EXCETO os do protocolo tcp.

Especificando um alvo

O alvo (-j) é o destino que um pacote terá quando conferir com as condições de uma regra, um alvo pode dizer para bloquear a passagem do pacote (-j DROP), aceitar a passagem do pacote (-j ACCEPT), registrar o pacote no sistema de log (-j LOG), rejeitar o pacote (-j REJECT), redirecionar um pacote -j REDIRECT, retornar ao chain anterior sem completar o processamento no chain atual (-j RETURN), passar para processamento de programas externos (-j QUEUE), fazer source nat (-j SNAT), destination nat (-j DNAT), etc. Podem existir mais alvos, pois o **iptables** é modularizável, e módulos que acrescentam mais funções podem ser carregados em adição aos já existentes no kernel.

Nos exemplos anteriores vimos o uso de diversos alvos como o DROP e o ACCEPT. Apenas farei uma breve referência sobre os alvos mais usados em operações comuns dos chains. Os alvos REDIRECT, SNAT e DNAT serão explicados em uma seção seguinte:

ACCEPT O pacote é ACEITO e o processamento das regras daquele chains é concluído. Pode ser usado como alvo em todos os chains de todas as tabelas do **iptables** e também pode ser especificado na política padrão das regras do firewall (veja “Especificando a política padrão de um chain - P”).

DROP Rejeita o pacote e o processamento das regras daquele chain é concluído. Pode ser usado como alvo em todos os chains de todas as tabelas do **iptables** e também pode ser especificado na política padrão das regras do firewall (veja “Especificando a política padrão de um chain - P”).

REJECT Este é um módulo opcional que faz a mesma função do alvo *DROP* com a diferença de que uma mensagem ICMP do tipo "icmp-port-unreachable" (TCP/UDP) ou "host-unreachable" (ICMP) é retornada para a máquina de origem. Pode ser usado como alvo somente nos chains da tabela (não como política padrão).

LOG Este módulo envia uma mensagem ao **syslog** caso a regra confira, o processamento continua normalmente para a próxima regra (o pacote não é nem considerado ACEITO ou REJEITADO).

RETURN Retorna o processamento do chain anterior sem processar o resto do chain atual.

QUEUE Passa o processamento para um programa a nível de usuário.

Alvo REJECT

Para ser usado, o módulo `ipt_REJECT` deve ser compilado no kernel ou como módulo. Este alvo rejeita o pacote (como o *DROP*) e envia uma mensagem ICMP do tipo "icmp-port-unreachable" como padrão para a máquina de origem.

É um alvo interessante para bloqueio de portas TCP, pois em alguns casos dá a impressão que a máquina não dispõe de um sistema de firewall (o alvo DROP causa uma parada de muito tempo em alguns

portscanners e tentativas de conexão de serviços, revelando imediatamente o uso de um sistema de firewall pela máquina). O alvo REJECT vem dos tempos do ipchains e somente pode ser usado na tabela *filter*. Quando um pacote confere, ele é rejeitado com a mensagem ICMP do tipo "port unreachable", é possível especificar outro tipo de mensagem ICMP com a opção *--reject-with tipo_icmp*.

OBS: REJECT pode ser usado somente como alvo na tabela filter e não é possível especifica-lo como política padrão do chain filter (como acontecia no **ipchains**. Uma forma alternativa é inserir como última regra uma que pegue todos os pacotes restantes daquele chain e tenha como alvo REJECT (como `iptables -A INPUT -j REJECT`), desta forma ele nunca atingirá a política padrão do chain.

```
# Rejeita pacotes vindos de 200.200.200.1 pela interface ppp0:
iptables -A INPUT -s 200.200.200.1 -i ppp+ -j REJECT
```

Especificando LOG como alvo

Este alvo é usado para registrar a passagem de pacotes no **syslog** do sistema. É um alvo muito interessante para ser usado para regras que bloqueiam determinados tráfegos no sistema (para que o administrador tome conhecimento sobre tais tentativas), para regras de fim de chain (quando você tem um grande conjunto de regras em um firewall restritivo e não sabe onde suas regras estão sendo bloqueadas), para satisfazer sua curiosidade, etc.

```
# Para registrar o bloqueio de pacotes vindos de 200.200.200.1 pela interface ppp0
iptables -A INPUT -s 200.200.200.1 -i ppp+ -j LOG
# Para efetuar o bloqueio
iptables -A INPUT -s 200.200.200.1 -i ppp+ -j REJECT
```

Note que no exemplo anterior a regra que registra o pacote (-j LOG) deve aparecer antes da regra que REJEITA (-j REJECT), caso contrário a regra de LOG nunca funcionará. A regra que REJEITA poderia também ser trocada por uma regra que ACEITA, caso queira registrar um pacote que deve ser aceito (se a política padrão do seu firewall for restritiva (-P DROP). A única coisa que muda nas regras de log é o alvo da regra, isto facilita a implementação de grandes conjuntos de regras de firewall.

A regra acima mostrará a seguinte saída no **syslog** do sistema:

```
Aug 25 10:08:01 debian kernel: IN=ppp0 OUT= MAC=10:20:30:40:50:60:70:80:90:00:00:00
```

Os campos possuem o seguinte significado:

Aug 25 10:08:01	Mês, dia e hora do registro do pacote.
debian	Nome do computador que registrou o pacote.
kernel:	Daemon que registrou a mensagem, no caso o iptables faz parte do próprio kernel.
IN=ppp0	Especifica a interface de entrada (de onde o pacote veio).
OUT=	Especifica a interface de saída (para onde o pacote foi).
MAC=10:20:30:40:50:60:70:80:90:00:00:00	Endereço mac da interface de rede (pode ser obtido com <code>arp -n</code> interface).
SRC=200.200.200.1	Endereço de origem do pacote.

DST=200.210.10.10	Endereço de destino do pacote.
SEQ=234234343	Número de seqüência da recepção. É ativado com a opção <i>--log-tcp-sequence</i> .
LEN=61	Tamanho em bytes do pacote IP.
TOS=0x00	Prioridade do cabeçalho TOS (Tipo). Veja a seção “Especificando o tipo de serviço” para mais detalhes.
PREC=0x00	Prioridade do cabeçalho TOS (Precedência). Veja a seção “Especificando o tipo de serviço” para mais detalhes.
TTL=64	Tempo de vida do pacote. No exemplo, 64 roteadores (hops).
ID=0	Identificação única destes datagrama. Esta identificação também é usada pelos fragmentos seguintes deste pacote.
DF	Opção "Don't fragment" (não fragmentar) do pacote. Usada quando o pacote é pequeno o bastante para não precisar ser fragmentado.
MF	Opção "More Fragments" (mais fragmentos) estão para ser recebidos.
FRAG=100	Tamanho do fragmento especificado em pacotes de 8 bits. No exemplo acima, o pacote tem o tamanho de 800 bytes (100*8).
PROTO=UDP	Nome do protocolo. Pode ser TCP, UDP ou ICMP
SPT=1031	Porta de origem da requisição.
DPT=53	Porta de destino da requisição.
LEN=41	Tamanho do pacote.

O log acima mostra uma consulta DNS (porta destino 53) para nossa máquina (INPUT) de 200.200.200.1 para 200.210.10.10.

O problema é que em um grande número de regras será difícil saber qual regra conferiu (pois teríamos que analisar o endereço/porta origem/destino) e o destino do pacote (se ele foi ACEITO ou BLOQUEADO) pois você pode ter regras para ambas as situações. Por este motivo existem algumas opções úteis que podemos usar com o alvo LOG:

<i>--log-prefix</i> "descrição"	Permite especificar uma descrição para a regra do firewall de até 29 caracteres. Caso tiver espaços, devem ser usadas "aspas".
<i>--log-level</i> nível	Especifica o nível da mensagem no syslog . Veja os níveis na seção “Arquivo de configuração <code>syslog.conf</code> ” do guia.
<i>--log-tcp-options</i>	Registra campos do cabeçalho TCP nos logs do sistema.
<i>--log-ip-options</i>	Registra campos do cabeçalho IP nos logs do sistema
<i>--log-tcp-sequence</i>	Registra os números de seqüencia TCP. Evite ao máximo o uso desta opção, pois a seqüencia de números TCP pode ser a chave para um seqüestro de seção ou IP spoofing em seu sistema caso algum usuário tenha acesso a estes logs. Caso utilize tcp/ip em servidores públicos, o uso desta opção ajudará a entender bem os

ataques DoS causados por syn-flood e porque ativar os SynCookies (veja “Proteção contra syn flood”).

OBS1:Lembre-se que estas opções são referentes ao alvo LOG e devem ser usadas após este, caso contrário você terá um pouco de trabalho para analisar e consertar erros em suas regras do firewall.

OBS2:Caso esteja usando o firewall em um servidor público, recomendo associar um limite a regra de log, pois um ataque poderia causar um DoS enchendo sua partição. Leia mais sobre isso em “Limitando o número de vezes que a regra confere”.

```
# Complementando o exemplo anterior:
# Para registrar o bloqueio de pacotes vindos de 200.200.200.1 pela interface ppp0
iptables -A INPUT -s 200.200.200.1 -i ppp+ -j LOG --log-prefix "FIREWALL: Derrubado"
# Para efetuar o bloqueio
iptables -A INPUT -s 200.200.200.1 -i ppp+ -j REJECT
```

Retornará a seguinte mensagem no syslog:

```
Aug 25 10:08:01 debian kernel: FIREWALL: Derrubado IN=ppp0 OUT= MAC=10:20:30:40:50
```

Agora você sabe o que aconteceu com o pacote (Rejeitado). A padronização de mensagens de firewall é também importante para a criação de scripts de análise que poderão fazer a análise dos logs do seu firewall (para criação de estatísticas que podem servir como base para a criação de novas regras de firewall ou eliminação de outras).

OBS: Se você sente falta da função "-l" do **ipchains** que combina o alvo e log na mesma regra você pode criar um alvo como o seguinte:

```
iptables -N log-drop
iptables -A log-drop -j LOG
iptables -A log-drop -j DROP
```

E usar "log-drop" como alvo em suas regras. Mesmo assim esta solução é "limitada" em relação a "-l" do **ipchains** porque o **iptables** não inclui detalhes de qual chain bloqueou o pacote/qual pacote foi bloqueado, assim é necessário a especificação da opção *--log-prefix* para as mensagens se tornarem mais compreensíveis. Esta limitação pode ser contornada utilizando um firewall feito em linguagem shell script, desta forma você terá um controle maior sobre o seu programa usando funções e integração com outros utilitários.

Especificando RETURN como alvo

O alvo RETURN diz ao **iptables** interromper o processamento no chain atual e retornar o processamento ao chain anterior. Ele é útil quando criamos um chain que faz um determinado tratamento de pacotes, por exemplo bloquear conexões vindas da internet para portas baixas, exceto para um endereço IP específico. Como segue:

```
1-) iptables -t filter -A INPUT -i ppp0 -j internet
2-) iptables -t filter -j ACCEPT
3-) iptables -t filter -N internet
4-) iptables -t filter -A internet -s www.debian.org -p tcp --dport 80 -j RETURN
5-) iptables -t filter -A internet -p tcp --dport 21 -j DROP
6-) iptables -t filter -A internet -p tcp --dport 23 -j DROP
```

```
7-) iptables -t filter -A internet -p tcp --dport 25 -j DROP
8-) iptables -t filter -A internet -p tcp --dport 80 -j DROP
```

Quando um pacote com o endereço `www.debian.org` tentando acessar a porta `www` (80) de nossa máquina através da internet (via interface `ppp0`), o chain número 1 confere, então o processamento continua no chain número 4, o chain número 4 confere então o processamento volta para a regra número 2, que diz para aceitar o pacote.

Agora se um pacote vem com o endereço `www.dominio.com.br` tentando acessar a porta `www` *80) de nossa máquina através da internet (via interface `ppp0`), o chain número 1 confere, então o processamento continua no chain número 4, que não confere. O mesmo acontece com os chains 5, 6 e 7. O chain número 8 confere, então o acesso é bloqueado.

Como pode notou, o alvo RETURN facilita bastante a construção das regras do seu firewall, caso existam máquinas/redes que sejam exceções as suas regras. Se ela não existisse, seria necessário especificar diversas opções `-s`, `-d`, etc para poder garantir o acesso livre a determinadas máquinas.

Salvando e Restaurando regras

As regras que você está trabalhosamente criando e testando manualmente enquanto manipula o **iptables** podem ser salvas de 2 formas; uma delas é escrevendo um shell script que tenha todos os comandos, um por linha. Isto é recomendado quando tem um firewall grande e que exige uma boa padronização de regras, bem como sua leitura, comentários. O script shell também permite o uso de funções presente no interpretador de comando, portanto se você é uma pessoa que gosta de interagir com as funções do shell e deixar as coisas mais flexíveis, prefira esta opção.

A outra forma é usando as ferramentas **iptables-save** e **iptables-restore** baseada na idéia do **ipchains-save** e **ipchains-restore**. O **iptables-save** deve ser usado sempre que modificar regras no firewall iptables da seguinte forma:

```
iptables-save >/dir/iptables-regras
```

Uma das vantagens do uso do **iptables-save** é ele também salvar os contadores de chains, ou seja, a quantidade de pacotes que conferiram com a regra. Isto também pode ser feito com algumas regras adicionais em seu shell script, caso tenha interesse nesses contadores para estatísticas ou outros tipos de relatórios.

Para restaurar as regras salvas, utilize o comando:

```
iptables-restore </dir/iptables-regras
```

A tabela nat (Network Address Translation) - fazendo nat

A tabela *nat* serve para controlar a tradução dos endereços que atravessam o código de roteamento da máquina **Linux**. Existem 3 chains na tabela *nat*: *PREROUTING*, *OUTPUT* e *POSTROUTING* (veja “O que são tabelas?” para maiores detalhes).

A tradução de endereços tem inúmeras utilidades, uma delas é o Masquerading, onde máquinas de uma rede interna podem acessar a Internet através de uma máquina **Linux**, redirecionamento de porta, proxy transparente, etc. Esta seção abordará os tipos de NAT, exemplos de como criar rapidamente uma conexão IP masquerading e entender como a tradução de endereços funciona no **iptables**.

Se sua intenção é ligar sua rede a Internet existem duas opções:

- Você possui uma conexão que lhe oferece um endereço IP dinâmico (a cada conexão é dado um endereço IP - como uma conexão PPP) então o IP masquerading é o que precisa (veja “Fazendo IP masquerading (para os apressados)” ou “Fazendo IP Masquerading”).
- Você tem uma conexão que lhe oferece um endereço IP permanente (ADSL, por exemplo) então o SNAT é o que precisa (veja “Fazendo SNAT”).

Criando um novo chain na tabela NAT

O procedimento para criação de um novo chain nesta tabela é o mesmo descrito em “Criando um novo chain - N” será necessário somente especificar a tabela nat (-t nat) para que o novo chain não seja criado na tabela padrão (-t filter).

```
iptables -t nat -N intra-inter
```

Que criará o chain chamado *intra-inter* na tabela *nat*. Para inserir regras neste chain será necessário especificar a opção "-t nat".

Fazendo IP masquerading (para os apressados)

Você precisará de um kernel com suporte ao **iptables** (veja “Habilitando o suporte ao iptables no kernel” e **ip_forward** e então digitar os dois comandos abaixo para habilitar o masquerading para todas as máquinas da rede 192.168.1.*:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j MASQUERADE  
echo "1" >/proc/sys/net/ipv4/ip_forward
```

A configuração do servidor **Linux** está completa, agora os clientes da rede precisarão ser configurados para usar o endereço IP do servidor **Linux** como gateway. É recomendável instalar um servidor proxy e DNS na máquina **Linux** para acelerar o desempenho das requisições/resolução de nomes das máquinas em rede. A utilização de bits TOS também pode trazer um grande aumento de velocidade para os diferentes serviços da rede (veja “Especificando o tipo de serviço”).

Fazendo SNAT

SNAT (source nat - nat no endereço de origem) consiste em modificar o endereço de origem das máquinas clientes antes dos pacotes serem enviados. A máquina roteadora é bastante inteligente o bastante para lembrar dos pacotes modificados e reescrever os endereços assim que obter a resposta da máquina de destino, direcionando os pacotes ao destino correto. Toda operação de SNAT é feita no chain *POSTROUTING*.

É permitido especificar endereços de origem/destino, protocolos, portas de origem/destino, interface de entrada/saída (dependendo do chain), alvos, etc. É desnecessário especificar fragmentos na tabela nat, pois eles serão remontados antes de entrar no código de roteamento.

O SNAT é a solução quando você tem acesso a internet através de um único IP e deseja fazer que sua rede tenha acesso a Internet através da máquina **Linux**. Nenhuma máquina da Internet poderá ter acesso direto as máquinas de sua rede interna via SNAT.

OBS: A observação acima não leva em conta o controle de acesso externo configurado na máquina que estiver configurando o **iptables**, uma configuração mau realizada pode expor sua máquina a acessos externos indesejados e comprometer sua rede interna caso alguém consiga acesso direto ao servidor.

É necessário especificar SNAT como alvo (-j SNAT) quando desejar que as máquinas de sua rede interna tenha acesso a Internet através do IP fixo da máquina **Linux** (para conexões intermitentes como PPP, veja “Fazendo IP Masquerading”). O parâmetro `--to IP:portas` deve ser usado após o alvo SNAT. Ele serve para especificar um endereço IP, faixa de endereços e opcionalmente uma porta ou faixa de portas que será substituída. Toda a operação de SNAT é realizada através do chain POSTROUTING:

```
# Modifica o endereço IP dos pacotes vindos da máquina 192.168.1.2 da rede interna
# que tem como destino a interface eth1 para 200.200.217.40 (que é o nosso endereço
# IP da interface ligada a Internet).
iptables -t nat -A POSTROUTING -s 192.168.1.2 -o eth1 -j SNAT --to 200.200.217.40
```

Os pacotes indo para a Internet (nossa conexão é feita via eth1, nossa interface externa) vindo do endereço 192.168.1.2, são substituídos por 200.241.200.40 e enviados para fora. Quando a resposta a requisição é retornada, a máquina com **iptables** recebe os pacotes e faz a operação inversa, modificando o endereço 200.241.200.40 novamente para 192.168.1.2 e enviando a resposta a máquina de nossa rede interna. Após definir suas regras de NAT, execute o comando `echo "1" >/proc/sys/net/ipv4/ip_forward` para habilitar o suporte a redirecionamento de pacotes no kernel.

Também é possível especificar faixas de endereços e portas que serão substituídas:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j SNAT --to 200.200.217.
```

Modifica o endereço IP de origem de todas as máquinas da rede 192.168.1.0/24 que tem o destino a interface eth0 para 200.241.200.40 a 200.241.200.50. O endereço IP selecionado é escolhido de acordo com o último IP alocado.

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j SNAT --to 200.200.217.
```

Idêntico ao anterior, mas faz somente substituições na faixa de portas de origem de 1 a 1023.

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j SNAT --to 200.200.217.
```

Faz o mapeamento para a faixa de portas 200.200.217.40 a 200.200.217.50 e de 200.200.217.70 a 200.200.217.73.

OBS1: Se por algum motivo não for possível mapear uma conexão NAT, ela será derrubada.

OBS2: Tenha certeza que as respostas podem chegar até a máquina que fez o NAT. Se estiver fazendo SNAT em um endereço livre em sua rede (como 200.200.217.73).

OBS3: Como notou acima, o SNAT é usado quando temos uma conexão externa com um ou mais IP's fixos. O Masquerading é uma forma especial de SNAT usada para funcionar em conexões que recebem endereços IP aleatórios (PPP).

OBS4: Não se esqueça de habilitar o redirecionamento de pacotes após fazer suas regra de NAT com o comando: `echo "1" >/proc/sys/net/ipv4/ip_forward`, caso contrário o redirecionamento de pacotes não funcionará.

Fazendo IP Masquerading

O IP Masquerading é um tipo especial de SNAT usado para conectar a sua rede interna a internet quando você recebe um IP dinâmico de seu provedor (como em conexões ppp). Todas as operações de

IP Masquerading são realizadas no chain *POSTROUTING*. Se você tem um IP fixo, deve ler “Fazendo SNAT”.

Para fazer IP Masquerading de uma máquina com o IP 192.168.1.2 para ter acesso a Internet, use o comando:

```
iptables -t nat -A POSTROUTING -s 192.168.1.2/32 -o ppp0 -j MASQUERADE
```

A diferença é que o alvo é *-j MASQUERADE*. O comando acima faz IP Masquerading de todo o tráfego de 192.168.1.2 indo para a interface ppp0: O endereço IP dos pacotes vindos de 192.168.1.2 são substituídos pelo IP oferecido pelo seu provedor de acesso no momento da conexão, quando a resposta é retornada a operação inversa é realizada para garantir que a resposta chegue ao destino. Nenhuma máquina da internet poderá ter acesso direto a sua máquina conectava via Masquerading.

Para fazer o IP Masquerading de todas as máquinas da rede 192.168.1.*:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp0 -j MASQUERADE
```

Após definir a regra para fazer Masquerading (SNAT), execute o comando `echo "1" >/proc/sys/net/ipv4/ip_forward` para habilitar o suporte a redirecionamento de pacotes no kernel.

Fazendo DNAT

DNAT (Destination nat - nat no endereço de destino) consiste em modificar o endereço de destino das máquinas clientes. O destination nat é muito usado para fazer redirecionamento de pacotes, proxies transparentes e balanceamento de carga.

Toda operação de DNAT é feita no chain *PREROUTING*. As demais opções e observações do SNAT são também válidas para DNAT (com exceção que somente é permitido especificar a interface de origem no chain *PREROUTING*).

```
# Modifica o endereço IP destino dos pacotes de 200.200.217.40 vindo da interface
# para 192.168.1.2.
iptables -t nat -A PREROUTING -s 200.200.217.40 -i eth0 -j DNAT --to 192.168.1.2
```

Também é possível especificar faixas de endereços e portas que serão substituídas no DNAT:

```
iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/24 -j DNAT --to 200.200.217.4
```

Modifica o endereço IP de destino do tráfego vindos da interface 192.168.1.0/24 para um IP de 200.241.200.40 a 200.241.200.50. Este é um excelente método para fazer o balanceamento de carga entre servidores. O endereço IP selecionado é escolhido de acordo com o último IP alocado.

```
iptables -t nat -A PREROUTING -i eth0 -s 192.168.1.0/24 -j DNAT --to 200.200.217.4
```

Idêntico ao anterior, mas faz somente substituições na faixa de portas de destino de 1024 a 5000. A operação acima é a mesma realizada pelo **ipmasqadm** dos kernels da série 2.2.

OBS1: Se por algum motivo não for possível mapear uma conexão NAT, ela será derrubada.

OBS2: Não se esqueça de conferir se o `ip_forward` está ajustado para 1: `echo "1" >/proc/sys/net/ipv4/ip_forward`.

Redirecionamento de portas

O redirecionamento de portas permite a você repassar conexões com destino a uma porta para outra porta na mesma máquina. O alvo *REDIRECT* é usado para fazer esta operação, junto com o argumento *--to-port* especificando a porta que será redirecionada. Este é o método DNAT específico para se fazer proxy transparente (para redirecionamento de endereços/portas, veja “Fazendo DNAT”). Todas as operações de redirecionamento de portas é realizada no chain *PREROUTING* e *OUTPUT* da tabela *nat*.

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 81
```

Redireciona as conexões indo para a porta 80 para a porta 81 (rodando **squid**) no firewall.

ATENÇÃO: O **squid** possui suporte a proxy transparente, e poderá atender as requisições acima da regra acima.

Monitorando conexões feitas na tabela nat

Use o comando `cat /proc/net/ip_conntrack` para listar todas as conexões atuais tratadas pelo módulo *nat*.

A tabela mangle

A tabela *mangle* serve para especificar ações especiais para o tratamento do tráfego que atravessa os chains. Nesta tabela existem cinco chains: *PREROUTING*, *POSTROUTING*, *INPUT*, *OUTPUT* e *FORWARD* (veja “O que são tabelas?” para maiores detalhes).

Em geral, cada um deste chain é processado antes do chain correspondente na tabela *filter* e *nat* para definir opções especiais para o tráfego (por exemplo, o chain *PREROUTING* da tabela *mangle* é processado antes do *PREROUTING* da tabela *nat*). O chain *OUTPUT* da tabela *mangle* corresponde ao *OUTPUT* da tabela *nat*. Opções como o *Tipo de Serviço (TOS)* é especificado nesta tabela para classificar e aumentar consideravelmente a velocidade de tráfego considerados em tempo real. Mesmo após o tráfego ser estabelecido, os chains da tabela *mangle* continuam ativos para garantir que as opções especiais relacionadas com a conexão continuem fazendo efeito (veja os exemplos de “Caminho percorrido pelos pacotes nas tabelas e chains”).

Especificando o tipo de serviço

O tipo de serviço é um campo existente no cabeçalho de pacotes do protocolo ipv4 que tem a função especificar qual é a prioridade daquele pacote. A prioridade é definida usando o algoritmo FIFO do próprio kernel, sendo uma das alternativas de controle/priorização de tráfego das mais simples e rápidas.

Uma das vantagens da utilização do tipo de serviço é dar prioridade ao tráfego de pacotes interativos (como os do ICQ, IRC, servidores de chat), etc. Com o TOS especificado, mesmo que esteja fazendo um download consumindo toda a banda de sua interface de rede, o tráfego com prioridade interativa será enviado antes, aumentando a eficiência do uso de serviços em sua máquina.

Em testes realizados em minha conexão de 56K, o uso de regras TOS aumentou bastante o desempenho em tráfego interativo (em torno de 300%), durante o uso total da banda da interface ppp em grande consumo de banda.

Usamos o alvo TOS (-j TOS) para especificar a modificação do tipo de serviço nos pacotes que atravessam as regras do firewall, acompanhada do argumento *--set-tos TOS* que define a nova prioridade dos pacotes. Os valores aceitos são os seguintes:

Espera Mínima	É especificado através de <i>Minimize-Delay</i> , 16 ou 0x10
Máximo Processamento	É especificado através de <i>Maximize-Throughput</i> , 8, ou 0x08.
Máxima Confiança	É especificado através de <i>Maximize-Reliability</i> , 4 ou 0x04.
Custo mínimo	Especificado através de <i>Minimize-Cost</i> , 2 ou 0x02.
Prioridade Normal	Especificado através de <i>Normal-Service</i> , 0 ou 0x00.

Os pacotes vem por padrão com o valor TOS ajustado como *prioridade normal* (bits tos ajustados para 0x00). O tipo *Mínima Espera* é o de maior prioridade, recomendado para tráfego interativo.

Especificando o TOS para tráfego de saída

Este é o mais usado, pois prioriza o tráfego que sai da máquina (com destino a Internet, por exemplo). Sua operação é realizada através do chain *OUTPUT* ou *POSTROUTING*.

Para priorizar todo o tráfego de IRC de nossa rede interna indo para a interface ppp0:

```
iptables -t mangle -A OUTPUT -o ppp0 -p tcp --dport 6666-6668 -j TOS --set-tos 16
```

O bit TOS é ajustado para *Espera mínima* e será enviado antes dos pacotes com prioridade normal para fora. Para priorizar a transmissão de dados ftp saindo da rede:

```
iptables -t mangle -A OUTPUT -o ppp0 -p tcp --dport 20 -j TOS --set-tos 8
```

Para priorizar o tráfego de ICQ da rede:

```
iptables -t mangle -A OUTPUT -o ppp0 -p tcp --dport 5190 -j TOS --set-tos 16
```

Existem muitas outras otimizações que podem ser feitas, só depende dos requerimentos e análise de cada serviço da rede pelo administrador.

OBS: - Os pacotes que atravessam o alvo TOS somente tem os bits tipo do serviço modificados, eles não serão de qualquer forma rejeitados.

Outros módulos do iptables

Os módulos do iptables são especificados com a opção *-m módulo* ou *--match módulo* e permitem expandir a funcionalidade do firewall através de novas conferências e recursos de filtragem adicionais, como limitar a conferência de regras do firewall (um método útil de limitar ping floods, syn floods, etc).

Conferindo de acordo com o estado da conexão

Este módulo permite especificar regras de acordo com o estado da conexão do pacote, isto é feito através da interpretação da saída do módulo *ip_conntrack*. O parâmetro *--state OPÇÕES* deve acompanhar este módulo. As opções permitidas são as seguintes:

- NEW - Confere com pacotes que criam novas conexões
- ESTABLISHED - Confere com conexões já estabelecidas

- RELATED - Confere com pacotes relacionados indiretamente a uma conexão, como mensagens de erro icmp, etc.
- INVALID - Confere com pacotes que não puderam ser identificados por algum motivo. Como respostas de conexões desconhecidas.

Caso seja necessário especificar mais de uma opções estas devem ser separadas por vírgulas.

```
iptables -A INPUT -m state --state NEW -i ppp0 -j DROP
```

Bloqueia qualquer tentativa de nova conexão vindo da interface ppp0.

```
iptables -A INPUT -m state --state NEW,INVALID -i ppp0 -j LOG
```

Permite registrar novas conexões e pacotes inválidos vindos da interface ppp0.

Limitando o número de vezes que a regra confere

A opção *-m limit* permite especificar o número de vezes que uma regra conferirá quando todas as outras condições forem satisfeitas. O número padrão de conferência é de 3 por hora, a não ser que seja modificado através dos argumentos aceitos pelo *limit*:

- `--limit num/tempo` - Permite especificar a taxa de conferências do limit. O parâmetro *num* especifica um número e *tempo* pode ser
 - s - Segundo
 - m - Minuto
 - h - Hora
 - d - Dia

Assim uma regra como `iptables -A INPUT -m limit --limit 5/m -j ACCEPT` permitirá que a regra acima confira apenas 5 vezes por minuto (`--limit 2/s`). Este limite pode ser facilmente adaptado para uma regra de log que confere constantemente não causar uma avalanche em seus logs. O valor padrão é 3/h.

- `--limit-burst num` - Especifica o número inicial máximo de pacotes que irão conferir, este número é aumentado por 1 a cada vez que o parâmetro `--limit` acima não for atingido. O valor padrão é 5.

Proteção contra ping da morte

A regra abaixo pode tomada como base para proteção contra ping flood:

```
iptables -t filter -A ping-chain -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
iptables -t filter -A ping-chain -j DROP
```

A regra acima limita em 1 vez por segundo (`--limit 1/s`) a passagem de pings (echo requests) para a máquina **Linux**.

```
iptables -t filter -A ping-chain -i ppp0 -p icmp --icmp-type echo-reply -m limit --limit 1/s -j ACCEPT
```

```
iptables -t filter -A ping-chain -j DROP
```

Limita respostas a pings (echo reply) vindos da interface ppp0 (-i ppp0) a 1 por segundo.

ATENÇÃO: O exemplo acima é somente para a criação de suas próprias regras com limitações, caso um pacote não confira ele será bloqueado pela próxima regra. Se uma regra como esta for colocada no chain INPUT sem modificações, ela não terá o efeito desejado, podendo colocar em risco a sua instalação pela falsa impressão de segurança. Portanto, é recomendável sempre testar as modificações para ter certeza que elas tem efeito.

Proteção contra syn flood

A regra abaixo é uma boa proteção para os ataques syn floods:

```
iptables -t filter -A syn-chain -p tcp --syn -m limit --limit 2/s -j ACCEPT
iptables -t filter -A syn-chain -j DROP
```

Esta regra limita o atendimento de requisições de conexões a 2 por segundo. Outra forma de aumentar a segurança contra syn-floods é através do próprio kernel ativando a opção "TCP Synflood" na compilação e depois executando: `echo "1" >/proc/sys/net/ipv4/tcp_synflood`. No entanto, utilize estas opções com cautela em servidores que possuem um grande número de acessos para não ter problemas que afetem seu clientes.

ATENÇÃO: Os exemplos acima devem são somente exemplos para criação de suas próprias regras com limitações, caso um pacote não confira com a regra ele será bloqueado pela próxima regra. Se uma regra como esta for colocada no chain INPUT sem modificações, ela não terá o efeito desejado, podendo colocar em risco a sua instalação pela falsa impressão de segurança. Portanto, é recomendável sempre testar as modificações para ter certeza que elas tem efeito.

Proteção contra IP spoofing

A especificação de endereços de origem/destino junto com a interface de rede pode ser usado como um detector de ataques spoofing. A lógica é que todos os endereços que NUNCA devem vir da interface X devem ser negados imediatamente. As regras abaixo são colocadas no início do chain INPUT para detectar tais ataques:

```
iptables -A INPUT -s 192.168.1.0/24 -i ! eth0 -j DROP
iptables -A INPUT ! -s 192.168.1.0/24 -i eth0 -j DROP
```

A primeira regra diz para bloquear todos os endereços da faixa de rede 192.168.1.* que NÃO vem da interface eth0, a segunda regra diz para bloquear todos os endereços que não sejam 192.168.1.* vindos da interface eth0. O símbolo "!" serve para especificar exceções (veja "Especificando uma exceção". O kernel do **Linux** automaticamente bloqueia a passagem de pacotes que dizem ser de 127.0.0.1 e não está vindo da interface loopback.

O método preferido para controlar o ip spoofing é através do código de roteamento do kernel (a não ser que esteja usando algum tipo de roteamento de origem assimétrico necessário por alguns programas especiais):

```
for i in /proc/sys/net/ipv4/conf/*/rp_filter; do
  echo 1 >$i
done
```

Desta forma qualquer endereço dizendo ser 192.168.1.5 vindo de ppp0 será imediatamente rejeitado. Uma checagem adicional contra IP spoofing pode ser feita no arquivo `/etc/host.conf` (veja “`/etc/host.conf`”).

Especificando múltiplas portas de origem/destino

O módulo `multiport` permite que seja especificado múltiplas portas para um alvo. Podem ser especificadas até 15 portas em um único parâmetro e basta que uma porta confira para que a regra entre em ação, pois a comparação é feita usando condições "or". O parâmetro `multiport` deve ser acompanhado de um dos argumentos abaixo:

- `--source-port [porta1, porta2...]` - Faz a regra conferir se a porta de origem estiver presente entre as portas especificadas.
- `--destination-port [porta1, porta2...]` - Faz a regra conferir se a porta de destino estiver presente entre as portas especificadas.
- `--port [porta1, porta2...]` - Faz a regra conferir caso a porta de origem ou destino esteja presente no parâmetro.

Este módulo pode eliminar muitas regras de firewall que fazem o mesmo tratamento de pacotes para diversas portas diferentes.

```
iptables -A INPUT -p tcp -i ppp0 -m multiport --destination-port 21,23,25,80,110,1
```

Bloqueia todos os pacotes vindo de ppp0 para as portas 21 (ftp), 23 (telnet), 25 (smtp), 80 (www), 110 (pop3), 113 (ident), 6667 (irc).

Especificando o endereço MAC da interface

O módulo `mac` serve para conferir com o endereço Ethernet dos pacotes de origem. Somente faz sentido se usado nos chains de PREROUTING (da tabela nat) ou INPUT (da tabela filter). Aceita como argumento a opção `--mac-source endereço`. O símbolo "!" pode ser usado para especificar uma exceção.

```
iptables -t filter -A INPUT -m mac --mac-source 00:80:AD:B2:60:0B -j DROP
```

Confere com a máquina com endereço ethernet igual a 00:80:AD:B2:60:0B.

Conferindo com quem criou o pacote

Este módulo confere com o usuário que iniciou a conexão. É somente válido no chain *OUTPUT* da tabela filter. Os seguintes argumentos são válidas para este módulo:

- `--uid-owner UID` - Confere se o pacote foi criado por um processo com o UID especificado. Até o momento somente UID numéricos são aceitos.
- `--gid-owner GID` - Confere se o pacote foi criado por um usuário pertencente ao grupo GID. Até o momento somente GID numéricos são aceitos.
- `--pid-owner PID` - Confere se o pacote foi criado por um processo com o PID especificado.
- `--sid-owner ID` - Confere se o pacote foi criado por um processo no grupo de seção especificado.

OBS: - Lembre-se que pacotes que não possuem detalhes suficientes de cabeçalho nunca conferirão!

```
iptables -A OUTPUT -m owner --gid-owner 100 -p udp -j DROP
```

Rejeita um conexões indo para portas UDP de pacotes criados pelo usuários pertencentes ao grupo 100.

Conferindo com o conteúdo do pacote

O módulo `string` do **iptables** permite a inspeção de conteúdo de um pacote e tomar uma ação se determinado tipo de tráfego for encontrado em um pacote. Esta técnica pode ser usada tanto para segurança como para economia de banda dentro da rede. Esta opção **NÃO** torna o iptables como um firewall proxy, pois o proxy tem a habilidade de inspecionar o conteúdo, protocolo, comandos do pacote e decidir se o seu conteúdo é nocivo ou não. O firewall em nível de pacotes fazendo inspeção de conteúdo, chega a ser 3 a 10 vezes mais rápido do que um proxy, assim seu uso deve ser analisado dependendo do tráfego que circula pelo link e da segurança dos dados que trafegam através dele.

Uma boa prática é aliar esta opção a um IDS externo usando o alvo *QUEUE* e deixando o trabalho de espeção de conteúdo para ele. Um exemplo de restrição direta é o bloqueio do envio de qualquer informação confidencial sigilosa para fora da rede interna (número de contas, tudo que conferir com CPF, CGC, endereços de e-mail, memorandos, etc). De qualquer forma, analise o tráfego de sua rede antes de querer implementar qualquer solução baseada neste método sob o risco de afetar tráfego legítimo.

Outra utilidade eficiente é a diminuição de tráfego, pois podemos barrar programas que sobrecarregam o link em uma rede com muitos usuários como, por exemplo, usando o **Kazaa** ou qualquer outro programa para cópia de arquivos via Internet. Veja alguns exemplos:

```
# Bloqueia qualquer tentativa de acesso ao programa Kazaa
iptables -A INPUT -m string --string "X-Kazaa" -j DROP
```

```
# Não permite que dados confidenciais sejam enviados para fora da empresa
# e registra o ocorrido.
```

```
iptables -A OUTPUT -m string --string "conta" -j LOG --log-prefix "ALERTA: dados c
iptables -A OUTPUT -m string --string "conta" -j DROP
```

```
# Somente permite a passagem de pacotes que não contém ".exe" em seu conteúdo
iptables -A INPUT -m string --string ! ".exe" -j ACCEPT
```

Conferindo com o tempo de vida do pacote

O módulo `tTL` pode ser usado junto com as seguintes opções para conferir com o tempo de vida (TTL) de um pacote:

- `--ttl-eq [num]`
- `--ttl-lt [num]`
- `--ttl-gq [num]`

Veja alguns exemplos:

```
# Confere com todos os pacotes que tem o TTL maior que 100
iptables -A INPUT -m ttl --ttl-gt 100 -j LOG --log-prefix "TTL alto"
```

```
# Confere com todos os pacotes que tem o TTL igual a 1
```

```
iptables -A INPUT -m ttl --ttl-eq 1 -j DROP
```

OBS: Tenha um especial cuidado durante a programação de regras que usem TTL, como elas estão especialmente associadas com o estado da comunicação estabelecida entre as duas pontas e o tipo de protocolo, cuidados especiais devem ser tomados para que seu firewall não manipule de forma incorreta tráfego válido.

Conferindo com números RPC

O módulo `rpc` permite um controle especial sobre o tráfego RPC que chega até a sua máquina. Um uso útil é restringir a chamada a determinados números RPC e permitir outros (por exemplo, permitindo somente o serviço `key serv` e bloqueando outros como o `ypserv` ou `portmapper`). As seguintes opções podem ser usadas com o módulo `nfs`:

- `--rpcs [procedimentos]` - Confere com a lista de chamadas RPC especificadas. Mais de um procedimento RPC pode ser especificado como nome ou número separando-os com vírgulas. Um arquivo útil que contém esta lista é o `/etc/rpc`.
- `--strict` - Ignora serviços RPC que não contenham a chamada `get` do `portmapper`. Em situações normais, o início de qualquer solicitação RPC.

Veja alguns exemplos:

```
# Para conferir com todas as chamadas RPC referentes a conexões iniciadas
# para o portmapper
iptables -A INPUT -m rpc --rpcs portmapper --strict -j DROP

# Para permitir que somente as chamadas para status e statmon sejam
# aceitas
iptables -A INPUT -m rpc --rpcs 100023,100024 -j ACCEPT
```

Conferindo com tipo de pacote

O módulo `pkttype` permite identificar um pacote do tipo *unicast* (direcionado a você), *broadcast* (direcionado a uma determinada rede, definida pela netmask) ou *multicast* (destinado a grupos de redes) e desta forma realizar ações em cima destes. O tipo de pacote é identificado logo após a opção `--pkt-type`. Veja alguns exemplos:

```
# Bloqueia a passagem de pacotes multicast de uma rede para outra
iptables -A FORWARD -i eth0 -o eth0 -m pkttype --pkt-type multicast -j DROP

# Como deve ter notado, é possível fazer a associação com diversas especificações
# de módulos, bastando apenas especificar uma opção "-m" para cada módulo
# adicional:
# Permite a passagem de pacotes broadcast de uma rede para outra com
# limitação de 5/s.
iptables -A FORWARD -i eth0 -o eth0 -m pkttype --pkt-type broadcast -m limit --lim
```

Conferindo com o tamanho do pacote

O tamanho do pacote pode ser usado como condição de filtragem através do módulo `length`. O tamanho do pacote é especificado através da opção `--length` e o argumento segue a mesma sintaxe da especificação de portas no **iptables** sendo separados por `:`. Veja alguns exemplos:

```
# Bloqueia qualquer pacote ICMP maior que 30Kb
iptables -A INPUT -i eth0 -m length --length 30000: -j DROP

# Bloqueia qualquer pacote com o tamanho entre 20 e 2000 bytes
iptables -A INPUT -i eth0 -m length --length 20:2000 -j DROP
```

Caminho percorrido pelos pacotes nas tabelas e chains

É MUITO importante entender a função de cada filtro e a ordem de acesso dos chains de acordo com o tipo de conexão e interface de origem/destino. Esta seção explica a ordem que as regras são atravessadas, isso lhe permitirá planejar a distribuição das regras nos chains, e evitar erros de localização de regras que poderia deixar seu firewall com sérios problemas de segurança, ou um sistema de firewall totalmente confuso e sem lógica.

Nos exemplos abaixo assumirei a seguinte configuração:

- A máquina do firewall com **iptables** possui o endereço IP 192.168.1.1 e conecta a rede interna ligada via interface eth0 a internet via a interface ppp0.
- Rede interna com a faixa de endereços 192.168.1.0 conectada ao firewall via interface eth0
- Interface ppp0 fazendo conexão com a Internet com o endereço IP 200.217.29.67.
- A conexão das máquinas da rede interna (eth0) com a rede externa (ppp0) é feita via *Masquerading*.

Também utilizarei a sintaxe *CHAIN-tabela* para fazer referência aos chains e tabelas dos blocos ASCII: *INPUT-filter* - chain *INPUT* da tabela *filter*.

ATENÇÃO: A ordem de processamento das regras do **iptables**, é diferente do **ipchains** devido a inclusão do novo sistema de nat e da tabela mangle.

Ping de 192.168.1.1 para 192.168.1.1

- Endereço de Origem: 192.168.1.1
- Endereço de Destino: 192.168.1.1
- Interface de Entrada: lo
- Interface de Saída: lo
- Protocolo: ICMP
- Descrição: Ping para o próprio firewall

SAÍDA DE PACOTES (envio do ping para 192.168.1.1):

```
+-----+ +-----+ +-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-nat| => |OUTPUT-filter| => |POSTROUTING-mangle| => |POSTRO
```

ENTRADA DOS PACOTES (Retorno da resposta ping acima):

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

Quando damos o ping (*echo request*) os pacotes seguem o caminho em *SAÍDA DE PACOTES* percorrendo os chains na ordem especificada e retornam via *ENTRADA DOS PACOTES* (*echo reply*). No envio da resposta da requisição de ping, o caminho de saída do pacote ignora os chains OUTPUT-nat e POSTROUTING-nat (já que não é necessário nat) mas sempre processa os chains correspondentes da tabela mangle na ordem indicada acima.

OBS1: Para conexões com destinos na própria máquina usando um endereço IP das interfaces locais, a interface será ajustada sempre para 1o (loopback).

OBS2: Em qualquer operação de entrada/saída de pacotes, os dois chains da tabela *mangle* são sempre os primeiros a serem acessados. Isto é necessário para definir a prioridade e controlar outros aspectos especiais dos pacotes que atravessam os filtros.

OBS3: O chain *OUTPUT* da tabela *filter* é consultado sempre quando existem conexões se originando em endereços de interfaces locais.

Conexão FTP de 192.168.1.1 para 192.168.1.1

- Endereço de Origem: 192.168.1.1
- Endereço de Destino: 192.168.1.1
- Interface de Origem: 1o
- Interface de Destino: 1o
- Porta Origem: 1404
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ftp (até o prompt de login, sem transferência de arquivos).

SAÍDA DOS PACOTES (envio da requisição para 192.168.1.1):

```
+-----+ +-----+ +-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-nat| => |OUTPUT-filter| => +POSTROUTING-mangle| => |POS
+-----+ +-----+ +-----+ +-----+ +-----+
```

ENTRADA DE PACOTES (respostas da requisição vindas de 192.168.1.1):

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

A requisição ftp passa através dos chains especificados em *SAÍDA DOS PACOTES* e retorna por *ENTRADA DE PACOTES*. Após a conexão ser estabelecida, o caminho de *SAÍDA DE PACOTES* será:

```
+-----+ +-----+ +-----+
```

```
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+
```

pois os dados de entrada que vem da interface externa, são passados diretamente a máquina do firewall, não necessitando de tratamento SNAT (os chains *OUTPUT-nat* e *POSTROUTING-nat* são processado somente uma vez a procura de regras que conferem, principalmente para fazer SNAT). Note novamente que mesmo não sendo necessário NAT, o chain *POSTROUTING-mangle* é checado.

OBS1: Para conexões com destinos na própria máquina usando um endereço IP das interfaces locais, a interface será ajustada sempre para `lo` (loopback).

OBS2: Em qualquer operação de entrada/saída de pacotes, os dois chains da tabela *mangle* são sempre os primeiros a serem acessados. Isto é necessário para definir a prioridade e controlar outros aspectos especiais dos pacotes que atravessam os filtros.

Conexão FTP de 192.168.1.1 para 192.168.1.4

- Endereço de Origem: 192.168.1.1
- Endereço de Destino: 192.168.1.4
- Interface de Origem: eth0
- Interface de Destino: eth0
- Porta Origem: 1405
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ftp (até o prompt de login, sem transferência de arquivos).

SAÍDA DOS PACOTES (envio da requisição para 192.168.1.4):

```
+-----+ +-----+ +-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-nat| => |OUTPUT-filter| => +POSTROUTING-mangle| => |POS
+-----+ +-----+ +-----+ +-----+ +-----+
```

ENTRADA DE PACOTES (respostas da requisição de 192.168.1.4):

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

A requisição ftp passa através dos chains especificados em *SAÍDA DOS PACOTES* com o destino 192.168.1.4 porta 21 e retorna por *ENTRADA DE PACOTES* para 192.168.1.1 porta 1405. Após a conexão ser estabelecida, o caminho de *SAÍDA DE PACOTES* será:

```
+-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+
```

pois os dados não precisam de tratamento SNAT (os chains *OUTPUT-nat* e *POSTROUTING-nat* são processado somente uma vez a procura de regras que conferem, principalmente para fazer SNAT).

OBS: Em qualquer operação de entrada/saída de pacotes, os dois chains da tabela mangle são sempre os primeiros a serem acessados. Isto é necessário para definir a prioridade e controlar outros aspectos especiais dos pacotes que atravessam os filtros.

Conexão FTP de 200.217.29.67 para a máquina ftp.debian.org.br

- Endereço de Origem: 200.217.29.67
- Endereço de Destino: 200.198.129.162
- Interface de Origem: ppp0
- Interface de Destino: ppp0
- Porta Origem: 1407
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ftp (até o prompt de login, sem transferência de arquivos).

SAÍDA DOS PACOTES (envio da requisição para 200.198.129.162):

```
+-----+ +-----+ +-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-nat| => |OUTPUT-filter| => +POSTROUTING-mangle| => |POS
+-----+ +-----+ +-----+ +-----+ +-----+
```

ENTRADA DE PACOTES (respostas da requisição vindas de 200.198.129.162):

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

A requisição ftp passa através dos chains especificados em *SAÍDA DOS PACOTES* com o destino 200.198.129.162 porta 21 (após a resolução DNS de www.debian.org.br) e retorna por *ENTRADA DE PACOTES* para 200.217.29.67 porta 1407. Após a conexão ser estabelecida, o caminho de saída de pacotes é:

```
+-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+
```

pois os dados não precisam de tratamento SNAT (os chains *OUTPUT-nat* e *POSTROUTING-nat* são processado somente uma vez a procura de regras que conferem, principalmente para fazer SNAT).

E após a conexão estabelecida, o caminho de entrada de pacotes passa a ser:

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

pois os dados não precisam de tratamento DNAT (o chain *PREROUTING-nat* é processado somente uma vez a procura de regras que conferem, principalmente para fazer DNAT).

OBS: Para qualquer operação de entrada/saída de pacotes, os dois chains da tabela mangle são sempre os primeiros a serem acessados. Isto é necessário para definir a prioridade e controlar outros aspectos especiais dos pacotes que atravessam os filtros.

Ping de 192.168.1.4 para 192.168.1.1

- Endereço de Origem: 192.168.1.4
- Endereço de Destino: 192.168.1.1
- Interface de Entrada: eth0
- Interface de Saída: eth0
- Protocolo: ICMP
- Descrição: Ping de 192.168.1.4 para a máquina do firewall.

ENTRADA DE PACOTES (recebimento da requisição, vinda de 192.168.1.4):

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |PREROUTING-nat| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+ +-----+
```

SAÍDA DE PACOTES (envio da resposta a 192.168.1.4)

```
+-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+
```

Quando damos o ping (*echo request*) os pacotes seguem o caminho em *ENTRADA DE PACOTES* percorrendo os chains na ordem especificada e retornam via *SAÍDA DOS PACOTES* (*echo reply*).

OBS1: Para qualquer operação de entrada/saída de pacotes, os dois chains da tabela mangle são sempre os primeiros a serem acessados. Isto é necessário para definir a prioridade e controlar outros aspectos especiais dos pacotes que atravessam os filtros.

Conexão FTP de 192.168.1.4 para 192.168.1.1

- Endereço de Origem: 192.168.1.4
- Endereço de Destino: 192.168.1.1
- Interface de Origem: eth0
- Interface de Destino: eth0
- Porta Origem: 1030
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ftp (até o prompt de login, sem transferência de dados).

ENTRADA DOS PACOTES (envio da requisição vindas de 192.168.1.4):

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |PREROUTING-nat| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+ +-----+
```

SAÍDA DE PACOTES (respostas da requisição acima para 192.168.1.4):

```
+-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+
```

A requisição ftp passa através dos chains especificados em *ENTRADA DOS PACOTES* com o destino 192.168.1.1 porta 21 e retorna por *SAÍDA DE PACOTES* para 192.168.1.4 porta 1030. Após a conexão ser estabelecida, o caminho de entrada de pacotes é:

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+
```

pois os dados não precisam de tratamento DNAT (o chain *PREROUTING-nat* é processado somente uma vez a procura de regras que conferem, principalmente para fazer DNAT).

OBS: O roteamento é sempre realizado após o processamento do chain *PREROUTING* da tabela *nat*.

Conexão FTP de 192.168.1.4 para ftp.debian.org.br

- Endereço de Origem: 192.168.1.4
- Endereço de Destino: 200.198.129.162
- Interface de Origem: eth0
- Interface de Destino: ppp0
- Porta Origem: 1032
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ftp (até o prompt de login, sem transferência de dados).

SAÍDA DOS PACOTES (requisição vindas de 192.168.1.4):

```
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |PREROUTING-nat| => |FORWARD-mangle| => (continua abaixo)
+-----+ +-----+ +-----+
+-----+ +-----+ +-----+
|FORWARD-filter| => |POSTROUTING-mangle| => |POSTROUTING-nat|
+-----+ +-----+ +-----+
```

ENTRADA DE PACOTES (respostas da requisição acima, enviadas para 192.168.1.4):

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |FORWARD-mangle| => |FORWARD-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+ +-----+
```

A requisição ftp passa através dos chains especificados em *SAÍDA DOS PACOTES* com o destino 200.198.129.162 porta 21 (após a resolução DNS de ftp.debian.org.br) e retorna por *ENTRADA DE PACOTES* para 192.168.1.4 porta 1032.

Note que o Masquerading regrava os pacotes; para a máquina 200.198.129.162 a conexão está sendo feita para 200.217.29.67. As respostas de conexões vindas de 200.198.129.162 e indo para 200.217.29.67 são regravadas no firewall com o destino 192.168.1.4 e enviadas para a máquina correspondente. Após a conexão ser estabelecida, o caminho de saída de pacotes para 200.198.129.163 é:

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |FORWARD-mangle| => |FORWARD-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+ +-----+
```

Após a conexão estabelecida, o caminho da entrada de pacotes vindos de 200.198.129.163 é:

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |FORWARD-mangle| => |FORWARD-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+ +-----+
```

Isto acontece porque após feita a conexão Masquerading (via PREROUTING-nat), o firewall já sabe como reescrever os pacotes para realizar a operação de Masquerading, reescrevendo todos os pacotes que chegam de www.debian.org.br para 192.168.1.4.

OBS: As conexões Masquerading feitas através da rede interna, são enviadas para 200.198.129.162 tem o endereço de origem ajustado para 200.217.29.67 que é o IP de nossa interface ppp0. Quando as respostas atravessam o firewall, os pacotes são checados pra saber se são uma resposta a uma conexão masquerading e fará a regravação dos pacotes substituindo o endereço de destino para 192.168.1.4. Caso uma operação de Masquerading falhe, os pacotes serão Bloqueados.

Conexão FTP de 200.198.129.162 para 200.217.29.167

- Endereço de Origem: 200.198.129.162
- Endereço de Destino: 200.217.29.67
- Interface de Origem: ppp0
- Interface de Destino: ppp0
- Porta Origem: 3716
- Porta Destino: 21
- Protocolo: TCP
- Descrição: Conexão ao serviço ftp do firewall

ENTRADA DOS PACOTES (requisição vinda de 200.198.129.162):

```
+-----+ +-----+ +-----+ +-----+
|PREROUTING-mangle| => |PREROUTING-nat| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+ +-----+
```

SAÍDA DE PACOTES (respostas da requisição de 200.198.129.162):

```

+-----+ +-----+ +-----+
|OUTPUT-mangle| => |OUTPUT-filter| => |POSTROUTING-mangle|
+-----+ +-----+ +-----+

```

A requisição ftp passa através dos chains especificados em *ENTRADA DOS PACOTES* com o destino 200.217.29.67 (nossa interface ppp0 local) porta 21 e retorna por *SAÍDA DE PACOTES* para 200.198.129.162 porta 3716 (também via ppp0). Após a conexão ser estabelecida, o caminho de entrada de pacotes é:

```

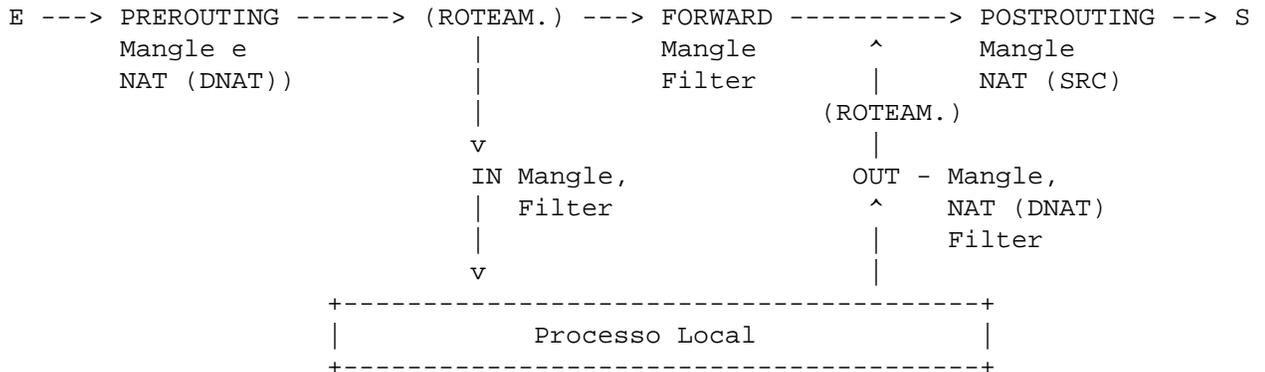
+-----+ +-----+ +-----+
|PREROUTING-mangle| => |INPUT-mangle| => |INPUT-filter|
+-----+ +-----+ +-----+

```

Isto acontece porque após feita a análise do chain *PREROUTING* (para necessidade de DNAT), a máquina já saberá tomar a decisão apropriada para gerenciar aquela conexão.

Gráfico geral da passagem dos pacotes

Este gráfico foi retirado do documento `netfilter-hacking-HOWTO.txt` e mostra a estrutura geral de passagem dos pacotes nas tabelas/chains. Os exemplos de passagem de pacotes acima poderão ser facilmente comparados com as etapas abaixo para compreender a estrutura do **iptables**.



Exemplos de configurações do iptables

Exemplo de como bloquear todas as conexões para a máquina do firewall permitindo somente conexões da máquina Linux para fora.

Bloqueando conexões de fora para sua máquina

As regras a seguir servem para bloquear tentativas de conexões da interface de Internet (ppp0) a sua rede sem bloquear o tráfego de conexões já iniciadas. O tráfego de outras interfaces não é afetado com as regras a seguir:

```
iptables -A INPUT -i ppp0 -m state --state ! ESTABLISHED,RELATED -j DROP
```

Todas as conexões vindas de ppp0 de estado diferente de ESTABLISHED e RELATED (NEW e INVALID) serão derrubadas. Veja “Conferindo de acordo com o estado da conexão” para detalhes.

```
iptables -A INPUT -i ppp0 --syn -j DROP
```

Este acima é mais simples e possui o mesmo efeito: Pacotes SYN são usados para iniciar conexões, derrubando pacotes deste tipo significa bloquear novas conexões. Pacotes de conexões já estabelecidas ainda são permitidos.

Estas regras acima servem para quem não deseja NENHUM acesso indevido a sua máquina. Existem outras formas de bloquear conexões de modo mais seletivo usando chains específicos, endereços de origem/destino, portas, etc., este tipo de configuração é muito usada caso precise fornecer algum tipo de serviço que seja acessível externamente e protegendo outros.

Monitorando tentativa de conexão de trojans em sua máquina

As regras abaixo alertam sobre a tentativa de conexão dos trojans "For Win" mais conhecidos. Coloquei isto aqui por curiosidade de algumas pessoas, pois máquinas **Linux** são imunes a este tipo de coisa:

```
#!/bin/sh

TROJAN_PORTS="12345 31336 31337 31338 3024 4092 5714 5742 2583 8787 5556 5557"

iptables -t filter -N trojans-in

for PORTA in ${TROJAN_PORTS};do
  iptables -A trojans-in -p tcp --sport=1024: --dport=${PORTA} -j LOG \
    --log-prefix "FIREWALL: Trojan ${PORTA} "
  iptables -A trojans-in -p tcp --sport=1024: --dport=${PORTA} -j DROP
done

iptables -t filter -A INPUT -i ppp0 -j trojans-in
```

A primeira linha do **iptables** cria o chain *trojans-in* dentro da tabela *filter* que usaremos para armazenar nossas regras de firewall. A segunda (dentro do laço *for*) faz uma regra de LOG para registrar as tentativas de acesso de trojans em nosso sistema, a terceira rejeita o acesso. A quarta regra do **iptables** cria de todo o tráfego vindo da interface *ppp0* pra o chain *trojans-in* (queremos que só o tráfego vindo da internet seja analisado pelo chain *trojans-in*).

Muitas das portas especificadas na variável *TROJAN_PORTS* são antigas conhecidas de quem já brincou ou sofreram com o **Back Orifice**, **Win Crack**, **NetBus** (quem nunca passou pela fase de ter uma lista com mais de 100 netmasks e conseguir encontrar centenas de máquinas por dia infectadas pelo BO? :-).

No código acima a única coisa que precisa fazer para adicionar mais portas é inseri-las na variável *TROJAN_PORTS* e executar o programa. O laço do *for* executará as 2 regras para cada porta processada (economizando linhas e linhas de regras, me livrando de uma LER e poupando muitos bytes neste guia :-).

Dependendo do número de portas alvo, este código pode ser muito simplificado usando o recurso multiport do **iptables** (veja "Especificando múltiplas portas de origem/destino" para detalhes).

Conectando sua rede interna a Internet

O seguinte exemplo permite ligar sua rede interna com a faixa de IP's *192.168.1.** a internet (usando uma conexão discada do tipo *ppp*):

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -j MASQUERADE
echo "1" >/proc/sys/net/ipv4/ip_forward
```

Um exemplo de firewall simples

Esta seção possui um exemplo mais elaborado de firewall que servirá para máquinas conectadas via ppp com uma rede interna conectada via Masquerading. Este exemplo não é tão complexo e cobre as expectativas mais comuns de pessoas que gostam de explorar os potenciais de rede no **Linux** ou satisfazer sua curiosidade. Ele poderá ser facilmente adaptado para atender outro tipo de necessidade. A configuração assumida é a seguinte:

1. Máquina do firewall com 2 interfaces de rede, uma é `eth0` com o IP `192.168.1.1` que serve de ligação a sua rede Interna, a outra é `ppp0` que é a interface Internet.
2. Qualquer acesso externo a máquinas da rede interna é bloqueado.
3. Os usuários da rede local tem acesso livre ao servidor **Linux**.
4. Qualquer acesso externo a máquina do firewall é bloqueado, exceto conexões para o serviço **Apache** (`httpd`). Outras tentativas de conexões devem ser explicitamente registradas nos logs do sistema para conhecimento do administrador.
5. Todos os usuários possuem acesso livre a Internet via Masquerading, exceto que o acesso para o serviço `www` deve ser obrigatoriamente feito via **squid**, e o servidor `smtp` a ser usado deverá ser o do firewall **Linux**.
6. Prioridades serão estabelecidas para os serviços de `telnet`, `IRC`, `talk` e `DNS`.

```
#!/bin/sh
# Modelo de configuração de firewall
# Autor: Gleydson M. Silva
# Data: 05/09/2001
# Descrição: Produzido para ser distribuído livremente, acompanha o guia
#             Foca GNU/Linux. http://www.guiafoca.org
#
# É assumido um sistema usando kmod para carga automática dos módulos usados por
# esta configuração do firewall:
# ipt_filter
# ipt_nat
# ipt_conntrack
# ipt_mangle
# ipt_TOS
# ipt_MASQUERADE
# ipt_LOG
#
# Se você tem um kernel modularizado que não utiliza o kmod, será necessário
# carregar estes módulos via modprobe, insmod ou iptables --modprobe=modulo
##### Definição de política padrão do firewall #####
# Tabela filter
iptables -t filter -P INPUT DROP
iptables -t filter -P OUTPUT ACCEPT
```

```

iptables -t filter -P FORWARD DROP
# Tabela nat
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t nat -P POSTROUTING DROP
# Tabela mangle
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P OUTPUT ACCEPT

##### Proteção contra IP Spoofing #####
for i in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 >$i
done

##### Ativamos o redirecionamento de pacotes (requerido para NAT) #####
echo "1" >/proc/sys/net/ipv4/ip_forward

# O iptables define automaticamente o número máximo de conexões simultâneas
# com base na memória do sistema. Para 32MB = 2048, 64MB = 4096, 128MB = 8192,
# sendo que são usados 350 bytes de memória residente para controlar
# cada conexão.
# Quando este limite é excedido a seguinte mensagem é mostrada:
# "ip_conntrack: maximum limit of XXX entries exceed"
#
# Como temos uma rede simples, vamos abaixar este limite. Por outro lado isto
# criará uma certa limitação de tráfego para evitar a sobrecarga do servidor.
echo "2048" > /proc/sys/net/ipv4/ip_conntrack_max

#####
#                               Tabela filter                               #
#####

##### Chain INPUT #####
# Criamos um chain que será usado para tratar o tráfego vindo da Internet e
iptables -N ppp-input

# Aceita todo o tráfego vindo do loopback e indo pro loopback
iptables -A INPUT -i lo -j ACCEPT
# Todo tráfego vindo da rede interna também é aceito
iptables -A INPUT -s 192.168.1.0/24 -i eth0 -j ACCEPT

# Conexões vindas da interface ppp0 são tratadas pelo chain ppp-input
iptables -A INPUT -i ppp+ -j ppp-input

# Qualquer outra conexão desconhecida é imediatamente registrada e derrubada
iptables -A INPUT -j LOG --log-prefix "FIREWALL: INPUT "
iptables -A INPUT -j DROP

##### Chain FORWARD #####
# Permite redirecionamento de conexões entre as interfaces locais
# especificadas abaixo. Qualquer tráfego vindo/indo para outras

```

```

# interfaces será bloqueado neste passo
iptables -A FORWARD -d 192.168.1.0/24 -i ppp+ -o eth0 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -o ppp+ -j ACCEPT
iptables -A FORWARD -j LOG --log-prefix "FIREWALL: FORWARD "
iptables -A FORWARD -j DROP

##### Chain ppp-input #####
# Aceitamos todas as mensagens icmp vindas de ppp0 com certa limitação
# O tráfego de pacotes icmp que superar este limite será bloqueado
# pela regra "...! ESTABLISHED,RELATED -j DROP" no final do
# chain ppp-input
#
iptables -A ppp-input -p icmp -m limit --limit 2/s -j ACCEPT

# Primeiro aceitamos o tráfego vindo da Internet para o serviço www (porta 80)
iptables -A ppp-input -p tcp --dport 80 -j ACCEPT

# A tentativa de acesso externo a estes serviços serão registrados no syslog
# do sistema e serão bloqueados pela última regra abaixo.
iptables -A ppp-input -p tcp --dport 21 -j LOG --log-prefix "FIREWALL: ftp "
iptables -A ppp-input -p tcp --dport 25 -j LOG --log-prefix "FIREWALL: smtp "
iptables -A ppp-input -p udp --dport 53 -j LOG --log-prefix "FIREWALL: dns "
iptables -A ppp-input -p tcp --dport 110 -j LOG --log-prefix "FIREWALL: pop3 "
iptables -A ppp-input -p tcp --dport 113 -j LOG --log-prefix "FIREWALL: identd "
iptables -A ppp-input -p udp --dport 111 -j LOG --log-prefix "FIREWALL: rpc"
iptables -A ppp-input -p tcp --dport 111 -j LOG --log-prefix "FIREWALL: rpc"
iptables -A ppp-input -p tcp --dport 137:139 -j LOG --log-prefix "FIREWALL: samba"
iptables -A ppp-input -p udp --dport 137:139 -j LOG --log-prefix "FIREWALL: samba"
# Bloqueia qualquer tentativa de nova conexão de fora para esta máquina
iptables -A ppp-input -m state --state ! ESTABLISHED,RELATED -j LOG --log-prefix "
iptables -A ppp-input -m state --state ! ESTABLISHED,RELATED -j DROP
# Qualquer outro tipo de tráfego é aceito
iptables -A ppp-input -j ACCEPT

#####
#                               Tabela nat                               #
#####

##### Chain POSTROUTING #####
# Permite qualquer conexão vinda com destino a lo e rede local para eth0
iptables -t nat -A POSTROUTING -o lo -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j ACCEPT

# Não queremos que usuários tenham acesso direto a www e smtp da rede externa, o
# squid e smtpd do firewall devem ser obrigatoriamente usados. Também registramos
# as tentativas para monitorarmos qual máquina está tentando conectar-se diretamen
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -p tcp --dport 80 -j LOG
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -p tcp --dport 25 -j LOG
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -p tcp --dport 25 -j DROP
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -p tcp --dport 80 -j DROP
# É feito masquerading dos outros serviços da rede interna indo para a interface
# ppp0

```

```

iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o ppp+ -j MASQUERADE

# Qualquer outra origem de tráfego desconhecida indo para eth0 (conexões vindas
# de ppp+) são bloqueadas aqui
iptables -t nat -A POSTROUTING -o eth0 -d 192.168.1.0/24 -j LOG --log-prefix "FIRE
iptables -t nat -A POSTROUTING -o eth0 -d 192.168.1.0/24 -j DROP
# Quando iniciamos uma conexão ppp, obtermos um endereço classe A (10.x.x.x) e após
# estabelecida a conexão real, este endereço é modificado. O tráfego indo para
# a interface ppp não deverá ser bloqueado. Os bloqueios serão feitos no
# chain INPUT da tabela filter
iptables -t nat -A POSTROUTING -o ppp+ -j ACCEPT

# Registra e bloqueia qualquer outro tipo de tráfego desconhecido
iptables -t nat -A POSTROUTING -j LOG --log-prefix "FIREWALL: SNAT "
iptables -t nat -A POSTROUTING -j DROP

#####
#                               Tabela mangle                               #
#####

##### Chain OUTPUT #####
# Define mínimo de espera para os serviços ftp, telnet, irc e DNS, isto
# dará uma melhor sensação de conexão em tempo real e diminuirá o tempo
# de espera para conexões que requerem resolução de nomes.
iptables -t mangle -A OUTPUT -o ppp+ -p tcp --dport 21 -j TOS --set-tos 0x10
iptables -t mangle -A OUTPUT -o ppp+ -p tcp --dport 23 -j TOS --set-tos 0x10
iptables -t mangle -A OUTPUT -o ppp+ -p tcp --dport 6665:6668 -j TOS --set-tos 0x1
iptables -t mangle -A OUTPUT -o ppp+ -p udp --dport 53 -j TOS --set-tos 0x10

```

Capítulo 11. Gerenciamento de contas e cuidados para a proteção de senhas

Este capítulo traz explicações e comandos úteis para o gerenciamento de contas e proteção de senhas de usuários em sistemas **Linux**. Também explica os principais métodos usados para quebra de senha usando diversos métodos como engenharia social, brute force, etc., bem como dicas de como escolher boas senhas para você e seus usuários e métodos automatizados de checagem de senhas vulneráveis.

Estes métodos são explicados para que você entenda, se previna destes tipos de ataques além de entender a importância de políticas de proteção de senhas.

Introdução

A criação de uma conta em uma máquina **Linux** pode expor seu sistema (ou todas suas redes) a crackers simplesmente com a falta de treinamento e políticas de segurança. Um invasor com um simples acesso a uma conta de usuário pode conseguir acesso a áreas que contém dados importantes expondo seu sistema a ataques ou roubo de dados.

Um firewall não pode fazer muito em uma situação dessas, um acesso através de uma conta de sistema válida é difícil de ser auditado e descoberto, a não ser que o usuário monitore seus acesso via **lastlog** e o administrador conheça os hábitos de seus usuários para notar o uso estranho de contas em determinados dias/horários. Evitar situações como esta depende mais de conscientização e treinamento tanto do administrador como dos usuários das contas para não expor o sistema a um ataque direto. Este capítulo do guia explicará as situações mais comuns e alguns exemplos de como tais ataques acontecem.

ATENÇÃO: - Os dados aqui disponibilizados são puramente para fins didáticos e compreensão de como tais situações funcionam para se criar mecanismos de defesa personalizados de acordo com o que deseja proteger.

Criação, monitoramento e segurança de contas

Para adicionar uma conta de usuário ao sistema é simples, basta um comando `adduser [usuário]` e alguns poucos segundos para responder as questões do programa. Quando criamos contas para outros usuários temos 2 alternativas: deixarmos a senha em branco ou escolher uma senha que será passada ao usuário para que ele possa fazer a troca mais tarde. A primeira alternativa é muito perigosa, pois uma pessoa com acesso a `/etc/passwd` poderá facilmente descobrir sua lista de usuários (principalmente em uma grande empresa quando conhecemos as políticas de criação de novas contas). Um funcionário notaria a presença do novato e poderia aproveitar esta oportunidade para tentar incriminar este usando a conta recém criada ou tentar outras coisas para obter benefício próprio através do descuido de outros.

O segundo método de senha inicial é um pouco mais seguro e de preferência a senha deve ser escolhida pelo usuário para que pessoas que conhecem o estilo de senhas iniciais escolhidas pelo administrador não possam deduzir a nova senha criada. É comum vermos senhas como "novo1234", "123456", "abcdef", "a1b3c3", o "nome do usuário" como senhas iniciais, pois é fácil de lembrar. Senhas deste tipo são as primeiras a ser tentadas por crackers e programas específicos para este fim. Mas se o o usuário esquecer de trocar sua senha provisória?

O programa **chage** e **passwd** possui recursos que permitem definir limites mínimos e máximo do tempo para troca de senha de acesso, número máximo de dias após expirar o tempo de troca da senha em que a conta será permanentemente desabilitada (até que o administrador a reative) e o período mínimo entre troca de senhas. Alguns exemplos:

```
passwd -x 10 -w 3 teste
```

A senha do usuário teste expirará após 10 dias (-x 10) e ele será avisado com 3 dias de antecedência (-w 3) para trocar sua senha. Após o período máximo o usuário será obrigado a trocar a senha.

Quando o usuário efetuar o login receberá a seguinte mensagem: `Warning: your password will expire in 3 days.`

```
passwd -x 10 -w 3 -i 2 teste
```

A senha do usuário teste expirará após 10 dias (-x 10) e ele será avisado com 3 dias de antecedência (-w 3) para trocar sua senha, após a expiração da senha, o usuário tem 2 dias antes da conta ser desativada (-i 2). Se o período expirar e o usuário tentar um novo login será mostrada a mensagem:

```
Your account has expired: Please contact your system administrator
```

Para reativar a conta acima, remova totalmente o bloqueio da conta do usuário teste com `passwd -x 0 teste`, `passwd -x 99999 -w 7 -i 0 teste` ou especifique um período de dias maior em adição àqueles especificados para que ele possa trocar a senha.

Por exemplo, caso tenha passado 3 dias desde que a conta acima expirou e deseje dar mais 2 dias para o usuário trocar a conta: `passwd -x 17 -i 0 teste` A conta será reativada por mais 2 dias dando a oportunidade do usuário trocar a senha. Preste atenção neste exemplo para entender bem a situação e prazos.

```
passwd -x 90 -n 60 -w 15 -i 0 teste
```

A senha do usuário teste expirará após 90 dias (-x 90), ele será avisado para trocar sua senha com 15 dias antes do prazo final (-w 15) e a conta será imediatamente desativada caso o prazo máximo para troca da senha expire (-i 0). O usuário também não poderá trocar sua senha durante os primeiros 60 dias desde a última troca de senha (-n 60).

Em sistemas onde precisa adicionar restrições a muitos usuários na criação da conta, é recomendável seguir os métodos descritos em “Definindo valores padrões de restrição”.

OBS1: Em sistemas com senhas ocultas ativadas (veja “Shadow Passwords”) as restrições acima serão especificadas no arquivo `/etc/shadow`, isto garante que só o usuário `root` tenha acesso aos detalhes fornecidos neste arquivo.

OBS2: A `-d` do `passwd` serve para remover a senha do usuário especificado ou seja somente será necessário fornecer o nome de usuário para ter acesso ao sistema.

OBS3: Leve em consideração que o uso do recursos de senhas de grupo é um risco de segurança, pois a mesma senha será compartilhada entre diversas pessoas.

OBS4: O programa `useradd` combina as funções do `adduser` e `passwd` para garantir que a conta seja criada com as restrições apropriadas. O único inconveniente é que o `useradd` quebra o *Debian Policy* e precisa de todos todos os parâmetros para a criação correta da conta (como o diretório home, senha criptografada, e UID numérico). Seu uso é indicado em shell scripts que cuidam automaticamente da tarefa de adicionar usuários ao sistema.

Definindo valores padrões de restrição

Isto é muito útil quando precisa criar diversos usuários com as mesmas restrições de contas, isto tornará o gerenciamento do sistema muito mais prático (tudo em Unix é feito para ser mais prático, só devemos

saber onde mexer). O arquivo `/etc/defaults/useradd` contém valores padrões que serão usados pelo **useradd** e **adduser** para definir valores de restrições de contas. Estes valores são gerados usando a opção `-D` em combinação com as seguintes opções do **useradd**:

- `-b [home]` - Especificar o diretório home de usuário. O padrão é `/home`.
- `-e [data]` - Data padrão de expiração de contas, especificada no formato `AnoMesDia`. Por exemplo, `20010920`.
- `-f [dias]` - Número máximo de dias que a conta permanece válida após a data de expiração até ser desativada.
- `-g [gid/grupo]` - ID do grupo ou nome do grupo que o usuário pertencerá inicialmente.
- `-s [shell]` - Shell do usuário. O padrão é `/bin/bash`.

OBS: Note que nem todas as opções acima terão efeito com o **adduser** (principalmente as opções `-f`, `-g` e `-s` que são especificadas no seu arquivo de configuração `/etc/adduser.conf`).

Senhas fáceis de adivinhar e escolha de boas senhas

A senha lhe identifica como o verdadeiro dono de uma conta em um sistema para garantir acesso a seus recursos. A senha de um sistema é tão importante quanto uma senha de sua conta bancária, caso caia em mãos erradas as consequências poderão ser catastróficas, todo cuidado é pouco na hora de escolher uma senha.

Senhas fáceis de adivinhar são o primeiro motivo de sucesso de crackers no acesso a sistemas de computadores (veja “Dedução” e “Engenharia Social”), o administrador pode forçar o usuário a fazer trocas periódicas de senhas através dos recursos citados em “Criação, monitoramento e segurança de contas”, mas quem vai garantir que ele esteja escolhendo boas senhas para que ninguém as descubra com facilidade? Abaixo uma lista de senhas ruins (que deverá evitar a todo custo usa-las) e boas:

Senhas Ruins

- O uso da palavra *senha* como senha! Isto parece idiota mais existe...
- Senhas com o mesmo nome do login (`joao/joao`).
- Compostas por letras ou números em sequência crescente ou decrescente (`abcdef`, `123456`, `654321`, etc, etc). Este tipo de senha pode ser adivinhada por dedução e são uma das primeiras combinações que crackers usam para acertar senhas.
- palavras relacionadas com o gosto pessoal. Por exemplo "escort", "vectra", "subaru" se a pessoa é amante de carros.
- Nome da esposa, filhos, familiares, animal de estimação, time de futebol, ídolo da TV/filmes ou qualquer coisa relacionada a familiares ou indiretamente ao usuário.
- Idade, data de aniversário, data de casamento, número de identidade, título de eleitor, placa de carro ou qualquer coisa que seja característica do usuário.
- Palavras existentes. Um ataque de dicionário poderá descobrir facilmente sua senha.
- Senhas com menos de 8 letras
- Senhas apenas em minúsculas ou MAIÚSCULAS.

Senhas Boas

- Uma boa senha nunca deverá ser lida mas fácil de lembrar. Por exemplo pense em uma frase importante para você "meu sistema operacional preferido é o Linux" e pegue a primeira letra de cada palavra: "msopeol". PRONTO esta escolhida uma boa senha que é fácil de se lembrar e difícil de ser quebrada por ataques de dicionário!
- Uma boa senha deve conter números e letras. A senha acima poderia ser modificada para "msopeol1"
- Conter letras maiúsculas e minúsculas. "msopeoL1".
- Conter 8 caracteres sempre que possível. Isto aumenta bastante o número de combinações necessárias para se quebrar uma senha em um ataque brute force (veja "Brute Force"). Mesmo que a senha escolhida não chegue a 8 caracteres mínimos, você poderá combina-la com números.

Com as dicas acima, a possibilidade de alguém conseguir quebrar uma senha criptografada em seu sistema usando os ataques descritos em "Tipos de ataques mais comuns para se conseguir uma senha." é praticamente nula! Para os paranóicos de plantão, o utilitário **makepasswd** pode criar uma senha com caracteres completamente aleatórios:

```
makepasswd --chars 8  
4y0sBdwM
```

Este comando retorna uma string com 8 caracteres (--chars 8) "4y0sBdwM". Se você entendeu boa parte deste guia tenho certeza que 1 ou 2 dias de treino e se acostuma com uma senha como esta ;-)

OBS: NUNCA NUNCA dê pistas sobre sua senha! Para você isto pode ser um desafio lançado a outras pessoas quase impossível de ser resolvido, mas não se esqueça que muita gente é especializada neste tipo de dedução.

Atualização de senhas de múltiplas contas

O programa **chpasswd** é usado para tal operação. Deve ser especificado um arquivo que contém os campos `usuário:senha` por linha. Caso as senhas estejam encriptadas deverá ser especificada a opção `-e` ao programa.

```
chpasswd -e /localadmin/contas/contas.db
```

O comando acima atualiza a senha de todos os usuários especificados no arquivo `contas.db` de uma só vez.

A senha do usuário root

Esta seção foi retirada do Manual de Instalação da Debian.

A conta root é também chamada de *super usuário*, este é um login que não possui restrições de segurança. A conta root somente deve ser usada para fazer a administração do sistema, e usada o menor tempo possível.

Qualquer senha que criar deverá conter de 6 a 8 caracteres, e também poderá conter letras maiúsculas e minúsculas, e também caracteres de pontuação. Tenha um cuidado especial quando escolher sua senha root, porque ela é a conta mais poderosa. Evite palavras de dicionário ou o uso de qualquer outros dados pessoais que podem ser adivinhados.

Se qualquer um lhe pedir senha root, seja extremamente cuidadoso. Você normalmente nunca deve distribuir sua conta root, a não ser que esteja administrando um computador com mais de um administrador do sistema.

Utilize uma conta de usuário normal ao invés da conta root para operar seu sistema. Porque não usar a conta root? Bem, uma razão para evitar usar privilégios root é por causa da facilidade de se cometer danos irreparáveis como root. Outra razão é que você pode ser enganado e rodar um programa *Cavalo de Tróia* -- que é um programa que obtém poderes do *super usuário* para comprometer a segurança do seu sistema sem que você saiba.

Tipos de ataques mais comuns para se conseguir uma senha.

Dedução

O cracker se aproveita da ingenuidade de usuários que deixam senhas em branco, usam senhas simples como o próprio nome, "abcdef", "asdfg", "123456", e outros tipos de senhas comuns para tentar obter acesso ao sistema. Senhas deduzidas são geralmente senhas muito simples e muito usadas... Uma situação comum para a escolha de uma senha deste tipo é o medo de esquecer a senha (quando não se consegue pensar em algo mais difícil e ao mesmo tempo que seja fácil de lembrar) e quando o usuário é pego desprevenido e não se sabe o que usar como senha (como na assinatura de um provedor Internet, muito comum essa situação).

Geralmente é muito rápido e muito eficaz dependendo das habilidades do atacante dispõe.

Engenharia Social

Ataques por engenharia social são feitos através de pesquisa de dados pessoais e outras características relacionadas ao usuário (time de futebol, data de nascimento dele, da esposa, filhos, nome da atriz predileta, etc) e usando estes dados coletados para auxiliar na descoberta da senha. Este ataque requer uma pesquisa sobre os hábitos, gostos, etc. Mas existem outros tipos de ataque baseados em engenharia social, inclusive com o cracker passando-se pelo usuário. Para diminuir as possibilidades deste tipo de ataque entenda e siga os procedimentos da parte "Senhas Boas" na "Senhas fáceis de adivinhar e escolha de boas senhas" e continue lendo esta seção.

Outro detalhe importante para diminuir as possibilidades de um ataque deste tipo bem sucedido é permitir somente o acesso do serviço de **finger** a redes confiáveis (locais onde uns conhecem os outros). Os detalhes fornecidos pelo finger podem ser suficientes para garantir sucesso deste tipo de ataque:

```
#finger joao
Login: joao                Name: Joao P. M.
Directory: /home/joao     Shell: /bin/bash
Office: Sala 400 Andar 2, 123-4567 Home: 123-7654
Last login Fri Aug 25 21:20 (AMT) on tty3
No mail.
Grupo de cadastramento.
```

As últimas linhas da saída do finger são os dados contidos nos arquivos `.plan` e `.project` do diretório de usuário. O cracker com base nos dados fornecidos acima pelo finger poderia inventar uma situação em que necessitaria de troca de senha por algum motivo. Abaixo uma situação onde o cracker sabe que não existe identificador de chamadas na empresa e conhece as fragilidades:

- Cracker: Disca para o CPD?
- Vitima: CPD?
- Cracker: Oi, eu sou o Joao P. M. do grupo de cadastramento aqui do segundo andar, estou tentando entrar no sistema mas por algum motivo ele não aceita minha senha (fazendo-se de ignorante no assunto).
- Vitima: Por favor Sr. verifique se o Caps Lock do seu teclado está ativado, letras em maiúsculas/ minúsculas fazem diferença em nossos sistemas.
- Cracker: Ok vou checar (espera um tempo). Não, esta tudo Ok, você poderia agilizar isto de alguma maneira, preciso lançar algumas fichas no sistema.
- Vitima: Posso modificar sua senha para um nome qualquer, depois você poderá trocar por si próprio.
- Cracker: Ok, por mim tudo bem.
- Vitima: Humm, modifiquei para "cad1234", basta você usa-la e terá acesso ao sistema. Após isso execute o utilitário passwd para troca-la para algo que desejar.
- Cracker: Ok, muito obrigado. Tenha um bom dia.

Este é um exemplo simples de ataque por engenharia social. Dependendo do objetivo, este tipo de ataque pode levar semanas e as vezes requer contatos com diversas empresas criando diversas situações para obter detalhes necessários para atingir o objetivo.

As políticas de segurança de senhas minimizam riscos deste tipo. Como este é um caso que o requisitante é um funcionário próximo do departamento de informática, o mais adequado seria o administrador se deslocar ao setor (ou enviar um técnico do setor treinado para tal situação) para saber se quem diz ser quem é está realmente no local enfrentando aquela situação. O contato com o responsável do setor (conhecido do técnico) também pode ser uma alternativa antes de entregar uma senha a um desconhecido.

Para casos externos (principalmente para empresas que mantém determinados serviços em funcionamento em nosso servidor, como servidores de páginas), o procedimento correto seria passar uma nova senha por e-mail (de preferência criptografado com **pgp**) ao invés de telefone. Isto garantirá que a senha não caia nas mãos erradas.

OBS1: Qualquer detalhe sobre a política de criação de senhas, trocas de senhas, etc. poderá ter muito valor para um cracker obter acesso ao seu sistema.

OBS2: Dificulte as maneiras para se obter acesso `root` ao sistema via conta de usuário comum. É de extrema importância utilizar conexões de dados criptografadas quando for necessário acesso externo ao seu sistema.

OBS3: Nunca use uma mesma senha para fazer tudo (banco, acessar seu sistema, conectar-se ao seu provedor, senha de root). Você estará em sérios apuros caso alguém tenha acesso a esta senha. É difícil lembrar de várias senhas, mas você pode aditar uma senha e criar modificações a partir dela para utilização em outros locais, por exemplo: "wekpdm" => "Bwekpdm1" => "3wekpdmS", etc.

Ataques por dicionário

De posse do arquivo de senhas `/etc/passwd`, o cracker utiliza um arquivo que contém diversas palavras que serão tentadas como senha. Este trabalho é feito automaticamente por ferramentas dedicadas a este tipo de tarefa e pode levar dias dependendo da lista de senhas do cracker e quantidades de usuários existentes no arquivo de senha.

Note que o uso de criptografia `md5` e *senhas ocultas* dificultam bastante ao arquivo de senhas e o sucesso de um ataque bem sucedido (veja "Shadow Passwords" e "Senhas MD5").

Brute Force

De posse do arquivo de senhas `/etc/passwd` o cracker utiliza uma ferramenta que tenta diversas combinações de letras sequencialmente na tentativa de descobrir uma senha. Este ataque geralmente é usado como último recurso após um ataque por dicionário, e leva muito tempo para descobrir uma senha.

Dependendo se uma senha conter caracteres aleatórios, combinação de letras maiúsculas/minúsculas, números, a senha será praticamente impossível de ser descoberta. Note que o uso de criptografia *md5* e *senhas ocultas* aumentam bastante a proteção das senhas (veja “Shadow Passwords” e “Senhas MD5”).

Monitoramento de toques do teclado

Este ataque é muito comum em sistemas DOS e Windows, um programa é instalado sem o conhecimento do usuário que grava todos os toques do teclado em um arquivo escondido pelo cracker. Após certo tempo o cracker obtém acesso ao arquivo e aos dados que ele contém. Este tipo de ataque é muito perigoso e pode capturar senhas não só do sistema como números de cartão de crédito digitados (caso o usuário tenha feito compras on-line), conta bancária+senha e tudo mais que for digitado pelo teclado.

Login falso

Esta é uma forma rápida de se conseguir acesso a um sistema. É criada uma tela de login idêntica a original do sistema, só que ao digitar nome e senha, estes são gravados em um arquivo (que será mais tarde recuperado pelo cracker para obter acesso ao sistema) e uma mensagem de erro será exibida pelo sistema.

Naturalmente o usuário pensará que digitou o nome/senha incorretamente e fará uma nova tentativa, a segunda ocorrerá com sucesso (fazendo este pensar que errou *mesmo* a senha).

Sua atenção é muito importante para evitar este tipo de ataque, caso desconfie de algo errado, entra no sistema e dê um `find --type f -cmin -3` para localizar os arquivos modificados nos últimos 3 minutos e localizar possíveis bancos de dados de senhas.

Outra alternativa é realmente digitar uma senha inválida intencionalmente (e diferente da correta) e na segunda tentativa lançar a senha válida (normalmente sistemas deste tipo bem elaborados chamam o verdadeiro sistema de login na segunda tentativa).

Melhorando a segurança das senhas armazenadas em seu sistema

Shadow Passwords

Senhas Ocultas (shadow passwords) aumentam consideravelmente a senha do seu sistema pois as senhas serão armazenadas em um arquivo separado: `/etc/shadow` para senhas de usuários e `/etc/gshadow` para senhas de grupos. Estes dois arquivos poderão ser acessados somente pelo usuário root. O armazenamento de senhas no arquivo `/etc/passwd` e `/etc/groups` não é seguro, estes arquivos devem ser lidos por todos os usuários porque muitos programas mapeiam a UID do usuário com seu nome e vice versa.

O utilitário **shadowconfig** é usado para ativar/desativar o suporte a senhas ocultas (de usuários e grupos) em seu sistema. Adicionalmente os utilitários **pwconv/grpconv** podem ser usados separadamente para ativar o suporte a senhas ocultas de usuários/grupos e **pwunconv/grpunconv** para desativar este suporte.

ATENÇÃO: Caso você inclua usuários em grupos manualmente no arquivo `/etc/passwd`, também precisará fazer isto no arquivo `/etc/shadow` para que não tenha problemas. Esta tarefa é feita automaticamente com o comando `adduser usuário grupo`. O programa **vipw** e **visudo** também podem ser usados com a opção `-s` para editar os arquivos `/etc/shadow` e `/etc/gshadow` respectivamente.

Senhas MD5

O sistema de criptografia usado pelas senhas MD5 é mais seguro que o padrão Crypto e permitem o uso de senhas maiores do que 8 caracteres.

O uso de senhas MD5 é recomendado para aumentar o nível de proteção da senha. Não use caso estiver executando um serviço de NIS. **OBS:** Caso utilize senhas MD5 em um sistema com PAM, inclua a palavra `md5` na linha de configuração do método de autenticação `password` do módulo `pam_unix.so`:

```
password required pam_unix.so md5
```

Capítulo 12. Apache

Esta capítulo documenta a configuração, personalização, introdução aos mecanismos de autenticação e controle de acesso do **Apache**, sistema proxy, virtual hosting, e exemplos de configuração do servidor **httpd**. Ele não tem como objetivo ser uma referência completa de configuração, mas sim abordar didaticamente o assunto.

Introdução

O servidor web é um programa responsável por disponibilizar páginas, fotos, ou qualquer outro tipo de objeto ao navegador do cliente. Ele também pode operar recebendo dados do cliente, processando e enviando o resultado para que o cliente possa tomar a ação desejada (como em aplicações CGI's, banco de dados web, preenchimento de formulários, etc).

O **Apache** é um servidor Web extremamente configurável, robusto e de alta performance desenvolvido por uma equipe de voluntários (conhecida como `Apache Group`) buscando criar um servidor web com muitas características e com código fonte disponível gratuitamente via Internet. Segundo a Netcraft (<http://www.netcraft.com/>), o **Apache** é mais usado que todos os outros servidores web do mundo juntos.

Este capítulo não tenta ser um guia completo ao **Apache**, mas tentará mostrar como sua estrutura é organizada, as diretivas principais de configuração, diretivas de segurança, virtual hosting, proxy, o uso de utilitários de gerenciamento do servidor, como personalizar algumas partes do servidor e programas úteis de terceiros para análise e diagnóstico do servidor web. Não deixe também de ver “Exemplo comentado de um arquivo de configuração do Apache” pois contém diretivas básicas de configuração comentadas e explicações interessante e faz parte do aprendizado.

Versão

É assumido que esteja usando a versão 1.3.22 do **apache**. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendável que leia a documentação sobre modificações no programa (changelog) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

Um resumo da História do Apache

O **Apache** tem como base o servidor web **NCSA 1.3** (*National Center of Supercomputing Applications*), que foi desenvolvido por Rob McCool. Quando Rob deixou o NCSA, o desenvolvimento foi interrompido, assim muitos desenvolvedores buscaram personalizar sua própria versão do NCSA ou adicionar mais características para atender as suas necessidades. Neste momento começa a história do **Apache** com *Brian Behlendorf* e *Cliff Skolnick* abrindo uma lista de discussão para interessados no desenvolvimento, conseguindo espaço em um servidor doado pela *HotWired* e trocando patches corrigindo problemas, adicionando recursos e discutindo idéias com outros desenvolvedores e hackers interessados neste projeto.

A primeira versão oficial do **Apache** foi a 0.6.2, lançada em Abril de 1995 (neste período a NCSA retomava o desenvolvimento de seu servidor web, tendo como desenvolvedores *Brandon Long* e *Beth Frank* que também se tornaram membros especiais do grupo **Apache**, compartilhando idéias sobre seus projetos).

Nas versões 2.x do Apache, a escalabilidade do servidor foi ampliada suportando as plataformas **Win32** (não obtendo o mesmo desempenho que em plataformas UNIX mas sendo melhorado gradativamente).

Enviando Correções/Contribuindo com o projeto

Um formulário está disponível na Web para o envio de correções/sugestões em http://www.apache.org/bug_report.html/. Uma lista de anuncio sobre o **Apache** está disponível em [<apache-announce@apache.org>](mailto:apache-announce@apache.org) que divulgam correções, novas versões e realização de eventos.

Mais detalhes sobre o desenvolvimento do Apache podem ser visualizadas na URL <http://dev.apache.org/>.

Características do Apache

Abaixo estão algumas características que fazem esse servidor web o preferido entre os administradores de sistemas:

- Possui suporte a scripts cgi usando linguagens como *Perl, PHP, Shell Script, ASP, etc.*
- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor.
- Autenticação requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto e MD5).
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador.
- Suporte a tipos mime.
- Personalização de logs.
- Mensagens de erro.
- Suporte a virtual hosting (é possível servir 2 ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço).
- Suporte a IP virtual hosting.
- Suporte a name virtual hosting.
- Suporte a servidor Proxy ftp e http, com limite de acesso, caching (todas flexivelmente configuráveis).
- Suporte a proxy e redirecionamentos baseados em URLs para endereços Internos.
- Suporte a criptografia via SSL, Certificados digitais
- Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa.

Ficha técnica

Pacote:

Utilitários:

- `apache` - Servidor Web Principal
- `apachectl` - Shell script que faz interface com o **apache** de forma mais amigável
- `apacheconfig` - Script em Perl para configuração interativa básica do **Apache**

- `htpasswd` - Cria/Gerencia senhas criptografadas Crypto/MD5
- `htdigest` - Cria/Gerencia senhas criptografadas Crypto/MD5
- `dbmmanage` - Cria/Gerencia senhas em formato DBM (Perl)
- `logresolve` - Faz um DNS reverso dos arquivos de log do **Apache** para obter o endereço de hosts com base nos endereços IP's.
- `ab` - Apache Benchmarking - Ferramenta de medida de desempenho do servidor Web Apache.

Por padrão, os arquivos de configuração do **Apache** residem no diretório `/etc/apache`:

<code>httpd.conf</code>	Arquivo de configuração principal do Apache , possui diretivas que controlam a operação do daemon servidor. Um arquivo de configuração alternativo pode ser especificado através da opção "-f" da linha de comando.
<code>srm.conf</code>	Contém diretivas que controlam a especificação de documentos que o servidor oferece aos clientes. O nome desse arquivo pode ser substituído através da diretiva <code>ResourceConfig</code> no arquivo principal de configuração.
<code>access.conf</code>	Contém diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva <code>AccessConfig</code> no arquivo principal de configuração.

O servidor Web lê os arquivos acima na ordem que estão especificados (`httpd.conf`, `srm.conf` e `access.conf`). As configurações também podem ser especificadas diretamente no arquivo `httpd.conf`. Note que não é obrigatório usar os arquivos `srm.conf` e `access.conf`, mas isto proporciona uma melhor organização das diretivas do servidor, principalmente quando se tem um grande conjunto de diretivas. Um exemplo comentado destes três arquivos de configuração é encontrado em "Exemplo comentado de um arquivo de configuração do Apache".

Requerimentos

A máquina mínima para se rodar um servidor **Apache** para atender a uma rede padrão 10MB/s é um Pentium 90, 24MB de RAM, um HD com um bom desempenho e espaço em disco considerável de acordo com o tamanho projetado de seu servidor web (considerando seu crescimento).

Uma configuração mais rápida para redes 100MB/s teria como processador um Cyrix MX ou Intel Pentium MMX como plataforma mínima (Cyrix é o recomendado pelo alto desempenho no processamento de strings), barramento de HD SCSI com uma boa placa controladora (Adaptec 19160 ou superior) com 64MB de RAM no mínimo.

Arquivos de log criados pelo Apache

O servidor **httpd** grava seus arquivos de log geralmente em `/var/log/apache`, não é possível descrever os arquivos de logs usados porque tanto seus nomes como conteúdo podem ser personalizados no arquivo `httpd.conf`. Mesmo assim, os arquivos de logs encontrados na instalação padrão do **Apache** são os seguintes:

- `access.log` - Registra detalhes sobre o acesso as páginas do servidor **httpd**.
- `error.log` - Registra detalhes saber erros de acesso as páginas ou erros internos do servidor.
- `agent.log` - Registra o nome do navegador do cliente (campo `UserAgent` do cabeçalho http).

Mais referências podem ser encontradas em "Sistema de Log do Apache". Um bom programa para geração de estatísticas de acesso com gráficos é o "Relatório gráfico de acesso ao sistema".

Instalação

```
apt-get install apache apache-doc
```

(o pacote contém a documentação de referencia do **Apache**, é recomendável instala-lo se estiver curioso e deseja entender melhor seu funcionamento ou consultar diretivas).

Iniciando o servidor/reiniciando/recarregando a configuração

O **Apache** pode ser executado tanto como um servidor *Inetd* ou como um *Daemon*. A inicialização de programas pelo *Inetd* é uma boa estratégia quando você precisa de um controle de acesso básico (o fornecido pelo **tcpd**), e o serviço é pouco usado na máquina.

A segurança de um serviço iniciado pelo **inetd** pode ser substituída e melhorada por um firewall bem configurado, garantindo facilidades extras como um relatório de tráfego para a porta do servidor web, por exemplo. Mesmo assim se o servidor **Apache** estiver rodando como daemon e estiver ocioso, ele será movido para swap liberando a memória RAM para a execução de outros programas.

Neste capítulo será assumido seu funcionamento do **Apache** como Daemon, que é o método de funcionamento recomendado para sites de grande tráfego onde ele é freqüentemente requisitado e considerado um serviço crítico.

O método padrão para iniciar programas como daemons na **Debian** é através dos diretórios `/etc/rc?.d`. Cada diretório deste contém os programas que serão executados/interrompidos no nível de execução "?" (`rc1.d/`, `rc2.d/` ...). O conteúdo destes diretórios são links para os scripts originais em `/etc/init.d/programa`, o nosso programa alvo é `/etc/init.d/apache`. O `/etc/init.d/apache` aceita os seguintes parâmetros:

- `start` - Inicia o **Apache**
- `stop` - Finaliza o **Apache**
- `restart` - Reinicia o **Apache**, efetuando uma pausa de 5 segundos entre a interrupção do seu funcionamento e reinício.
- `reload` - Recarrega os arquivos de configuração do Apache, as alterações entram em funcionamento imediatamente.
- `reload-modules` - Recarrega os módulos. Basicamente é feito um restart no servidor.
- `force-reload` - Faz a mesma função que o reload

Para reiniciar o **Apache** usando o `/etc/init.d/apache`, digite:

```
./etc/init.d/apache restart
```

ou

```
cd /etc/init.d;./apache restart
```

Na realidade, o que o `/etc/init.d/apache` faz é interagir diretamente com o shell script **apachectl**.

O **apachectl** recebe os parâmetros enviados pelo usuário e converte para sinais que serão enviados para o binário **apache**. Da mesma forma ele verifica os códigos de saída do **apache** e os transforma em mensagens de erro legíveis para o usuário comum. Os seguintes comandos são aceitos pelo **apachectl**:

- `httpd-server/start` - Inicia o **Apache**
- `stop` - Finaliza o **Apache** (enviando um sinal TERM)
- `restart` - Reinicia o **Apache** (enviando um sinal HUP)
- `graceful` - Recarrega os arquivos de configuração do **Apache** (enviando um sinalUSR1)
- `fullstatus` - Mostra o status completo do servidor **Apache** (requer o **lynx** e o módulo `mod_status` carregado).
- `status` - Mostra o status do processo do servidor **Apache** (requer o **lynx** e o módulo `mod_status` carregado).
- `configtest` - Verifica se a sintaxe dos arquivos de configuração está OK (executa um `apache -t`).

Opções de linha de comando

- `-D nome` - define um nome que será usado na diretiva `<IfDefine nome>`.
- `-d diretório` - especifica o diretório `ServerRoot` (substitui o do arquivo de configuração).
- `-f arquivo` - especifica um arquivo `ServerConfigFile` alternativo.
- `-C "diretiva"` - processa a diretiva antes de ler os arquivos de configuração.
- `-c "diretiva"` - processa a diretiva depois de ler os arquivos de configuração.
- `-v` - mostra a versão do programa.
- `-V` - mostra opções usadas na compilação do **Apache**.
- `-h` - Mostra o help on-line do programa
- `-l` - lista módulos compilados junto com o Apache (embutidos)
- `-L` - lista diretivas de configurações disponíveis
- `-S` - Mostra configurações de Virtual Hosting
- `-t` - executa a checagem de sintaxe nos arquivos de configuração do Apache (incluindo a checagem da diretiva `DocRoot`).
- `-T` - executa a checagem de sintaxe nos arquivos de configuração do **Apache** (menos da diretiva `DocRoot`).

Configurações Básicas do Apache

Esta seção traz algumas configurações obrigatórias para quem está iniciando a configuração do servidor web Apache.

Configurando a porta padrao

Use a diretiva `Port` para configurar a porta padrão que o **Apache** receberá requisições por padrão. A diretiva `Listen` também é usada para ajustar o endereço/portas alternativas (usadas também em `Virtual Hosts`) e substituirá as definições de `Port`(veja “Especificando endereços/portas adicionais (a diretiva `Listen`)” para detalhes).

OBS:: Somente uma diretiva *Port* e um argumento poderão ser especificados. Para mais controle sobre as portas do sistema use a diretiva *Listen*.

Adicionando uma página no Apache

Existem dois tipos de páginas que podem ser adicionadas ao **Apache**: a página raiz e sub-páginas.

Página Raíz A página raiz é especificada através da diretiva *DocumentRoot* e será mostrada quando se entrar no domínio principal, como `http://www.guiafoca.org`. Na configuração padrão do **Apache**, *DocumentRoot* aponta para o diretório `/var/www`. Este diretório será assumido como **raiz** caso os diretórios não sejam iniciados por uma `/`:

- `home/focalinux` - Aponta para `/var/www/home/focalinux`
- `/home/focalinux` - Aponta para `/home/focalinux`

Este diretório deve conter um arquivo de índice válido (especificado pela diretiva *DocumentIndex* no `srm.conf`) e permissões de acesso válidas no arquivo `access.conf` para autorizar o acesso as páginas em `/var/www` (veja “Restrições de Acesso” para detalhes).

Sub-páginas Sub páginas são armazenadas abaixo do diretório da *Página raiz*, como `http://www.guiafoca.org/download`. Elas podem ser um subdiretório da página principal em `/var/www` ou serem criadas através da diretiva *Alias* no arquivo `srm.conf`. Caso seja um sub-diretório, as permissões de acesso de `/var/www` serão herdadas para este subdiretório, mas também poderão ser modificadas com a especificação de uma nova diretiva de acesso.

Através da diretiva *Alias* a página pode estar localizada em outro diretório do disco (até mesmo outro sistema de arquivos) e as permissões de acesso deverão ser definidas para aquela página. Para criar um endereço `http://www.guiafoca.org/iniciante` que aponta para o diretório `/home/focalinux/download/iniciante` no disco local, basta usar a seguinte diretiva no `srm.conf`:

```
Alias /iniciante /home/focalinux/download/iniciante
```

Pode ser necessário permitir o acesso a nova página caso o servidor tenha uma configuração restritiva por padrão (veja “Restrições de Acesso” para detalhes). Após isto, faça o servidor **httpd** re-ler os arquivos de configuração ou reinicia-lo. Após isto, a página `/home/focalinux/download/iniciante` estará acessível via `http://www.guiafoca.org/iniciante`.

OBS: Caso inclua uma `/` no diretório que será acessível via URL, o endereço somente estará disponível caso você entre com `/` no final da URL:

```
Alias /doc/ /usr/doc/
```

O diretório `/doc` somente poderá ser acessado usando `http://www.guiafoca.org/doc/`, o uso de `http://www.guiafoca.org/doc` retornará uma mensagem de URL não encontrada.

Configurando as interfaces que o Apache atenderá

A diretiva *BindAddress* é usada para especificar endereços IP das interfaces ou endereços FQDN que o **Apache** responderá requisições. Mais de um endereço podem ser especificados separados por espaços. Caso não seja definido, o **Apache** assumirá o valor "*" (atenderá requisições vindas de qualquer interface).

OBS1: - É permitido usar somente uma diretiva *BindAddress*. A diretiva *Listen* deverá ser usada se desejar mais controle sobre as portas do servidor web. Veja “Especificando endereços/portas adicionais (a diretiva *Listen*)” para detalhes.

OBS2: - As interfaces especificadas pela diretiva *Listen* substituirá as especificadas em *BindAddress*.

Exemplo:

- `BindAddress 192.168.1.1` - Especifica que os usuários da faixa de rede `192.168.1.*` terão acesso ao servidor **httpd**. Isto assume que a máquina possui o endereço `192.168.1.1` em sua interface de rede interna.
- `BindAddress *` - Atenderá requisições vindas de qualquer interface de rede.

Especificando endereços/portas adicionais (a diretiva *Listen*)

A diretiva *Listen* é usada para se ter um controle maior sobre a especificação de endereços/portas alternativas que o servidor web esperará por requisições externas. Esta diretiva é muito usada na construção de *Virtual Hosts*. Esta diretiva pode substituir completamente as diretivas *Port* e *BindAddress*. Podem ser usados o número da porta, ou o par endereço:porta:

```
Listen 192.168.1.1:80
Listen 192.168.7.1:81
Listen 60000
```

O endereço que deverá ser usado é o da interface de rede (assim como na diretiva *BindAddress*). No exemplo acima, o servidor **httpd** esperará por requisições vindas de `192.168.1.*` na porta 80 e também 60000, e requisições vindas de `192.168.7.1` na porta 81 e também 60000.

Restrições de Acesso

A restrição de acesso do **Apache** é feita através de *Autorização* (“Autorização”) e *Autenticação* (“Autenticação”). Através da *autorização*, é checado se o endereço/rede especificada tem ou não permissão para acessar a página. A *autenticação* requer que seja passado nome e senha para garantir acesso a página. Os métodos de *Autorização* e *Autenticação* podem ser combinados como veremos mais adiante.

Especificando opções/permissões para as páginas

As opções de restrição podem tanto ser especificadas nas diretivas `<Directory>`, `<Location>` ou `<Files>` quanto nos arquivos `.htaccess` (ou outro nome de arquivo de controle de acesso especificado pela opção *AccessFileName* do arquivo de configuração do **Apache**). Cada diretiva de acesso é especificada entre `<tags>` e devem ser fechadas com `</tag>` (como na linguagem HTML). As seguintes diretivas de acesso são válidas no **Apache**:

Directory As restrição afetará o diretório no disco especificado, conseqüentemente a página armazenada nele. Por exemplo:

```
<Directory /var/www>
  Order deny,allow
  deny from all
  allow from 10.1.0.1
</Directory>
```

O acesso ao diretório `/var/www` será permitido somente ao computador com o endereço IP `10.1.0.1`.

DirectoryMatch Funciona como a diretiva `<Directory>` mas trabalha com expressões regulares como argumento. Por exemplo:

```
<DirectoryMatch "^/www/.*">
  Order deny,allow
  deny from all
</DirectoryMatch>
```

Bloqueará o acesso ao diretório `/www` e sub-diretórios dentro dele.

Files As restrições afetarão os arquivos do disco que conferem com o especificado. É possível usar os coringas `?` e `*` como no shell. Também podem ser usadas expressões regulares especificando um `"~"` após `Files` e antes da expressão. Por exemplo:

```
<Files *.txt>
  Order deny,allow
  deny from all
</Files>
```

Bloqueia o acesso a todos os arquivos com a extensão `.txt`

```
<Files ~ "\.(gif|jpe?g|bmp|png)$">
  Order deny,allow
</Files>
```

Bloqueia o acesso a arquivos `gif`, `jpg`, `jpeg`, `bmp`, `png` (note que o `"~"` ativa o modo de interpretação de expressões regulares).

FilesMatch Permite usar expressões regulares na especificação de arquivos (equivalente a diretiva `<Files ~ "expressão">`). Por exemplo:

```
<FilesMatch "\.(gif|jpe?g|bmp|png)$">
  Order deny,allow
</FilesMatch>
```

Bloqueia o acesso a arquivos gif, jpg, jpeg, bmp, png.

Location As restrições afetarão o diretório base especificado na URL e seus sub-diretórios. Por exemplo:

```
<Location /security>
  Order allow,deny
</Location>
```

Bloqueia o acesso de todos os usuários ao diretório /security da URL (a explicação porque o acesso é bloqueado neste caso será explicado em “Autorização”).

LocationMatch Idêntico a diretiva <Location> mas trabalha com expressões regulares. Por exemplo:

```
<LocationMatch "/(extra|special)/data">
  Order deny,allow
  deny from all
</LocationMatch>
```

Bloqueará URLs que contém a substring "/extra/data" ou "/special/data".

O uso das diretivas <Directory> e <Files> é apropriada quando você deseja trabalhar com permissões a nível de diretórios/arquivos no disco local (o controle do proxy também é feito via <Directory>), o uso da diretiva <Location> é adequado para trabalhar com permissões a nível de URL. A ordem de processamento das diretivas de acesso são processadas é a seguinte:

1. A diretiva <Directory> (com exceção de <DirectoryMatch>) e os arquivos .htaccess são processados simultaneamente. As definições dos arquivos .htaccess substituem as de <Directory>)
2. Expressões regulares de <DirectoryMatch>, <Directory>.
3. <Files> e <FilesMatch> são processados simultaneamente.
4. <Location> e <LocationMatch> são processados simultaneamente.

Normalmente é encontrado a opção *Options* dentro de uma das diretivas acima, a função desta diretiva é controlar os seguintes aspectos da listagem de diretórios:

All	Todas as opções são usadas exceto a MultiViews. É a padrão caso a opção <i>Options</i> não seja especificada.
ExecCGI	Permite a execução de scripts CGI.
FollowSymLinks	O servidor seguirá links simbólicos neste diretório (o caminho não é modificado). Esta opção é ignorada caso apareça dentro das diretivas <Location>, <LocationMatch> e <DirectoryMatch>.
Includes	É permitido o uso de includes no lado do servidor.

IncludesNOEXEC	É permitido o uso de includes do lado do servidor, mas o comando #exec e #include de um script CGI são desativados.
Indexes	Se não existir um arquivo especificado pela diretiva <DirectoryIndex> no diretório especificado, o servidor formatará automaticamente a listagem ao invés de gerar uma resposta de acesso negado.
MultiViews	Permite o uso da Negociação de conteúdo naquele diretório. A negociação de conteúdo permite o envio de um documento no idioma requisitado pelo navegador do cliente.
SymLinksIfOwnerMatch	O servidor somente seguirá links simbólicos se o arquivo ou diretório alvo tiver como dono o mesmo user ID do link. Esta opção é ignorada caso apareça dentro das diretivas <Location>, <LocationMatch> e <DirectoryMatch>.

Múltiplos parâmetros para *Options* podem ser especificados através de espaços.

OBS1: A opção Options não tem efeito dentro da diretiva FILES.

OBS2: Tanto faz usar maiúsculas quanto minúsculas nas diretivas de configuração, opções e parâmetros de configuração do **Apache**, a capitalização apenas ajuda a leitura e interpretação: SymLinksIfOwnerMatch (LinksSimbólicosSeDonoConferir).

As opções especificadas para o diretório afetam também seus sub-diretórios, a não ser que sejam especificadas opções separadas para o sub-diretório:

```
<Directory /var/www>
  Options Indexes FollowSymLinks
</Directory>
```

Ao acessar o diretório /var/www/focalinux, as permissões usadas serão de /var/www, ao menos que uma diretiva <Directory> ou <Location> seja especificada:

```
<Directory /var/www>
  Options Indexes FollowSymLinks
</Directory>

<Directory /var/www/focalinux>
  Options Includes
</Directory>
```

As opções e restrições de acesso de /var/www/focalinux serão EXATAMENTE as especificadas no bloco da diretiva <Directory /var/www/focalinux> e somente os *includes* serão permitidos. Para adicionar ou remover uma opção individual definidas por diretivas anteriores, podem ser usado os sinais "+" ou "-", por exemplo:

```
<Directory /var/www>
  Options Indexes FollowSymLinks
</Directory>
```

```
<Directory /var/www/focalinux>
  Options +Includes -Indexes
</Directory>
```

As opções *Indexes* e *FollowSymLinks* são definidas para o diretório `/var/www`, então as permissões do diretório `/var/www/focalinux` serão *FollowSymLinks* (do diretório `/web/docs`) e *Includes* (adicionada) e o parâmetro *Indexes* não terá efeito neste diretório.

É permitido fazer um aninhamento das diretivas `<Directory>` e `<Files>`:

```
<Directory /var/www>
  Order allow,deny
  allow from all

  <Files LEIAME-DONO.txt>
    Order deny,allow
    deny from all
  </Files>

</Directory>
```

Neste caso, somente os arquivos `LEIAME-DONO.txt` existentes no diretório `/var/www` e seus sub-diretórios serão bloqueados.

Se a diretiva `<Files>` for usada fora de uma estrutura `<Directory>`, ela terá efeito em todos os arquivos disponibilizados pelo servidor. Este é excelente método para proteger os arquivos de acesso, senhas e grupos, conforme será explicado mais adiante.

Qualquer outro tipo de aninhamento de diretivas resultará em um erro de configuração ao se tentar carregar/recarregar o **Apache**. Um exemplo de diretiva incorreta:

```
<Directory /var/www>
  Options Indexes FollowSymLinks

  <Directory /var/www/focalinux>
    Options +Includes -Indexes
  </Directory>

</Directory>
```

O correto é:

```
<Directory /var/www>
  Options Indexes FollowSymLinks
</Directory>

<Directory /var/www/focalinux>
  Options +Includes -Indexes
</Directory>
```

Espero que tenha observado o erro no exemplo acima.

OBS1: Você pode verificar se a configuração do apache está correta digitando `apache -t` como usuário root, se tudo estiver correto com suas configurações ele retornará a mensagem: "Syntax OK".

OBS2: Se *Options* não for especificado, o padrão será permitir tudo exceto *MultiViews*.

OBS3: Qualquer restrição afetará o diretório atual e todos os seus sub-diretórios! Defina permissões de sub-diretórios específicos separadamente caso precise de um nível de acesso diferente. Veja também a seção sobre arquivos *OverRide* (`.htaccess`) para detalhes sobre este tipo de arquivo.

OBS4: A diretiva de acesso "`<Directory />`" não afetará outros sistemas de arquivos montados dentro de seus subdiretórios. Caso uma diretiva de acesso padrão não seja especificada para outros sistemas de arquivos, o acesso será automaticamente negado.

Autorização

A restrição de acesso por autorização (controlado pelo módulo `mod_access`), permite ou não o acesso ao cliente de acordo com o endereço/rede especificada. As restrições afetam também os sub-diretórios do diretório alvo. Abaixo um exemplo de restrição de acesso que bloqueia o acesso de qualquer host que faz parte do domínio `.spammers.com.br` a URL `http://servidor/teste`:

```
<Location /teste>
  Option Indexes
  Order allow,deny
  allow from all
  deny from .spammers.com.br
</Location>
```

A opção `Option` foi explicada acima, seguem as explicações das outras diretivas:

Order Especifica em que ordem as opções de acesso *allow/deny* serão pesquisadas. Caso não seja especificada, o padrão será *deny/allow*. Note que a ordem de pesquisa de *allow* e *deny* é a inversa da especificada. A diretiva *Order* aceita os seguintes valores:

- `deny, allow` - Esta é a padrão, significa um servidor mais restritivo; a diretiva *allow* é processada primeiro e somente depois a diretiva *deny*. Caso nenhuma diretiva *allow* e *deny* forem especificadas ou não conferirem, **PERMITE TUDO** como padrão.
- `allow, deny` - Significa um servidor mais permissivo, a opção *deny* é processada primeiro e somente depois a opção *allow*. Caso nenhuma diretiva *allow* e *deny* for especificadas ou não conferirem, **BLOQUEIA TUDO** como padrão.
- `mutual-failure` - Somente permite o acesso se o usuário receber autorização através da opção *allow* e **NÃO** ser bloqueado pela opção *deny*, caso uma das checagens falhe, o acesso é imediatamente negado. É uma opção interessante quando você quer somente pessoas de um determinado endereço/rede acessando o seu sistema e não estejam em sua lista negra :-)

ATENÇÃO: É importante saber se a página será permissiva ou restritiva para escolher a ordem mais adequada ao seu caso, também leve em consideração a possibilidade do processamento cair na diretiva de acesso padrão, caso nem a diretiva *allow* e *deny* conferiram e estiver usando a ordem de acesso "allow,deny" ou "deny,allow". Um sistema mal configurado neste aspecto poderá trazer sérias conseqüências.

É comum em páginas permissivas se definir a seguinte configuração:

```
Order allow,deny
allow from all
```

O motivo é que em um grande site, se forem adicionadas mais restrições nesta página (devido a alguns domínios que tem usuários mal comportados, bloqueio de acesso a rede do concorrente, potenciais atacantes, etc...), estas deverão ser lidas antes da diretiva "allow from all" e podem passar despercebidas ao administrador e podem simplesmente não funcionar caso a opção *Order* não esteja ajustada corretamente (lembre-se, você é o administrador e a integridade do site depende de sua atenção na escolha da ordem correta das diretivas de acesso).

allow from Especifica o endereço que terá acesso ao recurso especificado. A diretiva *allow from* aceita os seguintes valores:

- `all` - O acesso é permitido a todos.
- um endereço de domínio completo (FQDN). Por exemplo `www.debian.org.br`.
- um endereço de domínio parcial. Qualquer computador que confira com o início ou fim terá o acesso permitido. Por exemplo, `.spammers.com.br`, `.debian.org`.
- um endereço IP completo, como `192.168.1.1`
- um endereço IP parcial como `192.168.1.`
- um par rede/máscara como `10.1.0.0/255.255.0.0` ou `10.1.0.0/16`, uma faixa de acesso a máquinas de uma mesma rede pode ser definida facilmente através deste método.

OBS1: É necessário reiniciar o **Apache** depois de qualquer modificação em seu arquivo de configuração (executando `apachectl restart`), ou recarregar os arquivos de configuração (`apachectl graceful`).

OBS2: Mais de um host pode ser especificado separando com um espaço:

```
allow from 192.168. .debian.org.br
```

Permitirá o acesso de qualquer máquina que o endereço IP confira com `192.168.*.*` e qualquer computador do domínio `debian.org.br`

OBS3: Regras baseadas em nomes simples de hosts (como `www`) não conferirão! Deverá ser usado o FQDN ou IP: `www.dominio.com.br`

OBS4: Caso *Order* não seja especificado, *deny,allow* será usado como padrão (ou seja, permitirá tudo como padrão).

deny from Especifica os endereços que **NÃO** terão acesso ao recurso especificado. As explicações referentes a esta diretiva de acesso são idêntica as de *allow from*.

É recomendável o uso de endereços IP ao invés de endereços DNS e um mecanismo anti-spoofing no firewall ou código de roteamento, pois ficará mais difícil um ataque baseado em DNS spoofing, aumentando consideravelmente a segurança de seu servidor web.

ATENÇÃO: Caso receba erros 403 (acesso negado) sem bloquear a URL nas diretivas de acesso, uma dos seguintes problemas pode ser a causa:

- O servidor Web não tem permissões para acessar/abrir o diretório da página. Certifique-se que o *dono* e *grupo* do processo **Apache** (especificado pela diretiva *User* e *Group*) possuem permissões de acesso àquele diretório.
- Quando quer fazer uma listagem de arquivos do diretório e não especifica a opção `Option Indexes` como opção de listagem.
- Quando não está usando `Option Indexes` para impedir a listagem de conteúdo do diretório e o não foi encontrado um arquivo de índice válido dentre os existentes na diretiva `DirectoryIndex` no diretório atual.

Abaixo alguns exemplos de permissões de acesso:

```
<Directory /var/www>
  Options SymLinksIfOwnerMatch Indexes MultiViews
  Order allow,deny
  allow from all
</Directory>
```

Permite o acesso a de qualquer usuário de qualquer lugar (allow from all), permite também a visualização da listagem formatada de arquivos caso nenhum arquivo especificado na diretiva `DirectoryIndex` seja encontrado (Indexes), permite negociação de conteúdo (`MultiViews`) e seguir links caso o dono do arquivo confira com o nome do link (`SymLinksIfOwnerMatch`).

```
<Directory /var/www>
  Options SymLinksIfOwnerMatch Indexes MultiViews
</Directory>
```

Tem o mesmo significado da diretiva acima por métodos diferentes; quando nenhuma opção `Order` é especificada, `deny,allow` é definido como padrão, e como nenhuma opção de acesso `allow/deny` foi especificada, o padrão "Order deny,allow" é usado e permite TUDO como padrão.

```
<Directory /var/www>
  Options Indexes
  Order deny,allow
  deny from all
</Directory>
```

Esta regra acima não tem muita lógica pois restringe o acesso de todos os usuários ao diretório `/var/www`, ao menos se esta for sua intenção...

```
<Location /focalinux>
  Options All
  Order allow,deny
  allow from all
</Location>
```

A regra acima permite o acesso a URL `http://www.servidor.org/focalinux` de qualquer host na Internet

```
<Files .htaccess>
  Order deny,allow
  deny from all
</Files>
```

Bloqueia o acesso a qualquer arquivo `.htaccess` do sistema

```
<Files ~ "leia-me-(arm|alpha|m68k|sparc|powerpc)\.txt">
  Order deny,allow
  deny from all
</Files>
```

Bloqueia o acesso a qualquer arquivo `leia-me-arm.txt`, `leia-me-alpha.txt`, `leia-me-m68k.txt`, `leia-me-sparc.txt` e `leia-me-powerpc.txt` fazendo uso de expressões regulares.

```
<Directory /var/www>
  Options Indexes
  Order mutual-failure
  allow from .dominio.com.br
  deny from lammer.dominio.com.br
</Directory>
```

A diretiva acima somente permite acesso ao diretório `/var/www` de máquinas pertencentes ao domínio `.dominio.com.br` desde que não seja `lammer.dominio.com.br`.

```
<Directory /var/www>
  Options Indexes MultiViews
  Order allow,deny
  deny from .com .com.br
  allow from all
</Directory>
```

Bloqueia o acesso ao diretório `/var/www` de computadores pertencentes aos domínios `.com` e `.com.br`.

```
<Directory /var/www>
  Options None
  Order deny,allow
  allow from 192.168.1. .guiafoca.org .debian.org
  deny from 200.200.123.
</Directory>
```

A regra acima permite o acesso de máquinas da rede `192.168.1.*`, do domínio `*.guiafoca.org` e `*.debian.org`, o acesso de máquinas da rede `200.200.123.*` é bloqueado (nada contra, peguei nesse número ao acaso :-).

Note que a máquina `192.168.4.10` terá acesso LIVRE a regra acima, pois não conferirá nem com *allow* nem com *deny*, então o processamento cairá na diretiva padrão de *deny,allow*, que neste caso permite o acesso caso nem *allow* e *deny* conferiram com o padrão.

```
<Directory /var/www>
  Options None
  Order allow,deny
  allow from 192.168.1. .cipsga.org.br .debian.org
  deny from 200.200.123.
</Directory>
```

A regra acima é idêntica a anterior somente com a mudança da opção *Order*. Bloqueia o acesso de máquinas da rede 200.200.123.* e permite o acesso de máquinas da rede 192.168.1.*, do domínio *.cipsga.org.br e *.debian.org.

Note que a máquina 192.168.4.10 terá acesso BLOQUEADO a regra acima, pois não conferirá nem com *allow* nem com *deny*, então o processamento cairá na diretiva padrão de *allow,deny* que neste caso bloqueia o acesso.

Autenticação

Através da *autenticação* (controlado pelo módulo `mod_auth`) é possível especificar um *nome* e *senha* para acesso ao recurso solicitado. As senhas são gravadas em formato criptografado usando *Crypto* ou *MD5* (conforme desejado). O arquivo de senhas pode ser centralizado ou especificado individualmente por usuário, diretório ou até mesmo por arquivo acessado.

Criando um arquivo de Senhas

O arquivo de senhas pode ser criado e mantido através do uso de 3 utilitários: **htpasswd**, **htdigest** e **dbmmanage**:

htpasswd

Este é usado para criar o arquivo de senhas. Para criar um banco de dados com o nome *senhas* para o usuário *convidado*, é usada a seguinte sintaxe:

```
htpasswd -c -m senhas convidado
```

Você será perguntado por uma senha para o usuário *convidado* e para redigita-la. A opção "-c" indica que deverá ser criado um arquivo, a opção "-m" indica a utilização de senhas criptografadas usando o algoritmo *MD5*, que garante maior segurança que o método *Crypto*. A senha pode ser especificada diretamente na linha de comando através da opção "-b" (isto é um ótimo recurso para utilização em shell scripts ou programas CGI de integração com o navegador).

```
htpasswd -b -d senhas chefe abcdef
```

No exemplo acima, uma senha de alta segurança será introduzida no banco de dados *senhas* tornando impossível o acesso a página do usuário :-)

Note que esta senha foi cadastrada usando o algoritmo de criptografia *Crypto* (opção -d). O algoritmo *SHA* também pode ser usado como alternativa, através da opção "-s". Para modificar a senha do usuário convidado, basta usar a mesma sintaxe (sem a opção "-c" que é usada para criar um novo arquivo):

```
htpasswd -m senhas convidado
```

ou

```
htpasswd -b -m senhas convidado nova_senha
```

Opcionalmente você pode especificar a opção "-d" para atualizar também o formato da senha para *Crypto*. Podem existir senhas de criptografias mistas (*SHA*, *Crypto*, *MD5*) no mesmo arquivo sem nenhum problema.

A mudança do formato de senhas é útil quando se deseja aumentar o nível de segurança oferecido por um melhor sistema ou para manter a compatibilidade com alguns scripts/programas que compartilhem o arquivo de senhas.

htdigest e dbmmanage

Estes são idênticos ao **htpasswd**, a diferença é que o **htdigest** permite criar/manter um arquivo de senhas usando a autenticação Digest, enquanto o **dbmmanage** permite manter o banco de dados de senhas em um arquivo DB, DBM, GDBM e NDBM, formatos conhecidos pelo Perl.

Autenticação através de usuários

Através deste método é possível especificar que usuários terão acesso ao recurso definido, usando senhas de acesso individuais criptografadas usando um dos utilitários da seção anterior. Para restringir o acesso ao endereço `http://servidor.org/teste`:

```
<Location /teste>
  AuthName "Acesso a página do Foca Linux"
  AuthType basic
  AuthUserFile /home/gleydson/SenhaUsuario
  # AuthGroupFile /home/users/SenhaGrupo
  Require valid-user
</Location>
```

Ao tentar acessar o endereço `http://servidor/teste`, será aberta uma janela no navegador com o título *Enter username for Acesso a página do Foca Linux at servidor.org*, a diretiva *Require valid-user* definem que o usuário e senha digitados devem existir no arquivo especificado por *AuthUserFile* para que o acesso seja garantido. Uma explicação de cada opção de acesso usado na autenticação:

AuthName	Será o nome que aparecerá na janela de autenticação do seu navegador indicando qual área restrita está solicitando senha (podem existir várias no servidor, bastando especificar várias diretivas de restrições).
AuthType	Especifica o método de que o nome e senha serão passados ao servidor. Este método de autenticação pode ser <i>Basic</i> ou <i>Digest</i> <ul style="list-style-type: none">• <i>Basic</i> - Utiliza a codificação <i>base64</i> para encodificação de nome e senha, enviando o resultado ao servidor. Este é um método muito usado e pouco seguro, pois qualquer sniffer instalado em um roteador pode capturar e descobrir facilmente seu nome e senha.• <i>Digest</i> - Transmite os dados de uma maneira que não pode ser facilmente decodificada, incluindo a codificação da área protegida (especificada pela diretiva <i>AuthName</i>) que possui a sequência de login/senha válida. A diferença deste método é que você precisará de arquivos de senhas diferentes para cada área protegida especificada por <i>AuthName</i> (também chamada de Realm).
AuthUserFile	É o arquivo gerado pelo utilitário htpasswd que contém a senha correspondente ao usuário

AuthGroupFile	É um arquivo texto que contém o nome do grupo, dois pontos (":") e o nome dos usuários que podem ter acesso ao recurso, separados por vírgulas. No exemplo acima ele se encontra comentado, mas a seguir encontrará exemplos que explicam em detalhes o funcionamento desta diretiva.
Require	<p>Especifica que usuários podem ter acesso ao diretório. Podem ser usadas uma das 3 sintaxes:</p> <ul style="list-style-type: none"> • <code>Require user usuário1 usuário2 usuário3</code> - Somente os usuários especificados são considerados válidos para ter acesso ao diretório. • <code>Require group grupo1 grupo2 grupo3</code> - Somente os usuários dos grupos especificados são considerados válidos para terem acesso ao diretório. Esta diretiva é útil quando deseja que somente alguns usuários de determinado grupo tenham acesso ao recurso (por exemplo, usuários do grupo admins). • <code>Require valid-user</code> - Qualquer usuário válido no banco de dados de senhas pode acessar o diretório. É bem útil quando as opções de acesso especificadas por <code>Require user</code> são muito longas. <p>A opção <code>Require</code> deve ser acompanhado das diretivas <code>AuthName</code>, <code>AuthType</code> e as diretivas <code>AuthUserFile</code> e <code>AuthGroupFile</code> para funcionar adequadamente.</p>

OBS: É necessário reiniciar o **Apache** depois de qualquer modificação em seu arquivo de configuração (`apachectl restart`), ou recarregar os arquivos de configuração (`apachectl graceful`). Note que o **apachectl** é somente um shell script para interação mais amigável com o servidor web **apache**, retornando mensagens indicando o sucesso/falha no comando ao invés de códigos de saída.

Alguns exemplos para melhor assimilação:

```
<Location /teste>
  AuthName "Acesso a página do Foca Linux"
  AuthType basic
  AuthUserFile /home/gleydson/SenhaUsuario
  Require user gleydson
</Location>
```

As explicações são idênticas a anterior, mas somente permite o acesso do usuário `gleydson` a URL `http://servidor.org/teste`, bloqueando o acesso de outros usuários contidos no arquivo `AuthUserFile`.

```
<Location /teste>
  AuthName "Acesso a página do Foca Linux"
  AuthType basic
  AuthUserFile /home/gleydson/SenhaUsuario
  Require user gleydson usuario1 usuario2
</Location>
```

```
<Location /teste>
  AuthName "Acesso a página do Foca Linux"
  AuthType basic
  AuthUserFile /home/gleydson/SenhaUsuario
  Require user gleydson
```

```
Require user usuario1
Require user usuario2
</Location>
```

As 2 especificações acima são equivalentes e permite o acesso aos usuários gleydson, usuario1 e usuario2 a página `http://servidor.org/teste`.

Autenticação usando grupos

Há casos onde existem usuários de um arquivo de senhas que devem ter acesso a um diretório e outros não, neste caso a diretiva *valid-user* não pode ser especificada (porque permitiria o acesso de todos os usuários do arquivo de senha ao diretório) e uma grande lista de usuários ficaria bastante complicada de ser gerenciada com vários usuários na diretiva *Require user*.

Quando existe esta situação, é recomendado o uso de grupos de usuários. Para fazer uso desse recurso, primeiro deverá ser criado um arquivo que armazenará o nome do *grupo* e dos usuários pertencente àquele grupo usando a seguinte sintaxe (vamos chamar este arquivo de *SenhaGrupo*):

```
admins: gleydson usuario2
usuarios: usuario1 usuario2 usuario3 gleydson
```

Agora adaptamos o exemplo anterior para que somente os usuários especificados no grupo *admins* do arquivo criado acima:

```
<Location /teste>
  AuthName "Acesso a página do Foca Linux"
  AuthType basic
  AuthUserFile /home/gleydson/SenhaUsuario
  AuthGroupFile /home/gleydson/SenhaGrupo
  Require group admins
</Location>
```

Agora somente os usuários pertencentes ao grupo *admins* (*gleydson* e *usuario2*) poderão ter acesso ao diretório `/teste`.

OBS1: Verifique se o servidor Web possui acesso a leitura no arquivo de senhas de usuários e grupos, caso contrário será retornado um código "500 - Internal Server Error". Este tipo de erro é caracterizado por tudo estar OK na sintaxe dos arquivos de configuração após checagem com "apache -t" e todas as diretivas de controle de acesso apontam para os diretórios e arquivos corretos.

OBS2:: Sempre use espaços para separar os nomes de usuários pertencentes a um grupo.

OBS3: NUNCA coloque os arquivos que contêm senhas e grupos em diretórios de acesso público onde usuários podem ter acesso via o servidor Web. Tais localizações são `/var/www`, `/home/"usuario"/public_html` e qualquer outro diretório de acesso público que defina em seu sistema.

É recomendável também ocultar estes arquivos através da diretiva `<Files>` evitando possíveis riscos de segurança com usuários acessando os arquivos de senha e grupo.

Na distribuição **Debian**, qualquer arquivo iniciando com `.ht*` será automaticamente ocultado pelo sistema, pois já existe uma diretiva `<Files ~ ".ht">`. Tal diretiva pode também ser especificada no arquivo de acesso `.htaccess`. Assim um arquivo `.htsenha` e `.htgroup` são bons nomes se estiver desejando ocultar dados de olhos curiosos...

Usando autorização e autenticação juntos

Os métodos de *autorização* e *autenticação* podem ser usados ao mesmo tempo dentro de qualquer uma das diretivas de controle de acesso. As diretivas de *autorização* são processadas primeiro (`mod_access`) e depois as diretivas de *autenticação* (`mod_auth`). Segue um exemplo:

```
<Directory /var/www>
  Options Indexes
  Order deny,allow
  allow from .dominiolocal.com.br
  deny from all
  AuthName "Acesso ao diretório do servidor Web"
  AuthType basic
  AuthUserFile /var/cache/apache/senhas
  Require valid-user
</Directory>
```

Para ter acesso ao diretório `/var/www`, primeiro o computador deve fazer parte do domínio `.dominiolocal.com.br`, assim ela passa pelo teste de autorização, depois disso será necessário fornecer o login e senha para acesso a página, digitando o login e senha corretos, o teste de autenticação será completado com sucesso e o acesso ao diretório `/var/www` autorizado.

```
<Directory /var/www>
  Options Indexes
  Order mutual-failure
  allow from .dominiolocal.com.br
  deny from lammer.dominiolocal.com.br
  AuthName "Acesso ao diretório do servidor Web"
  AuthType basic
  AuthUserFile /var/cache/apache/senhas
  AuthGroupFile /var/cache/apache/grupos
  Require group admins
</Directory>
```

No exemplo acima, é usado o método de autorização com a opção *Order mutual-failure* e o método de autenticação através de *grupos*. Primeiro é verificado se o usuário pertence ao domínio `.dominiolocal.com.br` e se ele não está acessando da máquina `lammer.dominiolocal.com.br`, neste caso ele passa pelo teste de autorização. Depois disso ele precisará fornecer o nome e senha válidos, com o login pertencente ao *AuthGroupFile*, passando pelo processo de autenticação e obtendo acesso ao diretório `/var/www`.

Acesso diferenciado em uma mesma diretiva

É interessante permitir usuários fazendo conexões de locais confiáveis terem acesso direto sem precisar fornecer nome e senha e de locais inseguros acessarem somente após comprovarem **quem** realmente são. Como é o caso de permitir usuários de uma rede privada terem acesso completo aos recursos e permitir o acesso externo ao mesmo recurso somente através de senha. Isto pode ser feito com o uso da diretiva *Satisfy* junto ao bloco de *autorização/autenticação*. Vamos tomar como base o exemplo anterior:

```
<Directory /var/www>
  Options Indexes
  Order mutual-failure
```

```
allow from .dominiolocal.com.br
deny from lammer.dominiolocal.com.br
AuthName "Acesso ao diretório do servidor Web"
AuthType basic
AuthUserFile /var/cache/apache/senhas
AuthGroupFile /var/cache/apache/grupos
Require group admins
Satisfy any
</Directory>
```

Note que o exemplo é o mesmo com a adição da diretiva *Satisfy any* no final do bloco do arquivo. Quando a opção *Satisfy* não é especificada, ela assumirá "all" como padrão, ou seja, o usuário deverá passar no teste de autorização e autenticação para ter acesso.

A diferença do exemplo acima em relação ao da seção anterior é se a máquina passar no teste de autorização ela já terá acesso garantido. Caso falhe no teste de autorização, ainda terá a chance de ter acesso a página passando na checagem de autenticação.

Isto garante acesso livre aos usuários do domínio `.dominiolocal.com.br`. Já os outros usuários, incluindo acessos vindos de `lammer.dominiolocal.com.br` que pode ser uma máquina com muito uso, poderá ter acesso ao recurso caso tenha fornecido um nome e senha válidos para passar pelo processo de autenticação. Tenha isto em mente... este tipo de problema é comum e depende mais de uma política de segurança e conduta interna, o sistema de segurança não pode fazer nada a não ser permitir acesso a um nome e senha válidos.

Tenha cuidado com o uso da opção *Satisfy* em diretivas que especificam somente o método de autenticação:

```
<Directory /var/www>
Options Indexes
AuthName "Acesso ao diretório do servidor Web"
AuthType basic
AuthUserFile /var/cache/apache/senhas
AuthGroupFile /var/cache/apache/grupos
Require group admins
Satisfy any
</Directory>
```

ATENÇÃO PARA O DESCUIDO ACIMA!: Como o método de autorização **NÃO** é especificado, é assumido *deny,allow* como padrão, que permite o acesso a **TODOS** os usuários. O bloco acima **NUNCA** executará o método de autenticação por este motivo. A melhor coisa é **NÃO** usar a opção *Satisfy* em casos que só requerem autenticação ou usar *Satisfy all* (que terá o mesmo efeito de não usa-la, hehehe).

A falta de atenção nisto pode comprometer silenciosamente a segurança de seu sistema.

O arquivo `.htaccess`

O arquivo `.htaccess` deve ser colocado no diretório da página que deverá ter suas permissões de acesso/listagem controladas. A vantagem em relação a inclusão direta de diretivas de acesso dentro do arquivo de configuração do **Apache**, é que o controle de acesso poderá ser definido pelo próprio webmaster da página, sem precisar ter acesso direto a configuração do **Apache**, que requerem privilégios de root.

Outro ponto fundamental é que não há necessidade de reiniciar o servidor Web, pois este arquivo é lido no momento de cada acesso ao diretório que controla. O nome do arquivo `Override` pode ser definido através da diretiva *AccessFileName* no arquivo de configuração do **Apache**, `.htaccess` é usado como padrão.

O controle de que opções estarão disponíveis no `.htaccess` são definidas na diretiva *AllowOverride* que pode conter o seguintes parâmetros:

- `None` - O servidor não buscará o arquivo `.htaccess` nos diretórios
- `All` - O servidor utilizará todas as opções abaixo no arquivo `.htaccess`
- `AuthConfig` - Permite o uso de diretivas de autenticação (*AuthDBMGroupFile*, *AuthDBMUserFile*, *AuthGroupFile*, *AuthName*, *AuthType*, *AuthUserFile*, *Require*, etc.).
- `FileInfo` - Permite o uso de diretivas controlando o tipo de documento (*AddEncoding*, *AddLanguage*, *AddType*, *DefaultType*, *ErrorDocument*, *LanguagePriority*, etc.).
- `Indexes` - Permite o uso de diretivas controlando a indexação de diretório (*AddDescription*, *AddIcon*, *AddIconByEncoding*, *AddIconByType*, *DefaultIcon*, *DirectoryIndex*, *FancyIndexing*, *HeaderName*, *IndexIgnore*, *IndexOptions*, *ReadmeName*, etc.).
- `Limit` - Permite o uso de diretivas controlando o acesso ao computador (*allow*, *deny* e *order*).
- `Options` - Permite o uso de diretivas controlando características específicas do diretório (*Options* e *XBitHack*).

OBS: Não tem sentido usar a opção *AllowOverride* dentro da diretiva `<Location>`, ela será simplesmente ignorada.

Para acesso ao arquivo `.htaccess` do diretório `/var/www/focalinux`, o **Apache** buscará os arquivos `.htaccess` na sequência: `/ .htaccess`, `/var/ .htaccess`, `/var/www/ .htaccess`, `/var/www/focalinux/ .htaccess`, qualquer diretiva que não exista no `.htaccess` do diretório `/var/www/focalinux` terá seu valor definido pela diretiva dos arquivos `.htaccess` dos diretórios anteriores. Somente após esta sequência de checagens o acesso ao documento é permitido (ou negado).

Por este motivo, muitos administradores decidem desativar completamente o uso de arquivos `.htaccess` no diretório raiz e habilitar somente nos diretórios especificados pela diretiva `<Directory>` no arquivo de configuração do **Apache**, evitando brechas de segurança na manipulação destes arquivos (esta é uma boa idéia a não ser que se dedique 24 horas somente na administração do seu servidor Web e conheça toda sua estrutura hierárquica de segurança:

```
<Directory />
  AllowOverride none
</Directory>
```

```
<Directory /var/www>
  AllowOverride limit authconfig indexes
</Directory>
```

Na especificação acima, o arquivo `.htaccess` será procurado no diretório `/var/www` e seus sub-diretórios, usando somente opções que controlam a autorização de acesso (*limit*), autenticação e opções (*authconfig*) e de indexação de documentos (*indexes*).

Alguns exemplos do uso do arquivo `.htaccess`:

Para permitir o acesso direto de usuários da rede `192.168.1.*` diretamente, e requerer senha de acesso para outros usuários, o seguinte arquivo `.htaccess` deve ser criado no diretório `/var/www`:

```
Order deny,allow
allow from 192.168.1.0/24
deny from all
AuthName "Acesso a página Web principal da Empresa"
AuthType basic
AuthUserFile /var/cache/apache/senhas
Require valid-user
Satisfy any
```

Note que a sintaxe é exatamente a mesma das usadas na diretivas de acesso, por este motivo vou dispensar explicações detalhadas a respeito.

ATENÇÃO: A diretiva *Options Indexes* deverá ser especificada no *AllowOverride* e não no arquivo *.htaccess*. Agora você já sabe o que fazer se estiver recebendo erros 500 ao tentar acessar a página (Erro interno no servidor)...

Usando a diretiva **SetEnvIf** com **Allow** e **Deny**

É possível especificar o acesso baseado em variáveis de ambiente usando a diretiva *SetEnvIf*, isto lhe permite controlar o acesso de acordo com o conteúdo de cabeçalhos HTTP. A sintaxe é a seguinte:

```
SetEnvIf [atributo] [expressão] [variável]
```

Isto poder ser facilmente interpretado como: Se o "atributo" especificado conter a "expressão", a "variável" será criada e armazenará o valor verdadeiro. Veja abaixo:

```
SetEnvIf User-Agent ".*MSIE*." EXPLODER
<Directory /var/www>
  Order deny,allow
  allow from all
  deny from env=EXPLODER
</Directory>
```

Se o Navegador (campo *User-Agent* do cabeçalho http) usado para acessar a página for o Internet Explorer, a variável *EXPLODER* será criada e terá o valor verdadeiro (porque a expressão de *SetEnvIf* conferiu com a expressão).

Note o uso de "deny from env=VARIÁVEL". Neste caso se o navegador for o Internet Explorer, o acesso será bloqueado (pois o navegador conferiu, assim a variável *EXPLODER* recebeu o valor verdadeiro).

É permitido especificar as diretivas de acesso normais junto com especificação de variáveis de ambiente, basta separa-los com espaços. Uma descrição completa dos cabeçalhos HTTP, conteúdo e parâmetros aceitos por cada um são descritos na RFC 2068.

A diretiva **<Limit>**

Esta diretiva é semelhante a *<Directory>* mas trabalha com métodos HTTP (como GET, PUT, POST, etc) ao invés de diretórios. A diretiva *<Limit>* pode ser usada dentro da diretiva de acesso *<Directory>*, *<Location>*, mas nenhuma diretiva de controle de acesso pode ser colocada dentro de *<Limit>*.

Os métodos HTTP válidos são: GET, POST, PUT DELETE, CONNECT, OPTIONS, TRACE, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK e UNLOCK. Note que os métodos são case-sensitive. Por exemplo:

```
<Directory /var/www>
  Option Indexes
  <Limit POST PUT DELETE>
    Order deny,allow
    allow from 192.168.1.0/24
    deny from all
  </Limit>
</Directory>
```

Somente permitem o uso dos métodos POST, PUT, DELETE de máquinas da rede interna.

OBS1: Se o método GET é bloqueado, o cabeçalho HTTP também será bloqueado.

OBS2: A diretiva de acesso <Limit> somente terá efeito na diretiva <Location> se for especificada no arquivo de configuração do servidor web. A diretiva <Location> simplesmente é ignorada nos arquivos .htaccess...

Este abaixo é usado por padrão na distribuição **Debian** para restringir para somente leitura o acesso aos diretórios de usuários acessados via módulo mod_userdir:

```
<Directory /home/*/public_html>
  AllowOverride FileInfo AuthConfig Limit
  Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
  <Limit GET POST OPTIONS PROPFIND>
    Order allow,deny
    Allow from all
  </Limit>
  <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
    Order deny,allow
    Deny from all
  </Limit>
</Directory>
```

Diretiva <LimitExcept>

Esta diretiva é semelhante a <Limit>, mas atinge todos os métodos HTTP, menos os especificados.

Definindo documentos de erro personalizados

Documentos de erros personalizados são definidos através da diretiva *ErrorDocument*. É possível especificar códigos de erros que serão atendidos por certos documentos ou colocar esta diretiva dentro de blocos de controle de acesso <Directory>, <Location> ou <VirtualHost> para que tenham mensagens de erro personalizadas, ao invés da padrão usada pelo servidor **httpd**.

```
ErrorDocument [código de erro] [documento]
```

Onde:

código de erro Código de erro da mensagem (veja “Códigos HTTP” como referência). O código de erro *401* deve referir-se a um arquivo local.

documento Documento, mensagem de erro ou redirecionamento que será usado no servidor caso aquele código de erro seja encontrado:

Para definir uma mensagem de erro padrão para todo servidor web, basta colocar a diretiva *ErrorDocument* fora das diretivas que controlam o acesso a diretórios e virtual hosts (o início do arquivo `httpd.conf` é ideal).

Exemplos:

- `ErrorDocument 404 /cgi-bin/erros404.pl` - Direciona para um script em Perl que manda um e-mail ao administrador falando sobre o link quebrado e envia o usuário a uma página de erro padrão.
- `ErrorDocument 404 /naoencontrada.html` - Direciona o usuário para o arquivo `naoencontrada.html` (dentro de *DocumentRoot*) quando ocorrer o erro 404. Note que o diretório / levado em consideração é o especificado pela diretiva *DocumentRoot*.
- `ErrorDocument 500 "Erro Interno no servidor"` - Mostra a mensagem na tela quando ocorrer o erro 500.
- `ErrorDocument 401 /obtendoacesso.html` - Direciona o usuário ao arquivo explicando como obter acesso ao sistema.
- `ErrorDocument 503 http://www.guiafoca.org/servicos.html` - Redireciona o usuário a URL especificada.
- `ErrorDocument 403 "Acesso negado"` - Mostra a mensagem na tela no caso de erros 403.

Módulos DSO

Os módulos *DSO* permitem adicionar/remover características do **Apache** sem necessidade de recompilar todo o servidor web, assim interrompendo o serviço para a atualização dos arquivos. Módulos de programas terceiros também podem ser compilados e adicionado sem problemas através deste recurso.

Os módulos são carregados para a memória no momento que o **apache** é iniciado através da diretiva *LoadModule* no arquivo de configuração. Dessa forma, toda vez que um novo módulo for adicionado, removido ou alterado, será necessário reiniciar o servidor **apache**. A sintaxe da linha para carregar módulos `.so` é a seguinte:

```
LoadModule [nome_do_modulo] [caminho_do_arquivo_so]
```

nome_do_modulo Especifica o nome do módulo, não deve conter espaços.

caminho_do_arquivo_so Define a localização do arquivo que contém o módulo especificado. Por padrão os módulos estão localizados em `/usr/lib/apache/[versão]`

A posição em que os módulos aparecem podem ter influência em seu funcionamento, alguns requerem que sejam especificados antes de outros módulos para funcionarem corretamente (como o módulo *php3_module*, que deve ser carregado antes de qualquer módulo de controle de CGI's). Leia a documentação específica sobre o módulo em caso de dúvidas, os módulos que acompanham o **Apache** são documentados em detalhes no manual do **Apache**.

Para usar uma característica/diretiva/opção do **Apache** que dependa de um certo módulo, obviamente você deverá carregar o módulo correspondente (em caso de dúvidas, leia a documentação sobre o módulo). Veja a “`httpd.conf`” para exemplos do uso da diretiva *LoadModule*.

Por exemplo, se você quiser utilizar as diretivas de autorização (*allow*, *deny*, *order*) deverá ter o módulo *mod_access* carregado, para usar as diretivas de autorização (*authname*, *authuserfile*, *authtype*, etc) deverá

ter o módulo *mod_auth* carregado. Mais detalhes podem ser encontrados em “Autorização”. **OBS1:** O suporte a *DSO* atualmente só está disponível para plataforma **UNIX** e seus derivados, como o **Linux**.

Também é possível ativar certas diretivas verificando se o módulo correspondente estiver ou não carregado através da diretiva *IfModule*:

```
<IfModule mod_userdir.c>
  UserDir disabled root
  UserDir public_html
</IfModule>
```

Nas linhas acima, as diretivas *UserDir* somente serão executadas se o módulo *mod_userdir.c* estiver carregado através da diretiva *LoadModule*.

Segue abaixo uma lista de módulos padrões que acompanham do **Apache**, os módulos marcados com "*" são ativados por padrão:

- | | |
|---|---|
| Criação de Ambiente | <ul style="list-style-type: none"> • * <i>mod_env</i> - Ajusta variáveis de ambiente para scripts CGI/SSI • * <i>mod_setenvif</i> - Ajusta variáveis de ambiente de acordo com cabeçalhos http • <i>mod_unique_id</i> - Gera identificadores únicos para requisições |
| Decisão de tipo de conteúdo de arquivos | <ul style="list-style-type: none"> • * <i>mod_mime</i> - Determinação de tipo/encodificação do conteúdo (configurado) • <i>mod_mime_magic</i> - Determinação de tipo/encodificação do conteúdo (automático) • * <i>mod_negotiation</i> - Seleção de conteúdo baseado nos cabeçalhos "HTTP Accept" |
| Mapeamento de URL | <ul style="list-style-type: none"> • * <i>mod_alias</i> - Tradução e redirecionamento de URL simples • <i>mod_rewrite</i> - Tradução e redirecionamento de URL avançado • * <i>mod_userdir</i> - Seleção de diretórios de recursos por nome de usuário • <i>mod_speling</i> - Correção de URLs digitadas incorretamente • <i>mod_vhost_alias</i> - Suporte para virtual hosts dinamicamente configurados em massa. |
| Manipulação de Diretórios | <ul style="list-style-type: none"> • * <i>mod_dir</i> - Manipulação de Diretório e arquivo padrão de diretório • * <i>mod_autoindex</i> - Geração de índice automático de diretório |
| Controle de Acesso | <ul style="list-style-type: none"> • * <i>mod_access</i> - Controle de acesso por autorização (usuário, endereço, rede) • * <i>mod_auth</i> - Autenticação HTTP básica (usuário, senha) |

- `mod_auth_dbm` - Autenticação HTTP básica (através de arquivos NDBM do Unix)
- `mod_auth_db` - Autenticação HTTP básica (através de arquivos Berkeley-DB)
- `mod_auth_anon` - Autenticação HTTP básica para usuários no estilo anônimo
- `mod_auth_digest` - Autenticação MD5
- `mod_digest` - Autenticação HTTP Digest
- `mod_headers` - Cabeçalhos de respostas HTTP (configurado)
- `mod_cern_meta` - Cabeçalhos de respostas HTTP (arquivos no estilo CERN)
- `mod_expires` - Respostas de expiração HTTP
- * `mod_asis` - Respostas HTTP em formato simples (raw)
- * `mod_include` - Suporte a Includes no lado do servidor (SSI - Server Sides Includes)
- * `mod_cgi` - Suporte a CGI (Common Gateway Interface)
- * `mod_actions` - Mapeia scripts CGI para funcionarem como 'handlers' internos.
- * `mod_status` - Visualiza status do servidor em tempo de execução.
- `mod_info` - Visualiza sumário de configuração do servidor.
- * `mod_log_config` - Registro de requisições personalizáveis
- `mod_log_agent` - Registro especializado do User-Agent HTTP (depreciado)
- `mod_log_refer` - Registro especializado do Referrer HTTP (depreciado)
- `mod_usertrack` - Registro de cliques de usuários através de Cookies HTTP
- * `mod_imap` - Suporte a Mapeamento de Imagem no lado do servidor.
- `mod_proxy` - Módulo de Cache do Proxy (HTTP, HTTPS, FTP).
- `mod_so` - Inicialização do Dynamic Shared Object (DSO)
- `mod_mmap_static` - Cache de páginas frequentemente servidas via `mmap()`

Desenvolvimento

- `mod_example` - Demonstração da API do Apache (somente desenvolvedores)

Sistema de Log do Apache

O **Apache** é bem flexível na especificação do que será registrado em seus arquivos de log, possibilitando utilizar um arquivo de log único, diversos arquivos de logs registrando cada evento ocorrido no sistema (conexão, navegador, bloqueio de acesso, erros, etc) incluindo os campos que deseja em cada arquivo e a ordem dos campos em cada um deles.

Enfim qualquer coisa pode ser especificada de forma que atenda as suas necessidades particulares de logging.

AgentLog

`AgentLog arquivo/pipe`: Indica o nome do arquivo que registrará o nome do navegador que está acessando a página (conteúdo do cabeçalho `User-Agent`). É possível usar o pipe "|" para direcionar os erros para um programa de formatação ou processamento. **ATENÇÃO**: Se um programa for usado como pipe, ele será executado sob o usuário que iniciou o **apache**. Revise o código fonte do programa para ter certeza que não contém falhas que possam comprometer a segurança de seu sistema.

Exemplo: `AgentLog /var/log/apache/agent.log`

ErrorLog

`ErrorLog arquivo/pipe` - Especifica o arquivo que registrará as mensagens de erro do servidor **Apache**. É possível usar o pipe "|" para direcionar os erros para um programa de formatação ou processamento.

Exemplo: `ErrorLog /var/log/apache/errors.log`

CustomLog

Permite especificar onde os logs serão gravados para os arquivos de logs personalizados. Esta diretiva também aceita apelidos definidos pela diretiva `LogFormat`.

`CustomLog [arquivo/pipe] [formato/nome]`

Onde:

arquivo/pipe Arquivo de log personalizado ou pipe.

formato/nome Especifica o formato do arquivo de log (da mesma forma que o especificado na opção `LogFormat`). Deverá ser especificado entre "aspas" caso tiver espaços. Veja "LogFormat" para detalhes.

Ao invés de especificar o formato, também é possível usar um apelido definido pela opção `LogFormat` ("LogFormat"), neste caso os parâmetros definidos pelo `LogFormat` para "nome" serão atribuídos a diretiva `CustomLog`.

Exemplos:

- `CustomLog /var/log/apache/common.log "%h %l %u %t \"%r\" %>s %b"`

- CustomLog /var/log/apache/common.log common

RefererLog

RefererLog [arquivo/pipe]: Indica que arquivo/pipe registrará os campos Referer do cabeçalho HTTP. Esta diretiva é mantida por compatibilidade com o servidor web NCSA 1.4.

A configuração padrão do **Apache** usa uma diretiva alternativa para a especificação do *referer* que é a seguinte:

```
LogFormat "%{Referer}i -> %U" referer
CustomLog /var/log/apache/referer.log referer
```

Exemplo: RefererLog /var/log/apache/referer.log

RewriteLog

RewriteLog: [arquivo/pipe]: Indica o arquivo/pipe que registrará qualquer regravação de URL feita pelo **Apache**.

OBS: Não é recomendável direcionar o nome de arquivo para /dev/null como forma de desativar este log, porque o módulo de regravação não cria a saída para um arquivo de log, ele cria a saída de log internamente. Isto somente deixará o servidor lento. Para desativar este registro, simplesmente remova/comente a diretiva RewriteLog ou use a opção RewriteLogLevel 0.

Exemplo: RewriteLog "/usr/local/var/apache/logs/rewrite.log

RewriteLogLevel

RewriteLogLevel [num]: Especifica os detalhes que serão incluídos no registro da opção RewriteLog, os valores permitidos estão entre 0 e 9. Se for usado 0, o registro do RewriteLog é totalmente desativado (esta é a padrão). **OBS:** Qualquer valor acima de 2 deixa o servidor Web cada vez mais lento devido ao processamento e a quantidade de detalhes registrados no arquivo especificado por *RewriteLog*.

ScriptLog

ScriptLog [arquivo]: Especifica o nome do arquivo de log que receberá as mensagens de erros gerados por scripts CGI executados no servidor. Esta opção é controlada pelo módulos *mod_cgi*.

Os arquivos de log serão abertos por um sub-processo rodando com as permissões do usuário especificado na diretiva "user".

OBS: Esta opção somente é recomendada como depuradora de scripts CGI, não para uso contínuo em servidores ativos.

Exemplo: ScriptLog /var/log/apache/cgiscripts.log

ScriptLogBuffer

ScriptLogBuffer: Especifica o tamanho do cabeçalho PUT ou POST gravado no arquivo especificado por *ScriptLog*. O valor padrão é 1024 bytes. Esta opção é controlada pelo módulos *mod_cgi*

Exemplo: `ScriptLogBuffer 512`

ScriptLogLength

`ScriptLogLength: [tamanho]`: Especifica o tamanho máximo do arquivo de log gerado pela opção `ScriptLog`. O valor padrão é 10385760 bytes (10.3MB). Esta opção é controlada pelo módulos `mod_cgi`

Exemplo: `ScriptLogLength 1024480`

LogFormat

`LogFormat`: Define os campos padrões do arquivo gerado pela opção `TransferLog`. O seu formato é o seguinte:

`LogFormat [formato] [nome]`

Quando o formato não é especificado, assume o valor padrão `%h %l %u %t \"%r\" %s %b`. A especificação do `[nome]` permite que você utilize o formato especificado em uma opção `CustomLog` ou outra diretiva `LogFormat`, facilitando a especificação do formato do log.

Os seguintes formatos são válidos:

- `%b` - Bytes enviados, excluindo cabeçalhos HTTP.
- `%f` - Nome do arquivo.
- `%{FOOBAR}e` - O conteúdo da variável de ambiente FOOBAR.
- `%h` - Máquina cliente.
- `%a` - Endereço IP da máquina cliente.
- `%A` - Endereço IP local. Muito útil em virtual hostings.
- `%{Foobar}i` - O conteúdo de Foobar: linhas de cabeçalho na requisição enviada ao servidor.
- `%l` - O nome de login remoto enviado pelo identd (se fornecido).
- `%{Foobar}n` - O conteúdo de "FooBar" de outro módulo.
- `%{Foobar}o:` - O conteúdo de Foobar: linhas de cabeçalho na resposta.
- `%p` - A porta do servidor servindo a requisição.
- `%P` - A identificação do processo filho que serviu a requisição.
- `%r` - A primeira linha da requisição.
- `%s` - Status. Para requisições que foram redirecionadas internamente. Este é o status de uma requisição `*original*`. Use `%s` para a última.
- `%t` - Hora, no formato do arquivo de log (formato inglês padrão).
- `%{format}t` - Hora, no formato definido por `strftime`.
- `%T` - O tempo necessário para servir a requisição, em segundos.

- %u - Usuário remoto (através do auth, pode ser falso se o status de retorno (%s) for 401).
- %U - O caminho da URL requisitada.
- %v - O nome canônico definido por *ServerName* que serviu a requisição.
- %V - O nome do servidor de acordo com a configuração de *UseCanonicalName*.

Exemplos:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" fu
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %P %T" de
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

TransferLog

TransferLog [arquivo/pipe]: Indica o arquivo que armazenará as transferências entre o servidor http e o cliente. Ela cria o arquivo de log com o formato definido pela opção *LogFormat* mais recente (sem a especificação do nome associado a diretiva) ou o formato padrão CLF do log do **Apache**.

Se omitido, o arquivo não será gerado

Exemplo: `TransferLog /var/log/apache/transferências.log`

OBS: Se esta não é uma opção muito utilizada na administração de seus sistemas, é recomendável o uso da diretiva *CustomLog* (veja “CustomLog”) para evitar confusões futuras.

LogLevel

Define o nível de alerta das mensagens que serão gravadas no arquivo especificado pela diretiva *ErrorLog*. Quando não é especificado, assume o nível "error" como padrão. Abaixo os parâmetros aceitos em sua respectiva ordem de importância:

- `emerg` - O sistema está inutilizável.
- `alert` - A ação deve ser tomada imediatamente.
- `crit` - Condições críticas.
- `error` - Condições de erro.
- `warn` - Condições de alerta.
- `notice` - Condição normal mas significativa.
- `info` - Mensagens informativas.
- `debug` - Mensagens do nível de depuração.

Note que os níveis são os mesmos usados pelo **syslog**. Quando um nível particular é especificado, as mensagens de todos os níveis de maior importância também serão registrados. Por exemplo, se o nível

"info" for especificado, as mensagens com os níveis de "notice" e "warn" também serão registradas. É recomendado o uso de um nível de no mínimo *crit*.

Anonymous_LogEmail

Se estiver como "on" a senha digitada será registrada no arquivo especificado por *ErrorLog*. Esta diretiva é *ativada* por padrão.

Exemplo: `Anonymous_LogEmail off`

CookieLog

Especifica o arquivo que será usado para registrar os cookies

OBS1: Caso o caminho do arquivo não for especificado nas diretivas, será assumido *DocumentRoot* como diretório padrão.

OBS2: Caso esteja usando o pipe, o dono do processo será o mesmo que iniciou o servidor WEB Apache. Tenha certeza do funcionamento do programa para não comprometer o seu sistema, e cuide para que ele não possa ser modificado indevidamente por outros usuários.

Exemplo: `CookieLog /var/log/apache/cookies.log`

Relatório gráfico de acesso ao sistema

O programa **webalizer** poderá ser instalado para gerar um relatório gráfico com a estatísticas de visitas por ano/mes/dia/hora usando os dados do `access.log`. Outra interessante característica são as estatísticas de códigos http (veja "Códigos HTTP"), onde é possível saber a quantidade de links quebrados existentes em nosso servidor (estes poderão ser detectados usando o pacote de análise de sites). O **webalizer** também é compatível com os formatos de log do **squid** e **proftpd**. Na distribuição **Debian** ele pode ser instalado a partir do pacote e gera um relatório geral quando é executado sem opções.

Configurando o Apache como servidor proxy

O **Apache** pode ser configurado para funcionar como servidor proxy transparente para sua rede interna, possibilitando inclusive o uso de cache de disco. É possível se fazer conexões HTTP (incluindo SSL) e FTP. Através desta característica também é possível usar uma das características mais interessante desse servidor web: o redirecionamento de conexões para uma determinada URL para uma outra máquina, que pode ser um outro host remoto ou uma máquina da rede interna (não acessível diretamente via Internet).

O primeiro passo é ativar o módulo de proxy no arquivo `httpd.conf`, basta descomentar a linha:

```
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
```

O seguinte bloco pode ser colocado no final do arquivo `httpd.conf` para configurar um servidor proxy para realizar conexões diretas (sem o uso de cache) e permitir o uso de servidores proxy em sua rede:

```
# Suporte a Proxy
#
<IfModule mod_proxy.c>
    ProxyRequests off
```

```

ProxyRemote * http://debian:3128
ProxyBlock microsoft.com microsoft.com.br
NoProxy 192.168.1.0/24
ProxyDomain .gms.com.br

# Ativa/Desativa a manipulação de cabeçalhos HTTP/1.1 "Via:".
#
# ("Full" adiciona a versão do servidor Apache; "Block" remove todos os cabeçalhos
# de saída "Via:")
# Escolha uma das opções: Off | On | Full | Block
#
#ProxyVia On
#</IfModule>

```

Segue a explicação de cada uma das diretivas acima:

ProxyRequests [<i>on/off</i>]	Ativa (on) ou Desativa (off) o serviço de proxy do servidor Apache. Note que o módulo <code>libproxy.so</code> deve estar carregado para que o bloco <code><IfModule libproxy.c></code> seja processado. A desativação desta diretiva não afeta a diretiva <i>ProxyPass</i> .
ProxyRemote [<i>origem</i>] [<i>URL</i>]	Esta opção é útil para fazer o Apache redirecionar suas requisições para outro servidor proxy (como o squid ou o gateway da rede, caso o Apache estiver sendo executado em uma máquina interna). A <i>origem</i> pode ser uma URL completa (como <code>http://www.debian.org</code>), uma URL parcial (como <code>ftp</code> , <code>http</code>) ou <code>"*"</code> para que o redirecionamento seja sempre usado.
ProxyBlock [<i>padrão</i>]	Permite bloquear o acesso a endereços que contenham o <i>padrão</i> especificado. Podem ser especificadas palavras, máquinas, domínios, URLs separados por espaços. O Apache fará a resolução DNS no caso de endereços IP e fará o cache para requisições futuras.
NoProxy [<i>endereços</i>]	Permite especificar endereços Internos que não serão redirecionados para o servidor proxy especificado por <i>ProxyRemote</i> . Podem ser usados nomes de máquinas, endereços IP, subredes ou domínios separados por espaços.
ProxyDomain [<i>endereço</i>]	Especifica o endereço que será adicionado a URL caso seja recebida uma requisição que contenha somente um nome de máquina. É útil em redes Internas.

Note que quando o suporte a proxy não está ativado no **Apache**, qualquer endereço de URL externa levará a página definida pela diretiva *DocumentRoot*. Isto deixará de funcionar após configurar o serviço de proxy.

O uso do cache é interessante para acelerar as requisições http da rede interna para a rede externa, desta forma, se uma requisição foi feita anteriormente, será descarregado o arquivo do disco rígido e assim evitar uma nova conexão externa (isto libera a rede para outras coisas). Para configurar um cache no serviço proxy, adicione as seguintes linhas no final do bloco anterior de proxy:

```

# As linhas abaixo ativam o cache do apache, o cache não funcionará ao menos que
# CacheRoot seja especificado
CacheRoot /var/spool/apache
CacheForceCompletion 70

```

```
CacheSize 5
CacheGcInterval 3
CacheDefaultExpire 5
CacheMaxExpire 300
NoCache 192.168.1.0/24 a_domain.com outrodomínio.com.br outro.dominio.net
```

Cada diretiva acima possui o seguinte significado:

CacheRoot	Diretório base onde serão criados os outros diretórios de cache. O cache só será ativado se esta diretiva for definida.
CacheForceCompletion [num]	Se uma transferência for cancelada e passar de num%, o Apache continuará a transferência e armazenará o arquivo no cache. O valor padrão é 90.
CacheSize [num]	Define o tamanho máximo do diretório de cache do Apache , em KB. Não especifique um valor que tome mais de 70% do espaço em disco. O valor padrão é 5.
CacheGcInterval [num]	Define o tempo que o cache será checado em busca de arquivos maiores que o total do cache. Arquivos que ultrapassem o tamanho do cache são automaticamente eliminados.
CacheDefaultExpire [num]	Define o tempo que os documentos ficarão no cache, se foram transferidos através de protocolos que não suportam horas de expiração. O valor padrão é 1 hora.
CacheMaxExpire [num]	Define o tempo que os documentos permanecerão armazenados no cache (em horas). Esta opção ignora a hora de expiração do documento (caso fornecida). O valor padrão é 24 horas.
NoCache [endereços]	Permite especificar lista de palavras, máquinas, domínios, IP's que não serão armazenados no cache do Apache . Caso seja usado NoCache * o cache será desativado completamente. Note que o cache também pode ser desativado comentando a diretiva <i>CacheRoot</i> .

Se você desejar um servidor cache mais flexível, rápido, dinâmico, configurável (com possibilidade de uso de restrições baseadas em URL, tempo de acesso, autenticação), instale o **squid** e configure o **apache** para fazer forward de conexões para ele (“Redirecionamento de conexões no Apache”).

Controlando o acesso ao servidor proxy

Incluir o bloco abaixo no arquivo `access.conf` para definir o acesso dos serviços de proxy nas redes desejadas (se a sua configuração for aberta como padrão isto pode ser opcional):

```
# Acesso aos serviços proxy do apache
<Directory proxy:*>
    Order deny,allow
    Deny from all
    Allow from .seudominio.com.br
</Directory>
```

Para explicações sobre o processo de bloqueio acima, veja “Autorização”.

Redirecionamento de conexões no Apache

Este recurso do **Apache** é interessante para criar clusters de servidores em sua rede interna. O que ele faz é pegar uma requisição a um determinado endereço e redireciona-lo a outra máquina e as respostas são repassadas ao servidor web (para o cliente a mesma máquina esta atendendo a requisição, para você o processamento das requisições esta sendo distribuído internamente na rede).

As seguintes diretivas são usadas para realizar o redirecionamento de conexões: *ProxyPass* e *ProxyPassReverse*

`ProxyPass [diretório_da_url
[outro_servidor:/diretório]`

A *ProxyPass* permite que a URL seja redirecionada para o servidor local e diretório especificado. Por exemplo, assumindo que o endereço principal de nosso servidor é `http://www.guiafoca.org` e desejamos que a URL `http://www.guiafoca.org/download` seja atendida por uma máquina localizada na nossa rede privada com o endereço `http://192.168.1.54`. Basta incluir a linha:

```
ProxyPass /download http://192.168.1.54
```

Qualquer requisição externa a `http://www.guiafoca.org/download/iniciante` será atendida por `http://192.168.1.54/iniciante`.

`ProxyPassRemote
[diretório_da_url [outro_servidor:/
diretório]`

Esta diretiva permite modificar o cabeçalho `Location` nas mensagens de respostas de redirecionamento enviadas pelo **Apache**. Isto permite que o endereço retornado seja o do servidor (que faz a interface externa com o cliente) e não da máquina do redirecionamento.

```
ProxyPass /download http://192.168.1.54  
ProxyPassReverse /download http://192.168.1.54
```

Se a máquina `192.168.1.54` redirecionar a URL para `http://192.168.1.54/download/iniciante`, a resposta será modificada para `http://www.guiafoca.org/download/iniciante` antes de ser retornada ao cliente.

Virtual Hosts

Virtual Hosts (sites virtuais) é um recurso que permite servir mais de um site no mesmo servidor. Podem ser usadas diretivas específicas para o controle do site virtual, como nome do administrador, erros de acesso a página, controle de acesso e outros dados úteis para personalizar e gerenciar o site. Existem 2 métodos de virtual hosts:

- **Virtual Hosts baseados em IP** - Requer um endereço IP diferente para cada site. Este poderá ser um IP real (da interface de rede) ou um apelido (veja "IP Alias"), o que interessa é que deve haver um endereço IP diferente para cada site. O número de sites servidos estará limitado a quantidade de endereços IP disponíveis em sua classe de rede. Veja "Virtual hosts baseados em IP" para detalhes de como construir um virtual host deste tipo.

O **apache** foi um dos primeiros servidores web a incluir suporte a virtual hosts baseados em IP.

- Virtual Hosts baseados em nome - Este utiliza nomes para identificar os sites servidos e requerem somente um endereço IP. Desta maneira é possível servir um número ilimitado de sites virtuais. O navegador do cliente deve suportar os cabeçalhos necessários para garantir o funcionamento deste recurso (praticamente todos os navegadores atuais possuem este suporte). Veja “Virtual hosts baseados em nome” para detalhes de como construir um virtual host deste tipo.

As explicações desta seção são baseadas na documentação do **Apache**.

Virtual hosts baseados em IP

Existem duas maneiras de rodar este tipo de host virtual: Através de daemons **httpd** separados ou em um único daemon **httpd** usando a diretiva `<VirtualHost>`.

As vantagens do uso de *daemons separados* para servir requisições é a proteção sob *UID* e *GID* diferente dos outros servidores, assim o administrador do *site1* não terá acesso ao `httpd.conf`, página do *site2* (porque ele estará rodando sob uma *UID* e *GID* diferentes e o acesso é restrito). Para usar este método, especifique a opção `-f [arquivo_cfg]` para utilizar um arquivo de configuração personalizado e a diretiva `Listen endereço:porta` para dizer onde o servidor aguardará as requisições.

As vantagens do uso de um *mesmo daemon* para servir as requisições são: quando não há problema se os administradores de outros sites tenham acesso ao mesmo arquivo de configuração ou quando há a necessidade de servir muitas requisições de uma só vez (quanto menos servidores web estiverem em execução, melhor o desempenho do sistema). Abaixo um exemplo de configuração de virtual hosts servindo os sites `www.site1.com.br` e `www.site2.com.br`:

```
ServerAdmin webmaster@site.com.br
```

```
<VirtualHost www.site1.com.br>
  ServerName www.site1.com.br
  ServerAdmin site1@site1.com.br
  DocumentRoot /var/www/www_site1_com_br
  TransferLog /var/log/apache/site1/access.log
  ErrorLog /var/log/apache/site1/error.log
  User www-data
  Group www-data
</VirtualHost>
```

```
<VirtualHost www.site2.com.br>
  ServerName www.site2.com.br
  DocumentRoot /var/www/www_site2_com_br
  CustomLog /var/log/apache/site2/access.log combined
  ErrorLog /var/log/apache/site2/error.log
</VirtualHost>
```

Qualquer diretiva dentro de `<VirtualHost>` controlarão terão efeito no site virtual especificado. Quando uma diretiva não for especificada dentro de `<VirtualHost>`, serão usados os valores padrões especificados no arquivo de configuração do **Apache** (como a diretiva `ServerAdmin webmaster@site.com.br` que será usado como padrão na configuração de `www.site2.com.br`).

Digite `apache -S` para ver suas configurações de virtual hosts atual.

OBS1: Desative a diretiva `UseCanonicalName off` quando utilizar o recurso de máquinas virtuais, esta diretiva faz que o nome do servidor retornado usando o valor em `ServerName` quando o cliente digita um endereço qualquer.

OBS2: Utilize sempre que possível endereços IP em configurações críticas, assim os serviços não serão tão vulneráveis a possíveis falsificações ou erros. Veja “/etc/host.conf” e “Proteção contra IP spoofing”. Leia também a seção “Segurança no uso de IP’s em Virtual Hosts”.

OBS3: Não permita que outros usuários a não ser o root e o dono do processo **Apache** (especificado pela diretiva *User*) tenham acesso de gravação aos logs gerados pelo servidor, pois os dados podem ser apagados ou criados links simbólicos para binários do sistema que serão destruídos quando o **Apache** gravar dados. Alguns binários e bibliotecas são essenciais para o funcionamento do sistema.

Virtual hosts baseados em nome

Este método é idêntico ao baseado em IP, em especial adicionamos a diretiva *NameVirtualHost* para dizer qual é o endereço IP do servidor que está servindo os virtual hosts baseados em nome. Veja o exemplo de configuração:

```
NameVirtualHost 200.200.200.10:80

<VirtualHost _default_:80 200.200.200.10:80>
  ServerName www.site.com.br
  ServerAdmin admin@site.com.br
  DocumentRoot /var/www
  TransferLog /var/log/apache/access.log
  ErrorLog /var/log/apache/error.log
</VirtualHost>

<VirtualHost 200.200.200.10>
  ServerName www.site1.com.br
  ServerAdmin admin1@site1.com.br
  DocumentRoot /var/www/www_site1_com_br
  TransferLog /var/log/apache/site1/access.log
  ErrorLog /var/log/apache/site1/error.log
</VirtualHost>

<VirtualHost 200.200.200.10>
  ServerName www.site2.com.br
  ServerAdmin admin2@site2.com.br
  DocumentRoot /var/www/www_site2_com_br
  TransferLog /var/log/apache/site2/access.log
  ErrorLog /var/log/apache/site2/error.log
</VirtualHost>
```

A diretiva *NameVirtualHost* diz que será usado virtual hosts baseados em nome servidos pela máquina com IP 200.200.200.10. Os parâmetros dentro do bloco das diretivas <VirtualHost > são específicas somente no site virtual especificado, caso contrário os valores padrões definidos no arquivo de configuração serão usados. Caso nenhum virtual host confira com a configuração, o virtualhost *_default_* será usado.

Digite `apache -S` para ver suas configurações de virtual hosts atual. Se sua intenção é criar um grande número de virtual hosts que serão servidos pela mesma máquina, o uso da expansão %0 e diretivas *VirtualDocumentRoot* e *VirtualScriptAlias* são recomendados:

```
NameVirtualHost 200.200.200.10:80
```

```
<VirtualHost 200.200.200.10>
  VirtualDocumentRoot /var/www/%0
  VirtualScriptAlias /var/www/%0/cgi-bin
  TransferLog log/apache/sitel/access.log
  ErrorLog log/apache/sitel/error.log
</VirtualHost>
```

Agora crie os diretórios em `/var/www` correspondentes aos nomes de domínios que serão servidos por sua máquina: `mkdir /var/www/www.site1.com.br, mkdir /var/www/www.site2.com.br`. Note que sua máquina deverá estar com o DNS configurado para responder por estes domínios .

ATENÇÃO É importante que os endereços especificados nas diretivas *ServerName* (`www.site1.com.br`) resolvam o endereço IP da diretiva *VirtualHost* (`200.200.200.10`). Isto deve ser feito via DNS ou nos arquivos `/etc/hosts`.

OBS1: Utilize sempre que possível endereços IP em configurações críticas, assim os serviços não serão tão vulneráveis a possíveis falsificações ou erros. Veja “`/etc/host.conf`” e “Proteção contra IP spoofing”. Leia também a seção “Segurança no uso de IP's em Virtual Hosts”.

OBS2: Não permita que outros usuários a não ser o root e o dono do processo **Apache** (especificado pela diretiva *User*) tenha acesso de gravação aos logs gerados pelo servidor. Pois os dados podem ser apagados ou criados links para binários do sistema que serão destruídos quando o apache gravar dados para os logs. Alguns binários e bibliotecas são essenciais para o funcionamento do sistema.

Segurança no uso de IP's em Virtual Hosts

Quando você está colocando um nome na diretiva de configuração do seu virtual hosts, está assumindo que ele resolverá o endereço IP corretamente (como `www.site1.com.br => 200.200.200.10`). Se por algum motivo o servidor DNS for modificado (por outra pessoa que tem acesso a isto), o endereço IP resolvido para o site `www.site1.com.br` poderá ser modificado para `200.200.200.20`, isto redirecionará as requisições para outra máquina ao invés da máquina correta. Este tipo de ataque é chamado “DNS Spoofing” e o uso de endereço IP (ao invés de nomes) praticamente evita que isto aconteça. Esta situação pode acontecer com a diretiva abaixo:

```
<VirtualHost www.gms.com.br>
  ServerName www.gms.com.br
  ServerAdmin gleydson@guiafoca.org
  DocumentRoot /var/www/www_gms_com_br
</VirtualHost>
```

Outra situação, que impede o funcionamento do servidor Web, é quando o servidor DNS está em manutenção ou por algum outro motivo não pode resolver o endereço IP de um nome especificado (como `www.site1.com.br`). O **apache** precisa saber qual é o seu endereço IP para ser executado. Veja a próxima modificação:

```
<VirtualHost 192.168.1.1>
  ServerName www.gms.com.br
  ServerAdmin gleydson@guiafoca.org
  DocumentRoot /var/www/www_gms_com_br
</VirtualHost>
```

Na configuração acima usamos o IP do servidor para especificar o virtual host. O **apache** tentará fazer o DNS reverso para determinar qual nome é servido por aquele endereço IP (`www.site1.com.br`). Se

ele falhar, somente a seção `<VirtualHost>` correspondente será desativada. Isto já é uma melhoria sobre a primeira configuração. O nome do servidor na diretiva `ServerName` garante que o servidor responda com o nome correto.

Para evitar ataques baseados em DNS siga os seguintes procedimentos de segurança:

1. Preferencialmente utilize o arquivo `/etc/hosts` para a resolução de nomes em máquinas locais (principalmente quando existe somente um administrador). É um método que evita diversas consultas ao servidor DNS (que pode deixar o acesso lento) e este arquivo é gerenciado pelo usuário `root`, isto evita o acesso de qualquer usuário para a falsificação de endereços.

Este arquivo também é útil caso a pesquisa DNS falhe (quando a ordem de pesquisa for do servidor DNS para o arquivo `hosts` no arquivo `/etc/host.conf`), pois de qualquer forma o nome será resolvido e o servidor **Apache** será executado.

2. Evite dar poderes a outros administradores manipularem seu próprio domínio DNS, não há nada que possa impedi-lo de modificar o endereço "X" para ser servido pelo IP "Y" desviando o tráfego para seu próprio servidor web. Se isto não for possível, siga as dicas abaixo para diminuir possíveis problemas.
3. Utilize endereços IP na diretiva `<VirtualHost>`.
4. Use endereços IP na diretiva `Listen`.
5. Use um endereço IP na diretiva `BindAddress`.
6. Sempre utilize o parâmetro `ServerName` em todas as diretivas `<VirtualHost>`, isto evita o retorno incorreto de nomes (que pode evitar/revelar fraudes).
7. Quando utilizar virtual hosts, crie uma diretiva `<VirtualHost _default_L:*>` usando uma diretiva `DocumentRoot` que não aponte para lugar algum. Esta diretiva será acessada quando nenhuma diretiva `VirtualHost` servir a requisição, conferindo com o endereço/ip.

Uso de criptografia SSL

Esta seção é uma referência rápida para configuração e uso do módulo `apache-ssl` com o servidor **Apache**. Este módulo realiza a comunicação segura de dados (criptografada) via porta 443 (que é usada como padrão quando especificamos uma url iniciando com `https://`). A transmissão criptografada de dados é importante quanto temos dados confidenciais que precisamos transmitir como movimentação bancária, senhas, número de cartões de crédito, fazer a administração remota do servidor, etc. SSL significa *Secure Sockets Layer* (camada segura de transferência) e *TLS Transport Layer Security* (camada segura de Transporte).

A intenção aqui é fornecer explicações práticas para colocar um servidor **Apache** com suporte a SSL funcionando no menor tempo possível. Detalhes sobre funcionamento de certificados, métodos de criptografia, assinatura, etc. deverão ser buscados na documentação deste módulo ou em sites especializados (é um assunto muito longo).

Servidor apache com suporte a ssl

Ao invés de utilizar o módulo `mod_ssl`, você poderá usar o pacote `apache-ssl`, ele nada mais é que um servidor **Apache** com o suporte SSL já incluso e não interfere no servidor **Apache** padrão, porque é executado somente na porta 443.

Se você tem um grande site com configurações de acesso personalizadas, ele trará mais trabalho de administração, pois as configurações e diretivas de restrições de acesso deverão ser copiadas para este

servidor web. No entanto, ele é indicado para máquinas que serão servidores SSL dedicados ou quando não possui configurações especiais em seu servidor web principal.

Esta seção tem por objetivo a instalação do suporte ao módulo SSL (`mod_ssl`) no servidor **Apache** padrão.

Instalando o suporte a módulo SSL no Apache

Instale o pacote `ssl`. Após instala-lo, edite o arquivo `/etc/apache/httpd.conf` adicionando a linha:

```
LoadModule ssl_module /usr/lib/apache/1.3/mod_ssl.so
```

Depois, gere um certificado digital ssl com o programa **mod-ssl-makecert**. Ele será armazenado por padrão nos diretórios em `/etc/apache/ssl.???` e seu uso explicado no resto desta seção.

Gerando um certificado digital

O certificado digital é a peça que garante a transferência segura de dados. Ele contém detalhes sobre a empresa que fará seu uso e quem o emitiu. Para gerar ou modificar um certificado digital, execute o comando `mod-ssl-makecert` e siga as instruções. O método de criptografia usado pelo certificado digital é baseado no conceito de chave pública/privada, a descrição sobre o funcionamento deste sistema de criptografia é feito em “Usando `pgp` (`gpg`) para criptografia de arquivos”.

OBS Não utilize acentos nos dados de seu certificado.

Exemplo de configuração do módulo `mod-ssl`

Abaixo uma configuração rápida para quem deseja ter um servidor com suporte a SSL funcionando em menor tempo possível (ela é feita para operar em todas as instalações e não leva em consideração o projeto de segurança de sua configuração atual do **Apache**). Note que todas as diretivas relacionadas com o módulo `mod_ssl` começam com o nome "SSL":

```
# Somente processa as diretivas relacionadas a SSL caso o módulo mod_ssl estiver
# carregado pela diretiva LoadModule
<IfModule mod_ssl.c>
# É necessário especificar as portas que o servidor Web aguardará conexões (normai
# ssl).
Listen 80
Listen 443

# Ativa o tratamento de conexões com o destino na porta 443 pela diretiva
# VirtualHost abaixo
<VirtualHost _default_:443>

# Ativa ou desativa o módulo SSL para este host virtual
SSLEngine on

# Certificado do servidor
SSLCertificateFile /etc/apache/ssl.crt/server.crt

# Chave privada de certificado do servidor.
SSLCertificateKeyFile /etc/apache/ssl.key/server.key
```

```
# A linha abaixo força o fechamento de conexões quando a
# conexão com o navegador Internet Explorer é interrompida. Isto
# viola o padrão SSL/TLS mas é necessário para este tipo de
# navegador. Alguns problemas de conexões de navegadores também
# são causados por não saberem lidar com pacotes keepalive.
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown

</VirtualHost>

</IfModule>

#####
# Adicionalmente poderão ser especificadas as seguintes opções para modificar #
# o comportamento da seção SSL (veja mais detalhes na documentação do mod-ssl) #
#####

# Formato e localização do cache paralelo de processos da seção. O cache de seção
# feito internamente pelo módulo mas esta diretiva acelera o processamento
# de requisições paralelas feitas por modernos clientes navegadores. Por padrão
# nenhum cache é usado ("none").
SSLSessionCache          dbm:/var/run/ssl-cache

# Localização do arquivo de lock que o módulo SSL utiliza para
# sincronização entre processos. O padrão é nenhum.
SSLMutex file:/var/run/ssl-mutex

# Especifica o método de embaralhamento de dados que será utilizado
# durante o inicio de uma seção SSL (startup) ou durante o processo
# de conexão (connect). Podem ser especificados "builtin" (é muito rápido
# pois consome poucos ciclos da CPU mas não gera tanta combinação aleatória), um
# programa que gera números aleatórios (com "exec") ou os dispositivos aleatórios
# /dev/random e /dev/urandom (com "file"). Por padrão nenhuma fonte
# adicional de números aleatórios é usada.
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/urandom 512
#SSLRandomSeed connect exec:/pub/bin/NumAleat

# Tipos MIME para download de certificados
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

# Tempo máximo de permanência dos objetos do cache acima. O valor padrão é
# 300 segundos (5 minutos).
SSLSessionCacheTimeout 300

# Versão do protocolo SSL que será usada. Podem ser especificadas
# SSLv2, SSLv3 TLSv1 ou all. O mais compatível com os navegadores atuais
# é o "SSLv2". Por padrão "all" é usado.
#SSLProtocol all
#SSLProtocol -all +SSLv3
```

```
# Registra detalhes sobre o tráfego neste arquivo. Mensagens de erro
# também são armazenadas no arquivo de registro padrão do Apache
SSLLog      /var/log/apache/ssl-mod.log
```

```
# Nível das mensagens de log registradas por SSLLog
SSLLogLevel info
```

Algumas diretivas deste módulo podem fazer parte tanto da configuração global do servidor como diretivas de acesso (`Directory`, `Location`, `.htaccess`, veja a opção "Context" na documentação do `mod_ssl`).

Autorizando acesso somente a conexões SSL

Existem casos que precisa restringir o uso de conexões normais e permitir somente conexões via SSL (como por exemplo, dentro da diretiva de acesso que controla seu acesso a uma página com listagem de clientes). A opção `SSLRequireSSL` é usada para tal e deve ser usada dentro das diretivas de controle acesso:

```
<Directory /var/www/secure/clientes>
  Options Indexes
  Order deny,allow
  Deny from evil.cracker.com
  SSLRequireSSL
</Directory>
```

A diretiva acima *requer* que sejam feitas conexões SSL (porta 443 - `https://`) para acesso ao diretório `/var/www/secure/clientes`, qualquer conexão padrão não criptografada (feita na porta 80) será rejeitada com o erro 403.

OBS: A diretiva `SSLRequireSSL` podia ser colocada entre as condicionais "IfModule `mod_ssl.c`" mas o servidor web permitiria conexões não criptografadas se por algum motivo esse módulo não estivesse carregado. Na configuração acima, ocorrerá um erro e impedirá o funcionamento do servidor web caso ocorra algum problema com o `mod_ssl`.

Iniciando o servidor Web com suporte a SSL

Verifique se a configuração do **Apache** está ok com `apache -t`. Caso positivo, reinicie o servidor usando um dos métodos descritos em "Iniciando o servidor/reiniciando/recarregando a configuração". O servidor web lhe pedirá a FraseSenha para descriptografar a chave privada SSL (esta senha foi escolhida durante o processo de criação do certificado).

Esta senha garante uma segurança adicional caso a chave privada do servidor seja copiada de alguma forma. Somente quem tem conhecimento da FraseSenha poderá iniciar o servidor com suporte a transferência segura de dados. Verifique se o virtual host está servindo as requisições na porta 443 com `apache -S`.

O único método para fazer o servidor web evitar de pedir a senha para descriptografar a chave privada é colocando uma senha em branco. Isto só é recomendado em ambientes seguros e o diretório que contém a chave privada deverá ter somente permissões para o dono/grupo que executa o servidor Web. Qualquer outra permissão poderá por em risco a segurança da instalação caso a chave privada seja roubada. Depois disso, execute o comando:

```
# entre no diretório que contém a chave privada
```

```
cd /etc/apache/ssl.key
# renomeie a chave privada para outro nome
ren server.key server.key-Csenha
openssl rsa -in server.key-Csenha -out server.key
```

Digite a senha quando pedido. A chave original (com senha) estará gravada no arquivo `server.key-Csenha` e poderá ser restaurada se necessário. Reinicie o servidor **Apache**, desta vez ele não pedirá a senha.

OBS1: Tire uma cópia de segurança da chave privada original antes de executar esta operação.

OBS2: Não se esqueça de ajustar as permissões de acesso no diretório `/etc/apache/ssl.key` caso não utilize senha para proteger seu certificado digital.

Exemplo comentado de um arquivo de configuração do Apache

O exemplo abaixo foi retirado da distribuição **Debian GNU/Linux**, fiz sua tradução, modificações e incluí alguns comentários sobre as diretivas para deixá-lo mais de acordo com o conteúdo abordado pelo guia e mais auto-explicativo.

A configuração do **Apache** está distribuída nos arquivos `httpd.conf`, `srm.conf` e `access.conf` e podem ser usados como modelo para a construção da configuração de seu servidor.

httpd.conf

```
##
## httpd.conf -- Arquivo de configuração do servidor httpd Apache
##

#
# Baseado nos arquivos de configuração originais do servidor NCSA por Rob McCool.
# Modificado para distribuição junto ao guia Foca GNU/Linux Avançado
# http://focalinux.cipsga.org.br/ <gleydson@guiafoca.org>
#
# Este é o arquivo de configuração principal do servidor Apache. Ele contém as
# diretivas de configuração que dão ao servidor suas instruções.
# Veja <http://www.apache.org/docs/> para informações detalhadas sobre as
# diretivas.
#
# NÃO leia simplesmente as instruções deste arquivo sem entender o que significam
# e o que fazem, se não tiver certeza do que está fazendo consulte a documentação
# on-line ou leia as seções apropriadas do guia. Você foi avisado.
#
# Após este arquivo ser processado, o servidor procurará e processará o arquivo
# /etc/apache/srm.conf e então /etc/apache/access.conf
# a não ser que você tenha modificado o nome dos arquivos acima através das
# diretivas ResourceConfig e/ou AccessConfig neste arquivo.
#
# Configuração e nomes de arquivos de log: Se os nomes de arquivos que
# especificar para os arquivos de controle do servidor iniciam com uma
```

```
# "/", o servidor usará aquele caminho explicitamente. Se os nomes *não*
# iniciarem com uma "/", o valor de ServerRoot é adicionado -- assim
# "logs/foo.log" com ServerRoot ajustado para "/usr/local/apache" será
# interpretado pelo servidor como "/usr/local/apache/logs/foo.log".
#
# Originalmente por Rob McCool
# modificado por Gleydson Mazioli da Silva para o guia Foca GNU/Linux

# Carga dos Módulos de Objetos Compartilhados:
# Para você ser capaz de usa a funcionalidade de um módulo que foi construído como
# um módulo compartilhado, será necessário adicionar as linhas 'LoadModule'
# correspondente a sua localização, assim as diretivas que os módulos contém
# estarão disponíveis _antes_ de serem usadas.
# Exemplo:

#
# ServerType pode ser inetd, ou standalone. O modo Inetd somente é suportado nas
# plataformas Unix. O modo standalone inicia o servidor como um daemon.
#

ServerType standalone

# Se estiver executando a partir do inetd, vá até a diretiva "ServerAdmin".

# Port: A porta que o servidor standalone escutará. Para portas < 1023, será
# necessário o servidor funcionando como root inicialmente.

Port 80

#
# HostnameLookups: Registra os nomes DNS dos clientes ou apenas seus endereços
# IP's
# ex., www.apache.org (on) ou 204.62.129.132 (off).
# O valor padrão é off porque permitirá menos tráfego na rede. Ativando
# esta opção significa que cada acesso de um cliente resultará em
# NO MÍNIMO uma requisição de procura ao servidor de nomes (DNS).
#

HostnameLookups off

# Caso desejar que o servidor http seja executado como um usuário ou grupo diferen
# você deve executar o httpd inicialmente como root e ele modificará sua ID para a
# especificada.

# User/Group: O nome (ou #número) do usuário/grupo que executará o servidor httpd.
# No SCO (ODT 3) use "User nouser" e "Group nogroup"
# No HPUX você pode não será capaz de usar memória compartilhada como nobody, e
# é sugerido que seja criado um usuário www e executar o servidor httpd como
# este usuário, adequando as permissões onde necessárias.

User www-data
Group www-data

# ServerAdmin: Seu endereço de e-mail, onde os problemas com o servidor devem ser
```

```

# enviadas. Este endereço aparecerá nas mensagens de erro do servidor.
ServerAdmin gleydson@guiafoca.org

#
# ServerRoot: O topo da árvore de diretórios onde os arquivos de configuração do
# servidor, erros, e log são mantidos.
#
# NOTA: Se tiver a intenção de colocar isto em um sistema de arquivos montado
# em um servidor NFS (ou outra rede) então por favor leia a documentação do
# LockFile
# (disponível em <http://www.apache.org/docs/mod/core.html#lockfile>);
# e se salvará de vários problemas.
#
# Não adicione uma barra no fim do caminho do diretório.
#

ServerRoot /etc/apache

# BindAddress: Você pode usar esta opção em virtual hosts. Esta
# opção é usada para dizer ao servidor que endereço IP escutar. Ele pode
# conter ou "*", um endereço IP, ou um nome de domínio completamente qualificado
# (FQDN). Veja também a diretiva VirtualHost.

BindAddress *

#
# Suporte a Objetos Compartilhados Dinamicamente (DSO - Dynamic Shared Object)
#
# Para ser capaz de usar a funcionalidade de um módulo que foi compilado como
# um módulo DSO, você terá que adicionar as linhas 'LoadModule' correspondentes
# nesta localização, assim as diretivas contidas nela estarão disponíveis
# _antes_ de serem usadas. Por favor leia o arquivo README.DSO na distribuição
# 1.3 do Apache para mais detalhes sobre o mecanismo DSO e execute o comando
# "apache -l" para a lista de módulos já compilados (estaticamente linkados e
# assim sempre disponíveis) em seu binário do Apache.
#
# Please keep this LoadModule: line here, it is needed for installation.
# LoadModule vhost_alias_module /usr/lib/apache/1.3/mod_vhost_alias.so
# LoadModule env_module /usr/lib/apache/1.3/mod_env.so
LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config.so
# LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
LoadModule mime_module /usr/lib/apache/1.3/mod_mime.so
LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
# LoadModule info_module /usr/lib/apache/1.3/mod_info.so
# LoadModule includes_module /usr/lib/apache/1.3/mod_include.so
LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
LoadModule php3_module /usr/lib/apache/1.3/libphp3.so
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
# LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so
# LoadModule imap_module /usr/lib/apache/1.3/mod_imap.so
# LoadModule action_module /usr/lib/apache/1.3/mod_actions.so
# LoadModule speling_module /usr/lib/apache/1.3/mod_speling.so

```

```

LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
LoadModule auth_module /usr/lib/apache/1.3/mod_auth.so
# LoadModule anon_auth_module /usr/lib/apache/1.3/mod_auth_anon.so
# LoadModule dbm_auth_module /usr/lib/apache/1.3/mod_auth_dbm.so
# LoadModule db_auth_module /usr/lib/apache/1.3/mod_auth_db.so
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
# LoadModule digest_module /usr/lib/apache/1.3/mod_digest.so
# LoadModule cern_meta_module /usr/lib/apache/1.3/mod_cern_meta.so
LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
# LoadModule headers_module /usr/lib/apache/1.3/mod_headers.so
# LoadModule usertrack_module /usr/lib/apache/1.3/mod_usertrack.so
LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so
LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
# LoadModule sys_auth_module /usr/lib/apache/1.3/mod_auth_sys.so
# LoadModule put_module /usr/lib/apache/1.3/mod_put.so
# LoadModule throttle_module /usr/lib/apache/1.3/mod_throttle.so
# LoadModule allowdev_module /usr/lib/apache/1.3/mod_allowdev.so
# LoadModule auth_mysql_module /usr/lib/apache/1.3/mod_auth_mysql.so
# LoadModule pgsql_auth_module /usr/lib/apache/1.3/mod_auth_pgsql.so
# LoadModule eaccess_module /usr/lib/apache/1.3/mod_eaccess.so
# LoadModule roaming_module /usr/lib/apache/1.3/mod_roaming.so

#
# ExtendedStatus: Controla de o Apache gerará detalhes completos de status
# (ExtendedStatus On) ou apenas detalhes básicos (ExtendedStatus Off) quando o
# manipulador (handler) "server-status" for usado. O padrão é Off.
#
ExtendedStatus on

#
# ErrorLog: A localização do arquivo de log de erros.
# Se não estiver especificando a diretiva ErrorLog dentro de <VirtualHost>,
# as mensagens de erros relativas aos hosts virtuais serão registradas neste
# arquivo. Se definir um arquivo de log de erros para <VirtualHost>, as
# mensagens relativas ao servidor controlados por ela serão registradas lá e
# não neste arquivo.
#
ErrorLog /var/log/apache/error.log

#
# LogLevel: Controla o número de mensagens registradas no ErrorLog.
# Facilidades possíveis incluem: debug, info, notice, warn, error, crit,
# alert, emerg.
# Veja as facilidades na seção do guia sobre o syslog para detalhes
#
LogLevel warn

# As seguintes diretivas definem alguns formatos de nomes que serão usadas com a
# diretiva CustomLog (veja abaixo).

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" ful

```

```

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %P %T" deb
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

#
# A localização e formato do arquivo de log de acesso (definida pela diretiva
# LogFormat acima).
# Se não definir quaisquer arquivos de log de acesso dentro de um
# <VirtualHost>, elas serão registradas aqui. Se for definida dentro
# de <VirtualHost> o arquivo de log de acesso será registrado no
# arquivo especificado na diretiva e não aqui.
#
#CustomLog /var/log/apache/access.log common

# Se você desejar ter um arquivo de log separado para o agent (navegador usado)
# e referer, descomente as seguintes diretivas.

#CustomLog /var/log/apache/referer.log referer
#CustomLog /var/log/apache/agent.log agent

# Se preferir um arquivo de log simples, com os detalhes de acesso, agent, e
# referer (usando o formato combined da diretiva LogFile acima), use a seguinte
# diretiva.

CustomLog /var/log/apache/access.log combined

#
# Incluir uma linha contendo a versão do servidor e um nome de host virtual
# para as páginas geradas pelo servidor (documentos de erro, listagens
# de diretórios FTP, saída dos módulos mod_status e mod_info, etc., exceto
# para documentos gerados via CGI). Use o valor "EMail" para também incluir
# um link mailto: para o ServerAdmin. Escolha entre "On", "Off" ou "EMail".
#
ServerSignature On

#
# PidFile: O arquivo que o servidor gravará os detalhes sobre seu PID quando
# iniciar.
#
PidFile /var/run/apache.pid

#
# ScoreBoardFile: Arquivo usado para armazenar detalhes do processo interno do
# servidor. Nem todas as arquiteturas requerem esta diretiva, mas se a sua
# requerer (você saberá porque este arquivo será criado quando executar o
# Apache) então você *deverá* ter certeza que dois processos do Apache não
# utilizam o mesmo arquivo ScoreBoardFile.
#
ScoreBoardFile /var/run/apache.scoreboard

#
# Na configuração padrão, o servidor processará este arquivo, o

```

```
# srm.conf e o access.conf neste ordem. Você pode fazer o servidor
# ignorar estes arquivos usando "/dev/null".
#
ResourceConfig /etc/apache/srm.conf
AccessConfig /etc/apache/access.conf

#
# A diretiva LockFile define o caminho do lockfile usado quando o servidor
# Apache for compilado com a opção USE_FCNTL_SERIALIZED_ACCEPT ou
# USE_FLOCK_SERIALIZED_ACCEPT. Esta diretiva normalmente deve ser deixada em seu
# valor padrão. A razão principal de modifica-la é no caso do diretório de logs
# for montado via um servidor NFS< pois o arquivo especificado em LockFile
# DEVE SER ARMAZENADO EM UM DISCO LOCAL. O PID do processo do servidor principal
# é automaticamente adicionado neste arquivo.
#
LockFile /var/run/apache.lock

# ServerName permite ajustar o nome de host que será enviado
# aos clientes, caso for diferente do nome real (por exemplo, se desejar usar
# www ao invés do nome real de seu servidor).
#
# Nota: Você não pode simplesmente inventar nomes e esperar que funcionem. O nome
# que definir deverá ser um nome DNS válido para sua máquina.

ServerName debian.meudominio.org

# UseCanonicalName: Com esta opção ligada, se o Apache precisar construir uma
# URL de referência (uma url que é um retorno do servidor a uma requisição) ele
# usará ServerName e Port para fazer o "nome canônico". Com esta opção desligada,
# o Apache usará computador:porta que o cliente forneceu, quando possível.
# Isto também afeta SERVER_NAME e SERVER_PORT nos scripts CGIs.
#
# Dependendo de sua configuração, principalmente em virtual hosts, é recomendável
# deixa-la desativada ou com o valor DNS. O valor DNS obtém o nome do servidor
# através de uma requisição DNS reversa do endereço IP (muito útil para virtual
# hosts baseados em IP).
UseCanonicalName off

# CacheNegotiatedDocs: Por padrão, o Apache envia Pragma: no-cache com cada
# documento que foi negociado na base do conteúdo. Isto permite dizer a
# servidores proxy para não fazerem cache do documento. Descomentando a
# seguinte linha desativa esta característica, e os proxies serão capazes
# de fazer cache dos documentos.

#CacheNegotiatedDocs

# Timeout: O número de segundos antes de receber e enviar um time out

Timeout 300

# KeepAlive: Se vai permitir ou não conexões persistentes (mais que uma requisição
# por conexão). Mude para "Off" para desativar.
KeepAlive On
```

```
# MaxKeepAliveRequests: O número máximo de requisições que serão permitidas
# durante uma conexão persistente. Mude para 0 para permitir uma quantidade
# ilimitada. Nós recomendamos deixar este número alto, para obter a máxima
# performance

MaxKeepAliveRequests 100

# KeepAliveTimeout: Número de segundos que aguardará a próxima requisição

KeepAliveTimeout 15

# Regulagem do tamanho de pool do servidor. Ao invés de fazer você adivinhar
# quantos processos servidores precisará, o Apache adapta dinamicamente
# de acordo com a carga que ele vê --- isto é, ele tenta manter o número de
# processos o bastante para manipular a carga atual, mas alguns poucos
# servidores esparsos para manipular requisições transientes (ex. requisições
# simultâneas múltiplas de um navegador Netscape simples).

# Ele faz isto verificando periodicamente quantos servidores estão
# aguardando por uma requisição. Se lá existe menos que MinSpareServers,
# ele cria um novo processo. Se existe mais que MaxSpareServers, ele
# fecha alguns processos. Os valores abaixo estão adequados para muitos
# sites

MinSpareServers 5
MaxSpareServers 10

# Número de servidores que serão iniciados --- deve conter um valor razoável.

StartServers 5

# Limita o número total de servidores rodando, i.e., limita o número de clientes
# que podem conectar simultaneamente --- se este limite é sempre atingido,
# os clientes podem ser BARRADOS, assim este valor NÃO DEVE SER MUITO PEQUENO.
# Ele tem a intenção principal de ser um freio para manter um em execução com
# uma performance aceitável de acordo com os requerimentos de construção e
# carga calculada no servidor.

MaxClients 150

#
# MaxRequestsPerChild: O número de requisições que cada processo tem permissão
# de processar antes do processo filho ser finalizado. O filho será finalizado
# para evitar problemas após uso prolongado quando o Apache (e talvez as
# bibliotecas que utiliza) tomar memória e outros recursos. Na maioria dos
# sistemas, isto realmente não é necessário, exceto para alguns (como o
# Solaris) que possuem ponteiros notáveis em suas bibliotecas. Para estas
# plataformas, ajuste para algo em torno de 10000 ou algo assim; uma
# configuração de 0 significa ilimitado.
#
# NOTA: Este valor não inclui requisições keepalive após a requisição
# inicial por conexão. Por exemplo, se um processo filho manipula
# uma requisição inicial e 10 requisições "keptalive" subseqüentes,
# ele somente contará 1 requisição neste limite.
```

```

#
MaxRequestsPerChild 30

# Listen: Permite fazer o Apache escutar um IP determinado e/ou porta, em
# adição a padrão. Veja também o comando VirtualHost

#Listen 3000
#Listen 12.34.56.78:80

# VirtualHost: Permite o daemon responder a requisições para mais que um
# endereço IP do servidor, se sua máquina estiver configurada para aceitar pacotes
# para múltiplos endereços de rede. Isto pode ser feito com a opção de aliasing
# do ifconfig ou através de patches do kernel como o de VIF.

# Qualquer diretiva httpd.conf ou srm.conf pode ir no comando VirtualHost.
# Veja também a entrada BindAddress.

#<VirtualHost host.some_domain.com>
#ServerAdmin webmaster@host.some_domain.com
#DocumentRoot /var/www/host.some_domain.com
#ServerName host.some_domain.com
#ErrorLog /var/log/apache/host.some_domain.com-error.log
#TransferLog /var/log/apache/host.some_domain.com-access.log
#</VirtualHost>

# VirtualHost: Se você quiser manter múltiplos domínios/nomes de máquinas em sua
# máquina você pode ajustar o conteúdo de VirtualHost para eles.
# Por favor veja a documentação em <http://www.apache.org/docs/vhosts/>
# para mais detalhes antes de tentar configurar seus hosts virtuais.
# Você pode usar a opção de linha de comando '-S' para verificar sua configuração
# de hosts virtuais.

#
# Se desejar usar hosts virtuais baseados em nome, será necessário definir no
# mínimo um endereço IP (e número de porta) para eles.
#
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78

#
# Exemplo de um Host Virtual:
# Praticamente qualquer diretiva do Apache pode entrar na condicional
# VirtualHost.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#   ServerAdmin webmaster@host.some_domain.com
#   DocumentRoot /www/docs/host.some_domain.com
#   ServerName host.some_domain.com
#   ErrorLog logs/host.some_domain.com-error.log
#   CustomLog logs/host.some_domain.com-access.log common
#</VirtualHost>

```

```
#<VirtualHost _default_:*>
#</VirtualHost>
```

srm.conf

```
# Neste arquivo são definidos o espaço de nomes que os usuários visualizarão no
# seu servidor http. Este arquivo também define configurações do servidor que
# afetam como as requisições são servidas e como os resultados deverão ser
# formatados.
```

```
# Veja os tutoriais em http://www.apache.org/ para mais detalhes
```

```
# DocumentRoot: O diretório principal onde você servira seus documentos.
# Por padrão, todas as requisições são tomadas através deste diretório,
# exceto links simbólicos e aliases que podem ser usados para apontar para
# outras localizações no sistema de arquivos.
```

```
DocumentRoot /var/www
```

```
#
# UserDir: O nome do diretório que será adicionado ao diretório home do usuário
# caso uma requisição ~usuário for recebida.
```

```
#
<IfModule mod_userdir.c>
    # Linha abaixo por recomendação de segurança do manual do Apache
    UserDir disabled root
    UserDir public_html
</IfModule>
```

```
#
# DirectoryIndex: Nome do arquivo ou arquivos que serão usados como índice do
# diretório. Especifique mais de um arquivos separados por espaços ao invés
# de um só um nome (como "index") para aumentar a performance do servidor.
```

```
#
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi
</IfModule>
```

```
#
# Diretivas que controlam a exibição de listagem de diretórios geradas pelo servidor
#
```

```
<IfModule mod_autoindex.c>
```

```
#
# FancyIndexing: se você deseja o padrão fancy index ou padrão para a indexação
# de arquivos no diretório. Usando FancyIndexing o servidor
# apache gerará uma listagem de arquivos que poderá ser
# ordenada, usar tipos de ícones e encoding, etc. Veja as
# próximas opções
IndexOptions FancyIndexing
```

```

#
# As diretivas AddIcon* dizem ao servidor que ícone mostrar para um determinado
# arquivo ou extensão de arquivos. Estes somente são mostrados para os
# diretórios classificados através da opção FancyIndexing.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif */core
AddIcon /icons/deb.gif .deb Debian

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon é o ícone que será mostrado para aplicativos que não tiverem um
# ícone explicitamente definido.

DefaultIcon /icons/unknown.gif

#
# AddDescription: isto lhe permite colocar uma curta descrição após um arquivo
# nos índices gerados pelo servidor. Estes somente são mostrados para diretóri
# com índices organizados usando a opção FancyIndexing.
# Formato: AddDescription "descrição" extensão
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

# ReadmeName é o nome do arquivo LEIAME que o servidor procurará como
# padrão. Estes serão inseridos no fim da listagem de diretórios.
Formato: ReadmeName nome

```

```

#
# O servidor procurará primeiro por nome.html, incluído se ele for encontrado,
# e então procurará pelo nome e incluirá ele como texto plano se encontrado..

ReadmeName README

# HeaderName é o nome do arquivo que deve ser colocado no topo do índice
# de diretórios. As regras de procura de nome são as mesmas do arquivo
# README
HeaderName HEADER

#
# IndexIgnore: um conjunto de nomes de arquivos que a listagem de diretórios
# deve ignorar e não incluir na listagem. É permitido o uso de coringas
# como no interpretador de comandos.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

</IfModule>

# AccessFileName: O nome do arquivo que será procurado em cada diretório
# que contém detalhes sobre as permissões de acesso a um determinado
# diretório e opções de listagem. Tenha cuidado ao modificar o nome
# deste arquivo, muitas definições que trabalham em cima do nome
# .htaccess nos arquivos de configuração deverão ser modificados para
# não comprometer a segurança de seu servidor.
# Uma falta de atenção neste ponto poderá deixar este arquivo visível
# em qualquer listagem de diretórios facilmente...

AccessFileName .htaccess

# TypesConfig especifica o arquivo de configuração que contém os tipos
# usados pelo servidor
TypesConfig /etc/mime.types

#
# DefaultType é o tipo MIME padrão que o servidor utilizará para um documento
# caso ele não possa determinar seu conteúdo, como através de extensões
# de arquivos. Se o servidor contém em sua maioria texto ou documentos em HTML,
# "text/plain" é um bom valor. Caso a maioria do conteúdo seja binários, tal
# como aplicativos ou fotos, o tipo mais adequado ao seu caso poderá ser
# "application/octet-stream" para evitar que navegadores tentem exibir
# aplicativos binários como se fossem texto.
# Se desejar uma referência rápida sobre tipos mime, consulte o arquivo
# /etc/mime.types
#
DefaultType text/plain

#
# Document types.
#
<IfModule mod_mime.c>

# AddEncoding permite que alguns navegadores (Mosaic/X 2.1+, Netscape, etc)

```

```
# descompactem dados durante sua abertura. N
# Nota: Nem todos os navegadores suportam isto. Esqueça os nomes parecidos,
# as seguintes diretivas Add* não tem nada a ver com personalizações
# da opção FancyIndexing usada nas diretivas acima.

AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

#
# AddLanguage: permite especificar o idioma do documento. Você pode
# então usar a negociação de conteúdo para dar ao navegador um
# arquivo no idioma solicitado.
#
# Nota 1: O sufixo não precisa ser o mesmo da palavra chave do
# idioma --- estes com o documento em Polonês (no qual o
# código padrão da rede é pl) pode desejar usar "AddLanguage pl .po"
# para evitar confusão de nomes com a extensão comum de scripts
# scripts em linguagem Perl.
#
# Nota 2: As entradas de exemplos abaixo mostram que em alguns casos
# as duas letras de abreviação do 'Idioma' não é idêntico as duas letras
# do 'País' para seu país, como 'Danmark/dk' versus 'Danish/da'.
#
# Nota 3: No caso de 'ltz' nós violamos a RFC usando uma especificação de
# três caracteres. Mas existe um 'trabalho em progresso' para corrigir isto
# e obter os dados de referência para limpar a RFC1766.
#
# Danish (da) - Dutch (nl) - English (en) - Estonian (ee)
# French (fr) - German (de) - Greek-Modern (el)
# Italian (it) - Portugese (pt) - Luxembourgeois* (ltz)
# Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
# Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .ee
AddLanguage fr .fr
AddLanguage de .de
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
# AddCharset ISO-2022-JP .jis
AddLanguage pl .po
# AddCharset ISO-8859-2 .iso-pl
AddLanguage pt .pt
AddLanguage pt-br .pt-br
AddLanguage ltz .lu
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz

# LanguagePriority: permite definir a prioridade para a exibição de
```

```
# documentos caso nenhum documento confira durante a negociação de
# conteúdo.
#
# Para fazer isto, especifique os idiomas em ordem de preferência de exibição
# de idiomas.
#
<IfModule mod_negotiation.c>
    LanguagePriority pt-br pt es en da nl et fr de el it ja pl ltz ca sv
</IfModule>

#
# AddType permite modificar o mime.types sem editar o arquivo, ou fazer
# a associação de arquivos a certos tipos de conteúdo.
#
# Por exemplo, o módulo PHP 3.x (que não faz parte da distribuição do
# Apache - veja http://www.php.net) tipicamente utiliza isto:
#
#AddType application/x-httpd-php3 .php3
#AddType application/x-httpd-php3-source .phps
#
# E para arquivos PHP 4.x use:
#
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps

AddType application/x-tar .tgz
AddType image/bmp .bmp

# hhtml
AddType text/x-hhtml .hhtml

#
# AddHandler permite mapear certas extensões de arquivos a programas
# "manipuladores" adequados a seu conteúdo. Estes podem ser construídos
# no servidor ou adicionados com o comando Action (veja abaixo).
#
# Se desejar usar includes no lado do servidor, ou servir diretórios
# com scripts CGI para fora, descomente as seguintes linhas.
#
# Para usar scripts CGI:
#
#AddHandler cgi-script .cgi .sh .pl

#
# Para usar arquivos html gerados através do servidor
#
#AddType text/html .shtml
#AddHandler server-parsed .shtml

#
# Descomente as seguintes linhas para ativar a características de arquivos
# send-asis HTTP do servidor Apache
#
#AddHandler send-as-is asis
```

```

#
# Se desejar usar arquivos de mapas de imagens processadas no servidor, use
#
#AddHandler imap-file map

#
# Para ativar tipo de mapas, você poderá usar
#
#AddHandler type-map var

</IfModule>
# Fim dos tipos de documentos

# Preferências padrões de exibição de caracteres (veja http://www.apache.org/info/

AddDefaultCharset on
AddDefaultCharsetName iso-8859-1

# Redirect permite dizer aos clientes que documentos não existem mais no seu servi
# e a nova localização do documento.
# Format: Redirect nomeurl url
# "nomeurl" é o caminho especificado na url e "url" é a nova localização do
# documento

# Aliases: Inclua aqui quantos apelidos você desejar (sem limite) o formato é:
# Alias nomeurl nomereal
# "nomeurl" é o caminho especificado na url e "nomereal" é a localização
# do documento no sistema de arquivos local

# Note que se você incluir uma / no fim de "nomeurl", então o servidor
# requisitará que também esteja presente na URL.

Alias /icons/ /usr/share/apache/icons/
Alias /doc/ /usr/doc/
Alias /focalinux /var/www/focalinux
Alias /debian-br /var/www/debian-br/htdocs
Alias /debian /pub/mirror/debian

# ScriptAlias: Esta diretiva controla que diretórios contém scripts do servidor.
# Format: ScriptAlias fakename realname

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

#
# Action: permite definir os tipos de mídia que executarão um script quando um
# arquivo que conferir for chamado. Isto elimina a necessidade de caminhos de URLs
# repetidas para processadores de arquivos CGI freqüentemente usados.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#

#
# MetaDir: especifica o nome do diretório no qual o apache procurará arquivos de

```

```

# detalhes do módulo mod_cern_meta. Os módulos meta contém cabeçalhos HTTP
# adicionais que serão incluídos durante o envio do documento.
#
#MetaDir .web

#
# Resposta de erros personalizada (no estilo do Apache)
# estas podem ser 3 tipos:
#
# 1) texto plano
#ErrorDocument 500 "O servidor fez boo boo.
# n.b. a aspa (") marca como texto, ela não será exibida
#
# 2) redirecionamentos locais
#ErrorDocument 404 /missing.html
# para redirecionar para a URL local /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
# N.B.: É também possível redirecionar a um script o documento usando includes
# do lado do servidor (server-side-includes).
#
# 3) redirecionamentos externos
#ErrorDocument 402 http://algum.outro_servidor.com/inscricao.html
# N.B.: Muitas das variáveis de ambientes associada com a requisição atual *não*
# estarão disponíveis para tal script.

#
# O módulo mod_mime_magic permite o servidor usar várias dicas através do conteúdo
# do arquivo para determinar o seu tipo. A diretiva MIMEMagicFile diz ao módulo
# onde as definições de dicas estão localizadas. O módulo mod_mime_magic não é
# parte do servidor padrão Apache (você precisará adiciona-lo manualmente com
# uma linha LoadModule (veja o parágrafo DSO na seção Ambiente Global no
# arquivo httpd.conf), ou recompile o servidor e inclua mod_mime_magic como
# parte de sua configuração), por este motivo ele está entre as condicionais
# <IfModule>. Isto significa que a diretiva MIMEMagicFile somente será processada
# caso o módulo estiver ativo no servidor.
#
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>

<IfModule mod_setenvif.c>
#
# As seguintes diretivas modificam o funcionamento da resposta normal do
# servidor HTTP.
# A primeira diretiva desativa o keepalive para o Netscape 2.x e navegadores q
# as falsificam. Existem problemas conhecidos com estas implementações de
# navegadores. A segunda diretiva é para o MS IE 4.0b2 que tem uma implementaçã
# defeituosa do HTTP/1.1 e não suporta adequadamente o keepalive quando ele
# utiliza as respostas de redirecionamento 301 e 302.
#
    BrowserMatch "Mozilla/2" nokeepalive
    BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0

```

```

#
# As seguintes diretivas desativam as respostas HTTP/1.1 para navegadores que
# violam a especificação HTTP/1.0 não sendo capaz de enviar uma resposta
# 1.1 básica.
#
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>

# Se o módulo Perl está instalado, isto será ativado.
<IfModule mod_perl.c>
  Alias /perl/ /var/www/perl/
  <Location /perl>
    Options +ExecCGI
    SetHandler perl-script
    PerlHandler Apache::Registry
  </Location>
</IfModule>

```

access.conf

```

# access.conf: Configuração de acesso Global
# Documentos on-line em http://www.apache.org/

# Este arquivo define as configurações do servidor que afetam que tipos de
# serviços são permitidos e em quais circunstâncias.

# Cada diretório que o Apache possui acesso, pode ser configurado respectivamente
# com quais serviços e características que podem ser permitidas e/ou bloqueadas
# no diretório (e seus subdiretórios).
#
# Primeiro a configuração restringe uma série de permissões
<Directory />
  Options SymLinksIfOwnerMatch
  AllowOverride None
#   Order deny,allow
#   Deny from all
</Directory>

# Desse ponto em diante, é necessário especificar o que será permitido
# caso contrário será bloqueado pelo bloco acima

# Esta parte deve ser modificada para a localização do documento raiz do servidor.

<Directory /var/www>

# A opção Options pode conter os valores "None", "All", ou quaisquer combinação
# de "Indexes", "Includes", "FollowSymLinks", "ExecCGI", ou "MultiViews".
#
# Note que "MultiViews" deve ser *explicitamente* especificada --- "Options All"
# não a ativa (pelo menos não ainda).

```

```
Options Indexes FollowSymLinks Includes MultiViews
```

```
# Esta opção controla que opções os arquivos .htaccess nos diretórios podem ser
# substituídas. Pode também conter "All", ou qualquer combinação de "Options",
# "FileInfo", "AuthConfig", e "Limit"
```

```
AllowOverride None
```

```
# Controla quem pode obter materiais deste servidor. Leia a seção adequada no
# guia para mais explicações sobre a ordem de acesso, padrões e valores permitidos
```

```
order allow,deny
allow from all
```

```
</Directory>
```

```
#
# O diretório "/usr/lib/cgi-bin" deve ser modificado para o diretório que
# possuem seus scripts CGI, caso tenha configurado o suporte a CGI's no
# servidor.
```

```
#
<Directory /usr/lib/cgi-bin/>
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
```

```
</Directory>
```

```
#
# Permite ver relatórios de status e funcionamento do servidor web e
# processos filhos, através da URL http://servidor/server-status
# isto requer o módulo status_module (mod_status.c) carregado no arquivo
# httpd.conf
```

```
#
#<Location /server-status>
#     SetHandler server-status
#     Order deny,allow
#     Deny from all
#     Allow from .meudominio.org
#</Location>
```

```
#
# Permite relatório de configuração remota do servidor, através da URL
# http://servername/server-info
# Isto requer o módulo info_module (mod_info.c) carregado no arquivo
# httpd.conf
```

```
#
#<Location /server-info>
#     SetHandler server-info
#     Order deny,allow
#     Deny from all
#     Allow from .meudominio.org
#</Location>
```

```

# Visualização do diretório de ícones
<Directory /usr/share/apache/icons>
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# O Debian Policy assume que /usr/doc é "/doc/" e linkado com /usr/share/doc,
# pelo menos para localhost.

<Directory /usr/doc>
    Options Indexes FollowSymLinks
    order deny,allow
    deny from all
    allow from 192.168.1.10/24
</Directory>

# Esta define a localização visualizável do monitor de status mod_throttle
#
<location /throttle-info>
    SetHandler throttle-info
</location>

#
# As seguintes linhas previnem os arquivos .htaccess de serem mostrados nos
# clientes Web. Pois os arquivos .htaccess freqüentemente contém detalhes
# de autorização, o acesso é desabilitado por razões de segurança. Comente
# estas linhas se desejar que seus visitantes vejam o conteúdo dos arquivos
# .htaccess. Se modificar a diretiva AccessFileName acima, tenha certeza de
# fazer as modificações correspondentes aqui.
#
# As pessoas também tendem a usar nomes como .htpasswd nos arquivos de senhas
# a diretiva abaixo os protegerá também.
#
<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>

#
# Controla o acesso a diretórios UserDir. As seguintes diretivas são um exemplo
# para um site onde estes diretórios estão restritos a somente-leitura. Veja
# detalhes sobre as opções de acesso, e limites na seção sobre controle
# de acesso do guia
#
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all

```

```
</Limit>
<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
    Order deny,allow
    Deny from all
</Limit>
</Directory>

#
# As vezes ocorrem relatos de pessoas tentando abusar de uma falha antiga nos
# dias do Apache 1.1 (muitas páginas na Net documentam isso). Esta falha envolve
# um script CGI distribuído como parte do Apache. Descomentando estas linhas você
# poderá redirecionar estes ataques a um script de registro em phf.apache.org. Ou
# poderá gravar em sua própria máquina, usando o script support/phf_abuse_log.cgi.
#
#<Location /cgi-bin/phf*>
#     Deny from all
#     ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

# Acesso aos serviços proxy do apache
#<Directory proxy:*>
#     Order deny,allow
#     Deny from all
#     Allow from .your_domain.com
#</Directory>

# a seguinte diretiva permite o acesso a todos os usuários ao conteúdo da página
# do guia Foca GNU/Linux exceto os que possuem navegadores MSIE ;-)
# Veja a seção sobre restrições de acesso para detalhes sobre a diretiva de
# controle de acesso baseado no user-agent
SetEnvIf User-Agent MSIE EXPLODER
<Directory /var/www/focalinux>
    Options Indexes
    Order allow,deny
    allow from all
    deny from env=EXPLODER
    ErrorDocument 403 "Explorer não entra, página com o conteúdo potencialmente perig
</Directory>

# A diretiva abaixo somente permite acesso a leitura do arquivo
# h-supor-fonte.txt a pessoas que fornecerem o nome/senha corretos
# que constam no arquivo passwd1
# Este bloco contém um erro que é a localização do arquivo da senha em um
# diretório público, você deverá adapta-lo se não quiser se ver em apuros.
#
# A permissão do diretório de nível superior prevalece sobre seus
# sub-diretórios no caso as permissões de /focalinux, a menos que
# sejam definidas opções de acesso específicas ao arquivo abaixo
<Location /focalinux/humor/h-supor-fonte.txt>
    AuthName "Piada de fonte de alimentação"
    AuthType basic
    AuthUserFile /home/gleydson/public_html/passwd1
    Require valid-user
```

```
# Satisfy all
</Location>

# Libera o acesso a localização /debian (acessada através de /pub/mirror/debian,
# definida no Alias acima)
<Location /debian>
  Options Indexes
  Order deny,allow
  allow from all
  deny from all
</Location>
```

Códigos HTTP

Esta seção pode ser uma interessante referência para a programação e configuração da diretiva *ErrorDocument*, etc.

- | | |
|------------------------|---|
| 2xx - Sucesso | <ul style="list-style-type: none"> • 200 OK • 201 Criado • 202 Aceito • 203 Informação não-autoritativa * • 204 Nenhum conteúdo • 205 Conteúdo resetado * • 206 Conteúdo parcial * |
| 3xx - Redirecionamento | <ul style="list-style-type: none"> • 300 Múltiplas escolhas • 301 Movido Permanentemente • 302 Movido Temporariamente • 303 Veja outra * • 304 Não modificada • 305 Use o Proxy (redirecionamento proxy) * |
| 4xx - Erros no Cliente | <ul style="list-style-type: none"> • 400 Requisição incorreta • 401 Não autorizado • 402 Pagamento Requerido * • 403 Bloqueado • 404 Não encontrada • 405 Método não permitido * • 406 Não aceitável * |

- 407 Autenticação via proxy requerida *
- 408 Tempo limite da requisição expirado *
- 409 Conflito *
- 410 Gone *
- 411 Tamanho requerido *
- 412 Falha na pré-condição *
- 413 A requisição parece ser grande *
- 414 A URL requisitada é muito longa *
- 415 Tipo de mídia não suportado
- 500 Erro Interno no Servidor
- 501 Não implementado
- 502 Gateway incorreto
- 503 Serviço não disponível
- 504 Tempo limite no gateway *
- 505 Versão HTTP não suportada *

5xx - Erros no Servidor

Os códigos de erros marcados com um "*" pertencem ao padrão HTTP 1.1

Capítulo 13. Servidor ident

Este capítulo documenta o uso, benefícios, configuração, utilização e exemplos do servidor `identd`. Também são explicados alguns pontos positivos/negativos de sua utilização para aumentar a segurança quando usado junto com o mecanismo de controle de acesso.

O servidor `identd` escolhido para ser descrito nesta seção do guia foi o **`oidentd`**.

Introdução

O `ident` (identidade) é um servidor que permite identificar qual o usuário efetuou determinada conexão e o sistema operacional usado. Ele opera na porta 113 por padrão e retorna nomes de usuários localmente válidos, e é consultado por serviços conhecidos como IRC, alguns servidores `ftp`, `smtp` e outros. Outro benefício é a utilização de mecanismos de restrições de acesso baseadas em usuários/endereçoIP (o **`tcpd`** é um exemplo de serviço que permite esta característica). A sintaxe usada para fazer tal restrição é universal: *usuário@endereçoIP* onde normalmente aparece o *endereçoIP* que é usado para bloquear/permitir o acesso.

No momento da conexão, o endereço IP é checado pra ver se confere, e o servidor `Ident` da máquina que está efetuando a conexão é consultado para checar se o usuário que tem acesso é o mesmo especificado no controle de acesso. Isso aumenta um pouco a segurança do sistema, mas existem algumas implicações e pontos frágeis do **`identd`** que serão explicados no decorrer deste capítulo.

Versão

É assumido que esteja usando a versão 1.7 do **`oidentd`**. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendável que leia a documentação sobre modificações no programa (`changelog`) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

Contribuindo

A Home page do projeto **`oidentd`** é <http://ojnk.sourceforge.net> Sugestões, críticas, comentários, etc., podem ser enviados para `<odin@numb.org>`.

Características

Características do **`oidentd`**:

- Pode ser executado tanto como *daemon* quanto via *inetd* (este último é indicado para sistemas com pouca memória onde o serviço é pouco solicitado).
- Pode mapear identificações de usuário via IP Masquerading, tornando este servidor muito versátil podendo ser usado tanto em máquina individuais como em servidores proxy/roteadores.
- Pode fazer forwarding de conexões para outras máquinas da rede local, quando não é executado no proxy/roteador.
- Spoofing de nomes: é possível mapear um nome de usuário para outra identificação; por exemplo, o usuário `root` poderá ser mapeado para outra conta de usuário antes da identificação ser enviada.

Ficha técnica

Pacote:

Utilitários:

- `oidentd` - Servidor identd

Arquivos de configuração do **oidentd**:

`identd.spoof` Controla o spoof (falsificação) de nomes de usuários. O formato deste arquivo são dois campos separados por ":", o primeiro contendo a identificação original do usuário e o segundo o nome que será enviado pelo **identd**. O segundo campo pode ser omitido, neste caso a resposta de identificação é lida através do arquivo `~/ .ispoof`.

Este arquivo deve ter como dono o usuário do primeiro campo do `identd.spoof` e a identificação retornada será a contida no arquivo. Esteja certo que o daemon **oidentd** tem permissões para acessar este arquivo, caso contrário nenhum spoof de identidade será realizado.

Para o spoof ser habilitado, o serviço **oidentd** deverá ser iniciado com a opção `-s` ou `-S` (veja mais detalhes “Opções de linha de comando”).

OBS: Certifique-se de colocar as permissões adequadas para que somente o daemon `oidentd` tenha acesso a este arquivo (de acordo com o *usuário* e *grupo* usado para executar o **oidentd**), os detalhes de mapeamento de nomes podem ser perigosos em mãos erradas, e garantir o sucesso de uma conexão indesejável.

`oidentd.users` Mapeamento de nomes de usuários efetuando conexões via Masquerading. O formato deste arquivo é o seguinte:

#EndereçoIP/máscara	Usuário	Sistema
192.168.1.1	john	WINDOWS
192.168.1.2	usuario1	WINDOWS
192.168.1.1/32	usuario2	UNIX
192.168.1.0/24	usuario3	UNIX
192.168.1.0/16	usuario4	UNIX

As conexões vindas dos endereços da primeira coluna são mapeados para o nome/sistema da segunda/terceira coluna e enviados a máquina que requisitou a identificação. Para o suporte a mapeamento de usuários via Masquerading funcionar, o daemon **oidentd** deverá ser iniciado com a opção `-m`.

Requerimentos de Hardware

O **oidentd** requer pouca memória e pode ser executado sem problemas em um sistema com o mínimo de memória necessária para rodar o kernel do **Linux** (2 MB para 2.2 e 4MB para as séries 2.4 do kernel). Mesmo assim é interessante considerar 1 MB a mais que o mínimo requerido pelo kernel para uma folga na execução do serviço de identificação junto a outros do sistema.

Arquivos de log criados pelo Ident

Mensagens informativas, erros, e outras sobre execuções do serviço **oidentd** são enviadas ao **syslog** do sistema.

Instalação

Para instalar o daemon do **oidentd** digite:

```
apt-get install oidentd
```

Por padrão o serviço é instalado para ser executado como daemon, para executa-lo através do **inetd** siga os passos em “Instalação via Inetd”. O serviço será executado sob o usuário **nobody** e grupo **nogroup** por motivos de segurança, alterações de nome/grupo que executará o **oidentd** podem ser feitas no arquivo `/etc/defaults/oidentd` ou `/etc/init.d/oidentd`.

Instalação via Inetd

Siga os procedimentos de instalação em “Instalação” e os seguintes passos:

1. Edite o arquivo `/etc/inetd.conf` e adicione a seguinte linha:

```
#:INFO: Info services
auth stream tcp nowait.40 nobody.nogroup /usr/sbin/oidentd oidentd -q -i -t 40
```

A opção `-i` permite o **oidentd** aceitar requisições via **inetd** (sem ela ele será executado no modo daemon). As opções `-s` e `-m` devem também ser especificadas caso desejar os recursos de falsificação de identificação (mapeamento de nomes) e masquerading (veja “Opções de linha de comando”). Aqui foi definido um parâmetro máximo de 40 requisições por minuto (típico de um serviço poucos usado no sistema), caso este limite seja ultrapassado o serviço será desativado na seção atual do **inetd**. Os outros campos são descritos em “`/etc/inetd.conf`”.

2. Interrompa a execução do daemon do **oidentd** atual dando um `./etc/init.d/oidentd stop`.
3. Remova os links dos runlevels em `/etc/rc?.d` que iniciam/interrompem a execução do daemon com o comando: `update-rc.d -f oidentd remove`. Neste ponto o daemon **oidentd** não será mais iniciado. Para reverter esta ação, execute o comando: `update-rc.d oidentd defaults`.
4. De um comando `killall -HUP inetd` para fazer o serviço **inetd** recarregar o arquivo de configuração `/etc/inetd.conf`. O serviço de `identd` já estará funcionando.

OBS: A configuração da distribuição **Debian** permite detectar quando o serviço `ident` (auth) está sendo executado no `/etc/inetd.conf` através de seus scripts de inicialização. Você poderá fazer as coisas manualmente baseado nisso se desejar.

Usando tcpwrappers com oidentd

Especifique a opção `-W` para fazer o **oidentd** utilizar o mecanismo de acesso em `hosts.allow` e `hosts.deny` para garantir/bloquear ao serviço de acordo com endereços/hosts especificados.

OBS O **oidentd** é somente executado após a conferência de todos os parâmetros de endereços nestes arquivos de acesso, não utilize a sintaxe “usuário@endereço” como endereço na linha de acesso do serviço **oidentd** (por motivos óbvios).

Iniciando o servidor/reiniciando/recarregando a configuração

O arquivo que controla o funcionamento do daemon do **oidentd** é controlado pelo arquivo `/etc/init.d/oidentd`.

A execução do **oidentd** através de **inetd** é automática quando é feita uma requisição para a porta `113`.

Opções de linha de comando

Opções de linha de comando do **oidentd**:

- `-a [endereçoIP]` - Espera por requisições somente no nome ou endereço IP da interface especificada.
- `-A` Quando o spoofing esta ativado, permite os usuários falsificarem o ident em conexões para portas privilegiadas.
- `-c [página código]` - Especifica uma página de código alternativa. O padrão é "US-ASCII".
- `-d` - Ativa o modo de depuração, mais detalhes serão exibidos.
- `-e` - Retorna "UNKNOWN-ERROR" (erro desconhecido) para qualquer tipo de erro.
- `-f [porta]` - Redireciona requisições de máquinas usando MASQUERADE para o computador na porta especificada.
- `-F` - O mesmo que `-f`, mas usa a porta 113 como padrão.
- `-g [gid]` - Executa o daemon do **oidentd** no grupo especificado.
- `-i` - Permite ser executado através do **inetd**.
- `-m` - Ativa o suporte a IP Masquerading.
- `-n` - Retorna números UID ao invés de nomes de usuários.
- `-N` - Permite ocultar a identificação de determinados usuários através de arquivos `~/ .noident`.
- `-o` - Retorna "OTHER" (outro qualquer) ao invés do sistema operacional especificado.
- `-p [porta]` - Espera por conexões na porta especificadas (a padrão é a 113 - serviço auth).
- `-q` - Oculta o logging normal.
- `-P [proxy]` - O proxy especificado (endereço IP) faz redirecionamento de conexões para a máquina executando o **oidentd**.
- `-r` - Retorna respostas aleatórias de `identd`. As opções `-n` e `-r` não podem ser usadas juntas.
- `-s` - Permite utilizar os mecanismos de spoofing (falsificação) do **oidentd**.
- `-S` - O mesmo que `-s` mas permitem todos os usuários EXCETO os especificados em `/etc/identd.spoof` falsificarem suas respostas.
- `-t [segundos]` - Espera o tempo especificado antes de ser encerrado.

- -T [segundos] - O **oidentd** permanecerá aceitando conexões quando é executado com a opção -w pelo número de segundos especificado.
- -u [uid] - Executa o servidor **oidentd** com a uid especificada.
- -v/-V - Mostra detalhes sobre a versão do servidor.
- -w - Modo de espera de conexões.
- -x [texto] - Se uma requisição falha, o texto especificado é retornado.
- -W - Utiliza os mecanismos de acesso `hosts.allow` e `hosts.deny` do **tcpd**.
- -h - Mostra as opções de linha de comando do **oidentd**.

Exemplos

Não faz muito sentido exemplos de arquivo de configuração do **oidentd** por estes serem muito simples e estarem bem explicados em “Ficha técnica”. No entanto acho interessante mostrar alguns exemplos de configurações do `hosts.allow` e `hosts.deny` fazendo uso dos recursos de restrições baseadas em `usuário@endereço` :

```
# Arquivo hosts.allow
#
# Permite requisições talk de qualquer lugar
in.ntalkd: ALL
in.talkd: ALL
#
# Permite que o usuário john acesse os serviços de ftp de qualquer máquina da
# rede 191.168.1.*
in.ftpd: john@192.168.1.
#
# O serviço telnet está permitido somente para john conectando de 192.168.1.1
in.telnetd: john@192.168.1.1
# Todos podem acessar os serviços samba (nomes e compartilhamentos) exceto
# o usuário evil conectando de qualquer host com o endereço cracker.com.*
smbd, nmbd: ALL EXCEPT evil@cracker.com.

# Arquivo hosts.deny
# Qualquer finger é bloqueado exceto vindos do usuário admin feitos em qualquer
# máquina da rede 192.168.1.*
in.fingerd: ALL EXCEPT admin@192.168.1.
# Qualquer outra coisa é bloqueada
ALL: ALL
```

Capítulo 14. Servidor telnet

Este capítulo ensina como instalar, configurar, usar e fazer restrições de acesso ao servidor telnet. Também é explicada a utilização do cliente telnet e o suporte a criptografia (ssl).

Introdução

O serviço telnet é oferecido o login remoto em seu computador, que lhe permite trabalhar conectado a distância como se estivesse em frente a ela. Ele substitui o **rlogin** e possui muitas melhorias em relação a ele, como o controle de acesso, personalização de seção e controle de terminal.

Versão

É assumido que esteja usando a versão 0.17.16 do **telnet**. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendável que leia a documentação sobre modificações no programa (changelog) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

Características

- Conexão rápida (não utiliza transmissão de dados criptografada), recomendado para ambientes seguros.
- Possui uma versão com suporte a criptografia via ssl.
- Possui controle de acesso tcpd (usando `/etc/hosts.allow` e `/etc/hosts.deny`).
- A maioria dos sistemas operacionais trazem este utilitário por padrão como sistema de acesso remoto a máquinas UNIX.
- Suporte a terminais ANSI (cores e códigos de escape especiais para o console) e uma grande variedade de outros terminais.

Ficha técnica

Pacotes:

- - Cliente telnet com suporte a autenticação.
- - Servidor telnet com suporte a autenticação.
- - Cliente telnet com suporte a autenticação e ssl. Também suporta conexão a servidores telnet padrão quando o servidor não suporta ssl. Por padrão é tentada a conexão usando ssl, se esta falhar será assumida a transmissão em texto plano.
- - Servidor telnet com suporte a autenticação e ssl. Também suporta conexão de clientes telnet padrão (sem suporte a ssl).

Utilitários:

- `in.telnetd` - Servidor telnet
- `telnet` - Cliente telnet padrão (quando o pacote está instalado, é simplesmente um link para `telnet-ssl`).
- `telnet-ssl` - Cliente telnet com suporte a ssl.

Requerimentos de Hardware

Normalmente o servidor telnet é carregado via `inetd`, o que permite sua utilização em uma máquina com a quantidade mínima de memória RAM requerida para o funcionamento do kernel: 2 MB para kernels da série 2.2 e 4MB para kernels da série 2.4.

Arquivos de log criados pelo servidor telnet

Mensagens do servidor telnet relacionadas com seções são enviadas para `/var/log/daemon.log`. Adicionalmente, as mensagens sobre autenticação (serviços de login) são registradas pelos módulos PAM em `/var/log/auth.log`.

Instalação

```
apt-get install telnet telnetd ou apt-get install telnet-ssl telnetd-ssl.
```

Os pacotes com o `-ssl` no final possuem suporte a criptografia ssl. Por padrão a porta usada para executar o serviço telnet é a 23 (ou outro número de porta definido no `/etc/services`). A instalação do servidor telnet é feita via `inetd` (no arquivo `/etc/inetd.conf`) e o controle de acesso ao serviço é feito através dos arquivos `/etc/hosts.allow` e `/etc/hosts.deny` (veja “Serviços iniciados através do `inetd`” e “O mecanismo de controle de acessos `tcpd`”).

O servidor tem o nome `in.telnetd` e este deverá ser usado para ajustar o controle de acesso nos arquivos acima.

Iniciando o servidor/reiniciando/recarregando a configuração

O arquivo que controla o funcionamento do servidor **telnet** é o `/etc/inetd.conf` e o controle de acesso sendo feito pelos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`. Será necessário reiniciar o servidor **inetd** caso algum destes três arquivos seja modificado: `killall -HUP inetd`. A porta de operação padrão é a 23 e pode ser modificada no arquivo `/etc/services`.

Opções de linha de comando

Opções de linha de comando do servidor **telnetd**:

- `-D nível_de_depuração` - Permite especificar o que será registrado pelo servidor durante a conexão dos clientes telnet. As seguintes opções são suportadas:
 - `options` - Mostra detalhes sobre a negociação das opções de conexão.
 - `report` - Mostra detalhe de opções e o que está sendo feito.
 - `netdata` - Mostra os dados transferidos na conexão telnetd.
 - `ptydata` - Mostra os dados mostrados na `pty`.
- `-edebug` - Ativa a depuração do código de criptografia apenas para o servidor telnet com suporte a ssl.
- `-h` - Somente mostra os detalhes de configuração do seu PC após o usuário fornecer um nome/senha válidos.

- `-L [programa]` - Utiliza o programa especificado para fazer o login do usuário (`/usr/sbin/telnetlogin` é o padrão).
- `-n` - Não envia pacotes `keep alive` para verificar o estado da conexão. Desativando esta opção poderá fazer o servidor ficar rodando constantemente caso aconteça algum problema e o usuário não consiga se desconectar normalmente.
- `-S TOS` - Ajusta o tipo de serviço usado na conexão para o valor especificado (veja “Especificando o tipo de serviço” para maiores detalhes sobre esta opção e os valores aceitos).

Estas opções deverão ser especificadas após o servidor `in.telnetd` no arquivo `/etc/inetd.conf`.

Controle de acesso

É feito pelos arquivos `hosts.allow` e `hosts.deny`. Veja “O mecanismo de controle de acessos tcpd”.

Recomendações

O serviço telnet utiliza texto plano para seção (exceto nas versões cliente/servidor “-ssl”). Os dados transmitidos por serviços que utilizam texto plano podem ser capturados por sniffers e trazer perigo ao seu sistema (veja “Sniffer”).

É recomendável somente executar o servidor telnet padrão em ambientes seguros (como em uma rede interna) e a versão com suporte a ssl para fazer conexões via redes inseguras (como a Internet). O serviço **ssh** (???) é uma excelente alternativa ao telnet, além de possuir outras características adicionais que justifiquem seu uso, além de programas cliente para **Linux** e **Windows**.

Fazendo conexões ao servidor telnet

Use o comando: `telnet [endereço] [porta]` para realizar conexões com uma máquina rodando o servidor telnet.

Adicionalmente as seguintes opções podem ser usadas:

- `-l [usuario]` - Envia o nome de usuário ao computador remoto. Muito útil com o **telnet-ssl**.
- `-E` - Desativa o caracter de escape
- `-a` - Tenta fazer o login automático usando o nome de usuário local. Se o login falhar, será solicitado o nome de usuário. Esta opção é usada por padrão com o cliente **telnet-ssl**.
- `-r` - Emula o comportamento do programa **rlogin**.

Exemplos:

```
# Conecta-se ao servidor telnet rodando na porta 23 de sua própria máquina
telnet localhost
```

```
# Conecta-se ao servidor telnet 200.200.200.200 operando na porta 53454 usando o
# nome de usuário john
telnet -l john 200.200.200.200 53454
```

Capítulo 15. Servidor ssh

Este capítulo documenta a instalação, configuração e personalização do servidor de shell seguro **ssh**, além de explicar as vantagens da utilização dos serviços criptográficos. A utilização do programa cliente **ssh** também é explicada, além de utilitários usados para geração de chaves pública/privada para o **ssh** (autenticação RSA/DAS - o que é, vantagens), cópia de arquivos e métodos de autenticação usando o método de chave pública/privada RSA.

Ambas as versões 1 e 2 do **ssh** são documentadas neste capítulo. Opções específicas do protocolo 1 ou 2 do **ssh** serão destacadas.

Introdução

O serviço de **ssh** permite fazer o acesso remoto ao console de sua máquina, em outras palavras, você poderá acessar sua máquina como se estivesse conectado localmente ao seu console (substituindo o **rlogin** e **rsh**). A principal diferença com relação ao serviço **telnet** padrão, **rlogin** e **rsh** é que toda a comunicação entre cliente/servidor é feita de forma encriptada usando chaves públicas/privadas RSA para criptografia garantindo uma transferência segura de dados.

A velocidade do console remoto conectado via Internet é excelente (melhor que a obtida pelo **telnet** e serviços **r***) dando a impressão de uma conexão em tempo real (mesmo em links discados de 9.600 KB/s), a compactação dos dados também pode ser ativada para elevar ainda mais a velocidade entre cliente-servidor **ssh**. Além do serviço de acesso remoto, o **scp** possibilita a transferência/recepção segura de arquivos (substituindo o **rmp**).

Em conexões sem criptografia (**rsh**, **rlogin**) os dados trafegam de forma desprotegida e caso exista algum sniffer instalado em sua rota com a máquina destino, todo o que fizer poderá ser capturado (incluindo senhas).

Versão

É assumido que esteja usando a versão 2.0 do **ssh**. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendável que leia a documentação sobre modificações no programa (**changelog**) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

História

O **openSSH** (explicado neste capítulo) é baseado na última versão livre do implementação de Tatu Ylonen com todos os algoritmos patenteados (para bibliotecas externas) removidos, todos as falhas de segurança corrigidas, novas características e muitas outras melhorias. O **openSSH** foi criado por Aaron Campbell, Bob Beck, Markus Friedl, Niels Provos, Theo de Raadt e Dug Song.

Contribuindo

A Home page principal é <http://www.unixuser.org/~haruyama/security/openssh/index.html>. Falhas, correções e sugestões podem ser enviadas para a lista de discussão <openssh-unix-dev@mindrot.org> (aberta a postagens de usuários não inscritos).

Características

Abaixo as principais características do serviço **ssh** (**Openssh**).

- Conexão de dados criptografada entre cliente/servidor.
- Cópia de arquivos usando conexão criptografada.
- Suporte a ftp criptografado (sftp).
- Suporte a compactação de dados entre cliente/servidor.
- Controle de acesso das interfaces servidas pelo servidor **ssh**.
- Suporte a controle de acesso tcp wrappers.
- Autenticação usando um par de chaves pública/privada RSA ou DSA.
- Algoritmo de criptografia livre de patentes.
- Suporte a PAM.
- Suporte a caracteres ANSI (cores e códigos de escape especiais no console).

Ficha técnica

Pacote:

Utilitários:

- `ssh` - Cliente ssh (console remoto).
- `slogin` - Link simbólico para o programa **ssh**.
- `sshd` - Servidor de shell seguro ssh.
- `scp` - Programa para transferência de arquivos entre cliente/servidor
- `ssh-keygen` - Gera chaves de autenticação para o ssh
- `sftp` - Cliente ftp com suporte a comunicação segura.
- `sftp-server` - Servidor ftp com suporte a comunicação segura.
- `ssh-add` - Adiciona chaves de autenticação DSA ou RSA ao programa de autenticação.
- `ssh-agent` - Agente de autenticação, sua função é armazenar a chave privada para autenticação via chave pública (DSA ou RSA).
- `ssh-keyscan` - Scaneia por chaves públicas de autenticação de hosts especificados. O principal objetivo é ajudar na construção do arquivo local `known_hosts`.
- `ssh-copy-id` - Usado para instalação do arquivo `identity.pub` em uma máquina remota.

Arquivos de configuração:

- `/etc/ssh/sshd_config` - Arquivo de configuração do servidor ssh.
- `/etc/ssh/ssh_config` - Arquivo de configuração do cliente ssh.

- `~/.ssh/config` - Arquivo de configuração pessoal do cliente ssh.

Requerimentos de Hardware

É recomendado no mínimo 6MB de memória RAM para a execução do serviço **ssh** mais o kernel do **Linux**. Este limite deve ser redimensionado para servidores de acesso dedicado, uma quantidade de 64MB deve ser confortável para centenas de usuários conectados simultaneamente (o que raramente acontece).

Veja também Capítulo 19, *Restrições de acesso, recursos e serviços* para configuração de restrições usando PAM. O **ssh** que acompanha a distribuição **Debian** vem com o suporte a `tcp wrappers` compilado por padrão.

Arquivos de log criados pelo servidor ssh

Detalhes sobre a execução do servidor **sshd** (como início, autenticação e término) são enviadas ao **syslog** do sistema. A *prioridade* e *nível* são definidos no arquivo de configuração `/etc/ssh/sshd_config` (veja “Exemplo de `sshd_config` com explicações das diretivas”).

Instalação do servidor openSSH

```
apt-get install ssh.
```

Por padrão o servidor **sshd** é instalado como `daemon`, também é possível executa-lo via **inetd** mas isto não é aconselhável porque o servidor gera uma chave aleatória de seção toda vez que é iniciado, isto podendo levar vários segundos (quando é usada a versão 1 do protocolo ssh, veja “Diferenças nas versões do protocolo”).

Iniciando o servidor/reiniciando/recarregando a configuração

O arquivo que controla o funcionamento do `daemon` do **ssh** é controlado pelo arquivo `/etc/init.d/ssh`.

A execução do **ssh** através de **inetd** é automática quando é feita uma requisição para a porta 22.

Opções de linha de comando

Opções de linha de comando do servidor **sshd**:

- `-b bits` - Especifica o número de bits da chave do servidor (768 por padrão).
- `-d` - Modo de depuração - O servidor envia detalhes sobre seu funcionamento aos logs do sistema e não é executado em segundo plano. Ele também responderá conexões pelo mesmo processo. Podem ser usadas no máximo 3 opções `-d` para aumentar os detalhes de depuração.
- `-f arquivo_configuração` Indica um arquivo de configuração alternativo (por padrão é usado `/etc/ssh/sshd_config`). O **ssh** pode ser configurado através de opções de linha de comando mas requer um arquivo de configuração para ser executado. Opções de linha de comando substituem as especificadas no arquivo de configuração.
- `-g segundos` - Especifica o tempo máximo para a digitação de senha de acesso. Após o tempo especificado o servidor encerra a conexão. O valor padrão é 600 segundos e 0 desativa este recurso.

- `-h arquivo_chave` - Diz qual arquivo contém a chave privada local. O padrão é `/etc/ssh/ssh_host_key` e somente o usuário `root` deve ter permissões de leitura neste arquivo. Será necessário especificar esta opção caso o `sshd` não esteja sendo executado como usuário `root`.

É possível ter múltiplos arquivos de chaves para os protocolos 1 e 2 do ssh.

- `-i` - Indica que o servidor `sshd` será executado pelo `inetd`. Isto não é aconselhável porque o servidor gerará a chave aleatória de seção toda vez que for iniciado e isto pode levar alguns segundos. Esta opção pode se tornar viável com o uso do protocolo 2 ou criando chaves pequenas como 512 bytes (no ssh 1), mas a segurança criptográfica também será diminuída. Veja as diferenças entre os dois protocolos em “Diferenças nas versões do protocolo”.
- `-k segundos` - Especifica a frequência da geração de novas chaves do daemon `sshd`. O valor padrão é 3600 segundos e 0 desativa este recurso.

ATENÇÃO: NÃO desative este recurso!!! Esta opção traz a segurança que uma nova chave gerada de servidor será gerada constantemente (esta chave é enviada junto com a chave pública quando o cliente conecta e fica residente na memória volátil), assim mesmo que um cracker consiga obtê-la interceptando as conexões, será praticamente impossível tentar qualquer coisa. Valores menores tendem a aumentar ainda mais a segurança.

- `-p porta` - Especifica a porta que o daemon `sshd` atenderá as requisições. Por padrão é usada a porta 22.
- `-q` - Nenhuma mensagem será enviada ao `syslog` do sistema.
- `-u tam` - Especifica o tamanho do campo de nome do computador que será armazenado no arquivo `utmp`. A opção `u0` faz somente endereços IP serem gravados.
- `-D` - Quando usada não faz o `sshd` iniciar em segundo plano.
- `-V versão_cliente` - Assume que o cliente possui a versão ssh especificada (1 ou 2) e não faz os testes de identificação de protocolo.
- `-4` - Força o uso do protocolo IP tradicional (IPv4).
- `-6` - Força o uso da nova geração do protocolo IP (IPv6).

A maioria das opções são realmente úteis para modificar o comportamento do servidor `ssh` sem mexer em seu arquivo de configuração (para fins de testes) ou para executar um servidor `ssh` pessoal, que deverá ter arquivos de configuração específicos.

Usando aplicativos clientes

Esta seção explicará o uso dos utilitários `ssh`, `scp` e `sftp`.

ssh

Esta é a ferramenta usada para seções de console remotos. O arquivo de configuração de usuários é `~/.ssh/config` e o arquivo global `/etc/ssh/ssh_config`. Para conectar a um servidor ssh remoto:

```
ssh usuario@ip/nome_do_servidor_ssh
```

Caso o nome do usuário seja omitido, seu login atual do sistema será usado. O uso da opção `-C` é recomendado para ativar o modo de compactação dos dados (útil em conexões lentas). A opção `-l usuário` pode ser usada para alterar a identificação de usuário (quando não é usada, o login local é usado como nome de usuário remoto). Uma porta alternativa pode ser especificada usando a opção `-p porta` (a 22 é usada por padrão).

Na primeira conexão, a chave pública do servidor remoto será gravada em `~/.ssh/known_hosts` ou `~/.ssh/known_hosts2` (dependendo da versão do servidor **ssh** remoto, veja “Diferenças nas versões do protocolo”), e verificada a cada conexão como checagem de segurança para se certificar que o servidor não foi alvo de qualquer ataque ou modificação não autorizada das chaves. Por padrão, o cliente utilizará o protocolo ssh versão 1, a opção `-2` permite usar o protocolo versão 2.

Variáveis de ambiente personalizadas para o **ssh** poderão ser definidas no arquivo `~/.ssh/environment`. Comandos que serão executados somente na conexão ssh em `~/.ssh/rc` e `/etc/ssh/sshrc` caso contrário será executado o **xauth** por padrão.

OBS: Para utilizar autenticação Rhosts/Rhosts+RSA (arquivos `~/.rhosts/~/.shosts`) o programa **ssh** deverá ter permissões SUID root e conectará usando portas baixas (menores que 1024).

Exemplos:

```
# Conecta-se ao servidor remoto usando o login do usuário atual
ssh ftp.sshserver.org

# Conecta-se ao servidor remoto usando o login john (via ssh versão 2)
ssh -2 ftp.sshserver.org -l john

# Conecta-se ao servidor remoto usando compactação e o login john
ssh ftp.sshserver.org -C -l john

# Semelhante ao exemplo acima, usando o formato "login@ip"
ssh john@ftp.sshserver.org -C

# Conecta-se ao servidor remoto usando compactação, o login john,
# ativa o redirecionamento do agente de autenticação (-A) e redirecionamento
# de conexões X11 (-X). Veja a próxima seção para entender como o
# suporte a redirecionamento de conexões do X funciona.
ssh ftp.sshserver.org -C -A -X -l john
```

Redirecionamento de conexões do X

O redirecionamento de conexões do X Window poderá ser habilitado em `~/.ssh/config` ou `/etc/ssh/ssh_config` ou usando as opções `-A -X` na linha de comando do **ssh** (as opções `-a` e `-x` desativam as opções acima respectivamente). Uma variável `$DISPLAY` é criada automaticamente para fazer o redirecionamento ao servidor X local.

Ao executar um aplicativo remoto, a conexão é redirecionada a um `DISPLAY` proxy criado pelo ssh (a partir de `:10`, por padrão) que faz a conexão com o display real do X (`:0`), ou seja, ele pulará os métodos de autenticação **xhost** e cookies. Por medidas de segurança é recomendável habilitar o redirecionamento individualmente somente se você confia no administrador do sistema remoto.

```
# Exemplo de configuração do ssh_config
```

```
# Permite Redirecionamento de conexões para o próprio computador (nomes de
# máquinas podem ser especificadas).
Host 127.0.0.1
    ForwardAgent yes
    ForwardX11 yes

# Opções específicas do cliente para conexões realizadas a 192.168.1.4 usando
# somente o protocolo 2
Host 192.168.1.4
    # As 2 linhas abaixo ativam o redirecionamento de conexões do X
    ForwardAgent yes
    ForwardX11 yes
    PasswordAuthentication yes
    Port 22
    Protocol 2
    Cipher blowfish

# Opções específicas do cliente para conexões realizadas a 192.168.1.5 usando
# somente o protocolo 1
Host 192.168.1.5
    # As 2 linhas abaixo desativam o redirecionamento de conexões do X
    ForwardAgent no
    ForwardX11 no
    PasswordAuthentication yes
    Port 22
    Protocol 1
    Cipher blowfish

# CheckHostIP yes
# RhostsAuthentication no
# RhostsRSAAuthentication yes
# RSAAuthentication yes
# FallBackToRsh no
# UseRsh no
# BatchMode no
# StrictHostKeyChecking yes
# IdentityFile ~/.ssh/identity
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_rsa1
# IdentityFile ~/.ssh/id_rsa2
# EscapeChar ~
```

Cliente ssh para Windows

O **putty** é um cliente ssh Win32 que possui suporte aos protocolos versão 1 e 2 do ssh, aceita compactação além de funcionar também como cliente **telnet**. Seu tamanho é pequeno, apenas um executável e requer 220KB de espaço em disco. Ele pode ser baixado de <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

Outra alternativa é o **MindTerm**, este é baseado em Java e pode inclusive ser executado como um applet em uma página web. Este programa é encontrado em <http://www.mindbright.se/mindterm/>.

scp

Permite a cópia de arquivos entre o cliente/servidor ssh. A sintaxe usada por este comando é a seguinte:

```
scp [origem] [destino]
```

Os parâmetros de *origem* e *destino* são semelhantes ao do comando **cp** mas possui um formato especial quando é especificado uma máquina remota:

- Um caminho padrão - Quando for especificado um arquivo local. Por exemplo: `/usr/src/arquivo.tar.gz`.
- `usuario@host_remoto:/diretório/arquivo` - Quando desejar copiar o arquivo de/para um servidor remoto usando sua conta de usuário. Por exemplo: `gleydson@ftp.debian.org:~/arqs`.

A opção `-C` é recomendável para aumentar a taxa de transferência de dados usando compactação. Caso a porta remota do servidor **ssh** seja diferente de 22, a opção `-P porta` deverá ser especificada (é "P" maiúscula mesmo, pois a `-p` é usada para preservar permissões/data/horas dos arquivos transferidos).

Exemplos:

```
# Para copiar um arquivo local chamado /pub/teste/script.sh para
# meu diretório pessoal em ftp.sshserver.org
scp -C /pub/teste/script.sh gleydson@ftp.sshserver.org:~/
```

```
# Para fazer a operação inversa a acima (copiando do servidor remoto para o local)
# é só inverter os parâmetros origem/destino:
scp -C gleydson@ftp.sshserver.org:~/script.sh /pub/teste
```

```
# Para copiar o arquivo local chamado /pub/teste/script.sh para
# o diretório /scripts dentro do meu diretório pessoal em ftp.sshserver.org
# com o nome teste.sh
scp -C /pub/teste/script.sh gleydson@ftp.sshserver.org:~/scripts/teste.sh
```

```
# O exemplo abaixo faz a transferência de arquivos entre 2 computadores remotos:
# O arquivo teste.sh é lido do servidor server1.ssh.org e copiado para
# server2.ssh.org (ambos usando o login gleydson)
scp -C gleydson@server1.ssh.org:~/teste.sh gleydson@server2.ssh.org:~/
```

Cliente scp para Windows

O **pscp** faz a tarefa equivalente ao **scp** no windows, e pode ser baixado de <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

sftp

Permite realizar transferência de arquivos seguras através do protocolo ssh. A conexão e transferências são realizadas através da porta 22 (ainda não é possível modificar a porta padrão). A sintaxe para uso deste comando é a seguinte:

```
sftp usuario@host_remoto
```

Compactação pode ser especificada através da opção `-C`. Um arquivo contendo os comandos usados na seção **sftp** poderá se especificado através da opção `-b arquivo` para automatizar tarefas.

OBS1: Para desativar o servidor **sftp**, remova a linha `Subsystem sftp /usr/lib/sftp-server` (que inicializa o sub-sistema ftp) do arquivo `/etc/ssh/sshd_config` e reinicie o servidor **sshd**.

OBS2: O suporte ao programa **sftp** somente está disponível ao protocolo ssh versão 2 e superiores.

OBS3: Algumas opções comuns do cliente **ftp** padrão (como *mget*) ainda não estão disponíveis ao **sftp**. Veja a página de manual para detalhe sobre as opções disponíveis.

Servidor ssh

sshd

Este é o daemon de controle da conexão encriptada via protocolo ssh, transferência de arquivos e shell interativo. As opções de linha de comando estão disponíveis em “Opções de linha de comando”. Seu arquivo de configuração principal é `/etc/ssh/sshd_config`, um exemplo e descrição das opções deste arquivo é encontrada em “Exemplo de `sshd_config` com explicações das diretivas”.

OBS1: É recomendável que o arquivo `/etc/ssh/sshd_config` seja lido somente pelo dono/grupo, por conter detalhes de acesso de usuários, grupos e intervalo entre a geração de chave de seção.

OBS2: Se estiver ocorrendo falhas no acesso ao servidor ssh, verifique as permissões nos arquivos `/etc/hosts.allow` e `/etc/hosts.deny` (o nome do serviço é **sshd**). Mesmo operando como daemon, o servidor utiliza estes arquivos para fazer um controle de acesso adicional.

Controle de acesso

É definido pelas opções `ListenAddress`, `AllowUsers`, `DenyUsers`, `AllowGroups`, `DenyGroups` e `PermitRootLogin` do arquivo de configuração `sshd_config` (veja “Exemplo de `sshd_config` com explicações das diretivas”) e via `tcpd` (arquivos `hosts.allow` e `hosts.deny`). Veja “O mecanismo de controle de acessos `tcpd`”.

Usando autenticação RSA/DSA - chave pública/privada

Este método de autenticação utiliza o par de chaves pública (que será distribuído nas máquinas que você conecta) e outra privada (que ficará em seu diretório pessoal) para autenticação. A encriptação e decriptação são feitas usando chaves separadas e não é possível conseguir a chave de decriptação usando a chave de encriptação. É possível inclusive gerar uma chave sem senha para efetuar o logon em um sistema ou execução de comandos remotos (este esquema é um pouco mais seguro que os arquivos `~/.rhosts` e `~/.shosts`).

Siga os seguintes passos para se autenticar usando RSA 1 - usada na versão 1 do **ssh**:

1. Gere um par de chaves pública/privada usando o comando:

```
ssh-keygen
```

Um par de chaves RSA versão 1 será gerado com o tamanho de 1024 bits por padrão, garantindo uma boa segurança/performance, e salvas no diretório `~/.ssh` com o nome `identity` e `identity.pub`. Para alterar o tamanho da chave use a opção `-b tamanho`. Depois de gerar a chave, o **ssh-keygen** pedirá uma `frase-senha` (é recomendável ter um tamanho maior que 10 caracteres e podem ser incluídos espaços). Se não quiser digitar uma senha para acesso ao sistema remoto, tecle `<Enter>` quando perguntado. Mude as permissões do diretório `~/.ssh` para 750.

A opção `-f` especifica o diretório e nome das chaves. A chave pública terá a extensão `.pub` adicionada ao nome especificado.

ATENÇÃO Nunca distribua sua chave privada, nem armazene-a em servidores de acesso públicos ou outros métodos que permitem outros terem acesso a ela. Se precisar de uma cópia de segurança, faça em disquetes e guarde-a em um lugar seguro.

2. Instale a chave pública no servidor remoto que deseja se conectar, por exemplo, `www.sshserver.org`:

```
ssh-copy-id -i ~/.ssh/identity gleydson@www.servidorssh.org
```

A função do utilitário acima é entrar no sistema remoto e adicionar a chave pública local `~/.ssh/identity.pub` no arquivo `/home/gleydson/.ssh/authorized_keys` do sistema remoto `www.sshserver.org`. O mesmo processo poderá ser feito manualmente usando os métodos tradicionais (**ssh/scp**). Caso o arquivo remoto `/home/gleydson/.ssh/authorized_keys` não existe, ele será criado. Seu formato é idêntico ao `~/.ssh/known_hosts` e contém uma chave pública por linha.

3. Agora utilize o **ssh** para entrar no sistema remoto usando o método de chave pública/privada. Entre com a senha que usou para gerar o par de chaves público/privado (ele entrará diretamente caso não tenha digitado uma senha).

Para autenticar em uma versão 2 do **ssh** (usando chave RSA 2 ou DSA):

1. Gere um par de chaves pública/privada usando o comando:

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa
```

ou

```
ssh-keygen -t dsa -f ~/.ssh/id_rsa
```

Um par de chaves RSA 2/DSA será gerado. Para alterar o tamanho da chave use a opção `-b tamanho`. Depois de gerar a chave, o **ssh-keygen** pedirá uma `frase-senha` (é recomendável ter um tamanho maior que 10 caracteres e podem ser incluídos espaços). Se não quiser digitar uma senha para acesso ao sistema remoto, tecle <Enter> quando perguntado. Mude as permissões do diretório `~/.ssh` para 750.

ATENÇÃO Nunca distribua sua chave privada, nem armazene-a em servidores de acesso públicos ou outros métodos que permitem outros terem acesso a ela. Se precisar de uma cópia de segurança, faça em disquetes e guarde-a em um lugar seguro.

2. Instale a chave pública no servidor remoto que deseja se conectar copiando o arquivo com:

```
scp ~/.ssh/id_rsa.pub usuario@servidorremoto:~/.ssh/authorized_keys2
ou
scp ~/.ssh/id_dsa.pub usuario@servidorremoto:~/.ssh/authorized_keys2
(caso tenha gerado a chave com a opção -t dsa)
```

Caso o arquivo remoto `/home/gleydson/.ssh/authorized_keys2` não existe, ele será criado. Seu formato é idêntico ao `~/.ssh/known_hosts2` e contém uma chave pública por linha.

3. Agora utilize o **ssh** para entrar no sistema remoto usando o método de chave pública/privada. Entre com a senha que usou para gerar o par de chaves público/privado (ele entrará diretamente caso não tenha digitado uma senha).

OBS: Deverá ser levado em consideração a possibilidade de acesso físico ao seu diretório pessoal, qualquer um que tenha posse de sua chave privada poderá ter acesso ao sistema remoto. O tipo de chave criada por padrão é a *rsa1* (compatível com as versões 1 e 2 do **ssh**). A opção *-t [chave]* poderá ser usada (ao gerar a chave) para selecionar o método de criptografia:

- *rsa1* - Cria uma chave *rsa* compatível com a versão 1 e 2 do **ssh** (esta é a padrão).
- *rsa* - Cria uma chave *rsa* compatível somente com a versão 2 do **ssh**.
- *dsa* - Cria uma chave *dsa* compatível somente com a versão 2 do **ssh**.

Para trocar a senha utilize o comando: `ssh-keygen -p -t tipo_chave -f ~/.ssh/identity` - será pedida sua senha antiga e a nova senha (no mesmo estilo do **passwd**). Opcionalmente você pode utilizar a sintaxe: `ssh-keygen -p -f ~/.ssh/identity -P senha_antiga -N senha_nova`, que troca a senha em um único comando (útil para ser usado em scripts junto com a opção *-q* para evitar a exibição de mensagens de saída do **ssh-keygen**).

Execução de comandos específicos usando chaves

Com o uso de chaves também é possível o uso do **ssh** para execução de comandos específicos em máquinas remotas, isto é possível com os novos recursos da versão 3 do **ssh**. Para fazer isto, siga os passos “Usando autenticação RSA/DSA - chave pública/privada” para gerar um par de chaves *DSA* (o par *RSA* não aceita execução de comandos específicos) e copiar para `authorized_keys2`. Após isto, entre no servidor remoto e edite a chave, inserindo o comando que deverá ser executado antes da linha *dds*, por exemplo:

```
command="ls / -la" ssh-dss ABCAB3NzaC5555MAAACBAL3...
```

Com este método é possível restringir a execução de alguns comandos/serviços além de outras possibilidades como a mudança de variáveis específicas para o comando:

```
no-port-forwarding,no-X11-forwarding,no-agent-forwarding,command="ls / -la" ssh-ds
```

Criando um gateway ssh

Imagine quando você deseja ter acesso a uma máquina de sua rede interna que esteja atrás de um gateway, isto é possível usando os recursos explicados em “Execução de comandos específicos usando chaves” fazendo um redirecionamento de acesso para seu usuário da seguinte forma:

```
command="ssh -t usuario@maquina.interna" ssh-dss DAK874CKLDSAUE83da9x...
```

Isto o acesso do usuário ser redirecionado automaticamente quando efetuar o logon. Caso tenha definido uma senha para a chave *DSA*, o usuário deverá fornecer a senha para entrar no gateway e outra para acessar sua estação de trabalho.

OBS: Não estou levando em conta as considerações de segurança que este exemplo tem em sua rede, bem como o que pode ou não ser redirecionado. A intenção foi manter a simplicidade para entender sem dificuldades como isto é feito.

Criando um tunel proxy

Aplicações remotas podem ser abertas localmente com o uso desta técnica. Você poderá usar para acessar portas que estariam disponíveis somente através do endereço remoto, realizar conexões criptografadas ou com compactação (garantindo uma boa taxa de transferência para protocolos que usem mais texto).

Por exemplo, para redirecionar o tráfego da porta 80 do servidor remoto para a porta 2003 local:

```
ssh -l seu_login servidor -L2003:servidor_remoto:80 -f sleep 60
```

O `sleep 60` tem a função de apenas deixar o tunel aberto por 60 segundos, tempo suficiente para realizarmos nossa conexão. Agora, entre no seu navegador local e acesse a porta 2003:

```
http://localhost:2003
```

A opção `-C` também pode ser especificada junto ao `ssh` para usar compactação dos dados da conexão. Como notou, este recurso também é útil para fazer a administração remota de máquinas, porque o que está realizando a conexão será o IP do servidor remoto, não o seu. Da mesma forma, você poderá ter problemas caso não tenha uma boa política de distribuição de contas de máquinas em sua rede. Veja Capítulo 11, *Gerenciamento de contas e cuidados para a proteção de senhas* para detalhes .

Diferenças nas versões do protocolo

Retirada da página de manual do `sshd`:

Protocolo SSH versão 1

Cada servidor possui uma chave RSA específica (1024 bits por padrão) usada para identifica-lo. Quando o `sshd` inicia, ele gera uma chave RSA do servidor (768 bits por padrão, valor definido por `ServerKeyBits`) que é recriada a cada hora (modificado por `KeyRegenerationInterval` no `sshd_config`) e permanece sempre residente na RAM.

Quando um cliente se conecta o `sshd` responde com sua chave pública da máquina e chaves do servidor. O cliente `ssh` compara a chave RSA com seu banco de dados (em `~/ .ssh/known_hosts`) para verificar se não foi modificada.

Estando tudo OK, o cliente gera um número aleatório de 256 bits, o encripta usando ambas as chaves de máquina e chave do servidor e envia este número ao servidor. Ambos os lados então usam este número aleatório como chave de seção que é usado para encriptar todas as comunicações seguintes na seção.

O resto da seção usa um método de embaralhamento de dados convencional, atualmente Blowfish ou 3DES (usado como padrão). O cliente seleciona o algoritmo de criptografia que será usado de um destes oferecidos pelo servidor. Após isto o servidor e cliente entram em um diálogo de autenticação. O cliente tenta se autenticar usando um dos seguintes métodos de autenticação:

- `~/ .rhosts` ou `~/ .shosts` (normalmente desativada).
- `~/ .rhosts` ou `~/ .shosts` combinado com autenticação RSA (normalmente desativada).
- Autenticação RSA por resposta de desafio.
- Autenticação baseada em senha.

A autenticação usando `Rhosts` normalmente é desativada por ser muito insegura mas pode ser ativada no arquivo de configuração do servidor

se realmente necessário. A segurança do sistema não é melhorada a não ser que os serviços **rshd**, **rlogind**, **rexecd** e **rexrd** estejam desativados (assim, o **rlogin** e **rsh** serão completamente desativados na máquina).

Protocolo SSH versão 2

A versão 2 funciona de forma parecida com a 1: Cada máquina possui uma chave RSA/DSA específica usada para se identificar. A diferença é que quando o **sshd** inicia, ele não gera uma chave de servidor. A segurança de redirecionamento é oferecida através da concordância do uso de uma chave Diffie-Hellman. Esta concordância de chave resulta em uma seção com chave compartilhada. O resto da seção é encriptada usando um algoritmo simétrico, como Blowfish, 3DES, CAST128, Arcfour, 128 bit AES, ou 256 bit AES.

O cliente que seleciona o algoritmo de criptografia que será usado entre os oferecidos pelo servidor. A versão 2 também possui integridade de seção feita através de um código de autenticação de mensagem criptográfica (hmac-sha1 ou hmac-md5). A versão 2 do protocolo oferece um método de autenticação baseado em chave pública (PubkeyAuthentication) e o método de autenticação convencional usando senhas.

Exemplo de `sshd_config` com explicações das diretivas

Abaixo segue um exemplo deste arquivo que poderá ser adaptado ao seu sistema. O objetivo é ser ao mesmo tempo útil para sua configuração e didático:

```
# Modelo personalizado para o guia Foca GNU/Linux baseado na configuração
# original do FreeBSD.
# Autor: Gleydson Mazioli da Silva
# Data: 20/09/2001.

# Porta padrão usada pelo servidor sshd. Múltiplas portas podem ser
# especificadas separadas por espaços.
Port 22

# Especifica o endereço IP das interfaces de rede que o servidor sshd
# servirá requisições. Múltiplos endereços podem ser especificados
# separados por espaços. A opção Port deve vir antes desta opção
ListenAddress 0.0.0.0

# Protocolos aceitos pelo servidor, primeiro será verificado se o cliente é
# compatível com a versão 2 e depois a versão 1. Caso seja especificado
# somente a versão 2 e o cliente seja versão 1, a conexão será descartada.
# Quando não é especificada, o protocolo ssh 1 é usado como padrão.
Protocol 2,1

# As 4 opções abaixo controlam o acesso de usuários/grupos no sistema.
# Por padrão o acesso a todos é garantido (exceto o acesso root se
# PermitRootLogin for "no"). AllowUsers e AllowGroups definem uma lista
# de usuários/grupos que poderão ter acesso ao sistema. Os coringas
# "*" e "?" podem ser especificados. Note que somente NOMES são válidos,
```

```
# UID e GID não podem ser especificados.
#
# As diretivas Allow são processadas primeiro e depois Deny. O método que
# estas diretivas são processadas é idêntico a diretiva
# "Order mutual-failure" do controle de acesso do Apache:
# O usuário deverá TER acesso via AllowUsers e AllowGroups e NÃO ser bloqueado
# por DenyUsers e DenyGroups para ter acesso ao sistema. Se uma das diretivas
# não for especificada, "*" é assumido como padrão.
# Estas permissões são checadas após a autenticação do usuário, porque
# dados a ele pelo /etc/passwd e PAM são obtidos após o processo de
# autenticação.
#AllowUsers gleydson teste?
#DenyUsers root adm
#AllowGroups users
#DenyGroups root adm bin

# Permite (yes) ou não (no) o login do usuário root
PermitRootLogin no

# Chaves privadas do servidor (as chaves públicas possuem um ".pub" adicionado
# no final do arquivo.
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# Tamanho da chave. 768 bits é o padrão
ServerKeyBits 768

# Tempo máximo para login no sistema antes da conexão ser fechada
LoginGraceTime 600

# Tempo para geração de nova chave do servidor (segundos). O padrão é
# 3600 segundos (1 hora).
KeyRegenerationInterval 3600

# Ignora os arquivos ~/.rhosts e ~/.shosts
IgnoreRhosts yes

# Ignora (yes) ou não (no) os arquivos ~/.ssh/known_hosts quando for usado
# para a opção RhostsRSAAuthentication. Se você não confia neste mecanismo
# ajuste esta opção para yes.
IgnoreUserKnownHosts no

# Checa por permissões de dono dos arquivos e diretório de usuário antes de
# fazer o login. É muito recomendável para evitar riscos de segurança
# com arquivos lidos por todos os usuários.
StrictModes yes

# Permite (yes) ou não (no) o redirecionamento de conexões X11. A segurança
# do sistema não é aumentada com a desativação desta opção, outros métodos
# de redirecionamento podem ser usados
X11Forwarding yes

# Especifica o número do primeiro display que será usado para o redirecionamento
```

```
# X11 do ssh. Por padrão é usado o display 10 como inicial para evitar conflito
# com display X locais
X11DisplayOffset 10

# Mostra (yes) ou não (no) a mensagem em /etc/motd no login. O padrão é "no".
PrintMotd no

# Mostra (yes) ou não (no) a mensagem de último login do usuário. O padrão é "no".
PrintLastLog no

# Permite (yes) ou não (no) o envio de pacotes keepalive (para verificar se o
# cliente responde. Isto é bom para fechar conexões que não respondem mas
# também podem fechar conexões caso não existam rotas para o cliente
# naquele momento (é um problema temporário). Colocando esta opção como
# "no" por outro lado pode deixar usuários que não tiveram a oportunidade
# de efetuar o logout do servidor dados como "permanentemente conectados"
# no sistema. Esta opção deve ser ativada/desativada aqui e no programa
# cliente para funcionar.
KeepAlive yes

# Facilidade e nível das mensagens do sshd que aparecerão no syslogd
SyslogFacility AUTH
LogLevel INFO

# Especifica se somente a autenticação via arquivos ~/.rhosts e /etc/hosts.equiv é
# suficiente para entrar no sistema. Não é muito bom usar "yes" aqui.
RhostsAuthentication no

# Mesmo que o acima com o acréscimo que o arquivo /etc/ssh/ssh_known_hosts também
# é verificado. Também evite usar "yes" aqui.
RhostsRSAAuthentication no

# Especifica se a autenticação via RSA é permitida (só usado na versão 1 do
# protocolo ssh). Por padrão "yes".
RSAAuthentication yes

# Permite autenticação usando senhas (serve para ambas as versões 1 e 2 do ssh).
# O padrão é "yes".
PasswordAuthentication yes

# Se a PasswordAuthentication for usada, permite (yes) ou não (no) login
# sem senha. O padrão é "no".
PermitEmptyPasswords no

# Ativa senhas s/key ou autenticação PAM NB interativa. Nenhum destes é
# compilado por padrão junto com o sshd. Leia a página de manual do
# sshd antes de ativar esta opção em um sistema que usa PAM.
ChallengeResponseAuthentication no

# Verifica se o usuário possui emails ao entrar no sistema. O padrão é "no".
# Este módulo também pode estar sendo habilitado usando PAM (neste caso
# cheque a configuração em /etc/pam.d/ssh).
CheckMail no
```

```
# Especifica se o programa login é usado para controlar a seções de shell
# interativo. O padrão é "no".
UseLogin no

# Especifica o número máximo de conexões de autenticação simultâneas feitas
# pelo daemon sshd. O valor padrão é 10. Valores aleatórios podem ser
# especificados usando os campos "inicio:taxa:máximo". Por exemplo,
# 5:40:15 rejeita até 40% das tentativas de autenticação que excedam o
# limite de 5 até atingir o limite máximo de 15 conexões, quando
# nenhuma nova autenticação é permitida.
MaxStartups 10
#MaxStartups 10:30:60

# Mostra uma mensagem antes do nome de usuário/senha
Banner /etc/issue.net

# Especifica se o servidor sshd fará um DNS reverso para verificar se o
# endereço confere com a origem (isto é útil para bloquear conexões
# falsificadas - spoofing). O padrão é "no".
ReverseMappingCheck yes

# Ativa o subsistema de ftp seguro. Para desabilitar comente a linha
# abaixo
Subsystem sftp /usr/lib/sftp-server
```

Capítulo 16. Servidor pop3

Este capítulo descreve a instalação, configuração, criação de contas e controle de acesso ao servidor pop3. Este capítulo é baseado no servidor qpopper da Qualcomm.

Introdução

É o servidor para recebimento de mensagens eletrônicas (e-mails) para o cliente de e-mails. O servidor pop3 documentado é o **qpopper** (da Qualcomm), é um dos mais usados em ambiente **Linux**, simples de configurar e distribuído livremente. O que este programa faz é ler os e-mails de usuários em `/var/mail` e os envia via porta 110 ao programa cliente (**Netscape**, **sylpheed**, **mutt**, **balsa**, **Pegasus**, **Outlook**, ou qualquer outro que suporte o protocolo pop3).

Versão

É assumido que esteja usando a versão 4.0.3 do **qpopper**. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendável que leia a documentação sobre modificações no programa (`changelog`) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

Contribuindo

O site do **qpopper** é <http://www.eudora.com/qpopper/>, anúncios de novas versões, bugs e correções são enviados para `<qpopper-announce@rohan.qualcomm.com>` (inscreva-se enviando uma mensagem com o assunto "subscribe" para o nome da lista acrescentando "-request"). A lista de suporte aos usuários é `<qpopper@lists.pensive.org>` (o método de inscrição é idêntico a lista announce).

Características

- Simples de configurar.
- Possui um timeout padrão de 30 segundos ao invés de 10 minutos do protocolo pop3 padrão.
- O protocolo pop3 é mais simples e consome menos recursos no servidor que o IMAP.
- Suporte a envio de boletins aos usuários do sistema.
- Inclui suporte a TLS/SSL.
- Suporte a senhas ocultas (shadow passwords).
- Suporta PAM.
- Suporte a autenticação via APOP.
- Alta performance.

Ficha técnica

Pacote: .

Utilitários:

- `in.qpopper` - Servidor pop3.
- `popauth` - Manipula os bancos de dados de autorização quando é usado o método de autenticação APOP.

Arquivos de configuração:

- `/etc/popper.allow` - Contém a lista de usuários autorizados a usar o serviço pop3.
- `/etc/popper.deny` - Contém uma lista de usuários NÃO autorizados a usar o serviço pop3.
- `/etc/pop.auth` - Contém dados de autenticação criados pelo programa **popauth**.

Requerimentos de Hardware

O servidor **qpopper** requer no mínimo 6MB de memória para rodar e espaço em disco suficiente para acomodar os e-mails de usuários.

Arquivos de log criados pelo qpopper

Mensagens sobre a execução do **qpopper** são enviadas aos seguintes arquivos em `/var/log`:

- `mail.info` - Detalhes sobre autenticação de usuários e mensagens.
- `mail.warn` - Erros e avisos diversos ocorridos na seção pop3.
- `syslog` e `daemon.log` - Mensagens sobre a execução do servidor qpopper.
- `auth.log` - Mensagens de autenticação gerados pelo PAM.

Instalação

```
apt-get install qpopper
```

Por padrão o servidor qpopper é instalado via inetd:

```
pop-3 stream tcp nowait.60 root /usr/sbin/tcpd /usr/sbin/in.qpopper -s
```

Se estiver configurando um servidor pop3 com um grande número de conexões, é recomendável aumentar o número de execuções do serviço pop3 por minuto (no `inetd.conf`) ou rodar o servidor **qpopper** como daemon (preferido). Para fazer isto, remova a linha que inicia o **qpopper** no `inetd.conf` e construa um script que inicie o serviço como daemon usando a opção `-S` (veja outras opções em “Opções de linha de comando”).

Iniciando o servidor/reiniciando/recarregando a configuração

O serviço é executado por padrão via inetd e utiliza o controle de acesso **tcpd** (veja “O mecanismo de controle de acessos tcpd”). Adicionalmente você pode definir que usuários terão/não acesso ao serviço pop3 nos arquivos `/etc/popper.allow` e `popper.deny`. Por padrão, o acesso é garantido para qualquer usuário.

Após instalar o servidor pop3 instalado, resta configurar o cliente para conectar ao servidor pop3 do servidor. O nome de usuário e senha são os usados no arquivo `/etc/passwd`.

Teste de acesso no pop3

Um simples teste consiste em usar o **telnet** conectando a porta pop3 (110): `telnet 127.0.0.1 110:`

```
Connected to 127.0.0.1.  
Escape character is '^]'.  
+OK Qpopper (version 4.0.3) at server.org starting. [2122.11132222@server.org]
```

A resposta acima indica que o servidor pop3 está funcionando corretamente.

Opções de linha de comando

Opções de linha de comando do servidor **in.qpopper**:

- | | |
|----------------|--|
| endereço:porta | Quando está operando em modo daemon (iniciado com <code>-S</code>), espera por conexões no <i>endereço</i> e opcionalmente na <i>porta</i> especificada. O endereço deverá ser o da interface de rede local do servidor (como <code>192.168.1.1</code>) caso não seja especificado, o servidor qpopper monitorará todos os endereços. A porta padrão é 110 caso não seja especificada. |
| -b [diretório] | Ativa o sistema de envio de boletins. O diretório especificado é o que contém os boletins que serão enviados (na distribuição Debian , o <code>/var/spool/popbull</code> é o indicado). Veja “Enviando boletins de mensagens” para instruções de utilização deste recurso. |
| -c | Modifica a senha para caracteres minúsculos antes de autenticar, permitindo que clientes conectem com a senha em MAIÚSCULAS ou caracteres misturados. |
| -f [arquivo] | Especifica um arquivo de configuração para o servidor qpopper. Veja a página de manual para detalhes sobre as opções. Recomendo usar as opções de linha de comando exceto se for requerida configurações especiais para modificar o comportamento do servidor pop3. |
| -l [num] | Modifica as opções de criptografia TLS/SSL usada no transporta da seção. Os seguintes valores são aceitos: <ul style="list-style-type: none">• 0 - Desativa TLS/SSL. É o padrão.• 1 - Ativa o suporte a TLS/SSL. Se o cliente não suportar criptografia, os dados serão transmitidos usando a forma padrão.• 2 - Tenta ativar uma conexão TLS quando o cliente conecta ao servidor usando uma porta alternativa. |
| -p [num] | Seleciona como a senha em texto plano será manipulada. O servidor deverá estar compilado com suporte a outras formas de autenticação (como APOP) ao invés de texto plano. As seguintes opções são suportadas. <ul style="list-style-type: none">• 0 - Senhas em texto plano podem ser usadas para usuários não cadastrados no arquivo <code>/etc/pop.auth</code> (gerenciado pelo popauth. Este é o padrão. |

- 1 - Somente permite acesso de usuários cadastrados no arquivo `/etc/pop.auth`. Qualquer acesso usando texto plano é negado.
 - 2 - Permite autenticação usando texto plano como preferência, até mesmo para usuários que estejam no `/etc/pop.auth`. É útil para clientes que não suportam autenticação usando texto plano.
 - 3 - Somente usuários conectando da mesma máquina (127.0.0.1) podem usar autenticação em texto plano.
 - 4 - Permite autenticação usando texto plano somente se uma conexão criptográfica foi estabelecida usando TLS ou SSL.
- R Desativa a resolução reversa de endereços IP de clientes.
- s Registra dados de início da seção, nome de usuário, número de bytes/mensagens apagadas, número de mensagens deixadas no servidor e fim da seção. Estes detalhes são registrados pelo **syslogd**. Seu uso é recomendável para ter controle sobre o que está acontecendo em seu servidor.
- S Ativa o modo daemon. Útil para servidores pop3 com grande número de acessos.
- T [num] Tempo máximo em segundos para finalização da seção quando o cliente não envia nenhuma resposta ou comando. Valores pequenos (como 20) podem ser especificados para servidores que possuem poucos usuários e um link rápido. Para grande quantidade de usuários ou conexão feita via links lentos (como ppp, slip, plip, etc.) use valores como 600 (10 minutos) ou mais. O valor padrão é 120 segundos (2 minutos).
- u Lê o arquivo `~/ .qpopper.options` no diretório do usuário em busca de opções adicionais para o servidor. Este arquivo é lido após o processo de autenticação e deve ter permissões leitura/gravação para o dono. Isto não é recomendável em servidores seguros, a criptografia ou método de autenticação podem ser desativados sem o conhecimento do administrador comprometendo a segurança dos dados.
- U Idêntica a opção acima, mas o arquivo deve residir no diretório de spool (`/var/spool/pop`) e ter o formato:
- ```
.usuario.qpopper-options
```
- Este arquivo deve ter como dono o administrador ou dono do servidor pop3. Esta alternativa é mais segura que a anterior porque o usuário não terá acesso ou desativar opções específicas.
- y [facilidade] Permite modificar o nível *facilidade* que as mensagens são registradas no **syslogd** (veja “Arquivo de configuração `syslog.conf`”).

## Enviando boletins de mensagens

Este recurso é muito útil para enviar alertas ou avisos para todos os usuários em seu sistema de uma só vez. A mensagem é escrita no diretório `/var/spool/popbull` seguindo um formato especial e quando o usuário pop3 se conecta para pegar seus e-mails receberá também uma cópia do boletim. O controle de

boletins já recebido pelo usuário é feito no arquivo `~/ .popbull`. Siga os passos a seguir para configurar este sistema:

1. Ative o suporte a envio de boletins no servidor `qpopper`, adicionando a opção `-b /var/spool/popbull` a linha de comando.
2. Os números de boletins são controlados seqüencialmente pelos arquivos `~/ .popbull`, portanto é importante começar com o nome do boletim com pelo menos 5 dígitos (00001, 00002, 00003, etc). Vamos usar `00001-teste` em nosso exemplo.
3. A primeira linha do boletim deve conter a palavra "From" e um espaço e deve ser completada com um nome e data, seguido de campos essenciais para o envio da mensagem:

```
From teste Sex Set 29 21:40:00 2001
To: user@localhost
From: Administrador do Sistema <root@localhost>
Date: Fri, 29 Sep 2001 21:40:00 -0800 (PST)
Subject: Teste do sistema de boletins
```

Este é apenas um teste para o sistema de boletins. Se tudo estiver OK você receberá esta mensagem quando pegar seus e-mails no cliente pop3 e este boletim será registrado no arquivo `~/ .popbull` para que não seja novamente recebido.

Deve haver uma linha em branco para separar o cabeçalho da mensagem.

**OBS:** Quando incluir novos usuários no sistema, somente os últimos 10 boletins serão enviados.

## Especificando quotas para as caixas de correio

Crie o diretório de spool `/var/mail` em uma partição separada e ative o sistema de quota do **Linux** nela. Leia as instruções em "Limitando o uso de espaço em disco (quotas)".

## Restringindo acesso ao servidor pop3

O controle de acesso de conexões é feito via método `tcpd` usando o daemon `in.qpopper` (veja "O mecanismo de controle de acessos tcpd"). O controle de acesso dos usuários é feito através dos arquivos `/etc/popper.allow` e `/etc/popper.deny`, respectivamente contém os nomes de usuários que podem e não podem ter acesso ao servidor `qpopper`. Por motivos de segurança é recomendável redirecionar os e-mails do usuário `root` para outra conta (no arquivo `/etc/aliases` e bloquear o acesso do usuário `root` ao pop3 no arquivo `/etc/popper.deny`.

Se a máquina servidora pop3 não for utilizada para acesso remoto, é recomendável desativar os serviços de login (veja "Desabilitando serviços de shell para usuários").

---

# Capítulo 17. CVS

Este capítulo explica os requerimentos, instalação, configuração, segurança e diversos modelos de configuração de acesso para trabalho em grupo utilizados pelo CVS.

Não tome-o como uma referência completa ao uso e configuração do cvs, a pesquisa de sua info page é muito importante.

## Introdução ao CVS

O CVS (*Concurrent Version Software*) permite que se organizem grupos de trabalho para desenvolvimento de projetos colaborativos. Um projeto pode ser desde um programa em C, documentação em equipe, etc. O uso do CVS é recomendado para qualquer desenvolvimento de projeto que tenha vários envolvidos trabalhando ao mesmo tempo.

Para cada mudança feita no programa, é pedido uma descrição dos trabalhos realizados e o sistema registra todas as modificações realizadas ao longo do desenvolvimento, permitindo voltar a uma versão anterior ou ver as mudanças entre elas facilmente.

Imagine uma situação onde você está desenvolvendo um programa de computador e após a última modificação ele para de funcionar. Com o CVS é possível ver o que foi modificado e voltar até a versão que estava funcionando para consertar o problema. No desenvolvimento de documentação e tradução o CVS também desempenha um papel importante, pois com ele o tradutor pode ver o que foi modificado entre a versão do documento original que ele usou para tradução e uma versão recente, traduzindo apenas as diferenças.

Uma seção de **cvs** é feita de modo interativo através do comando **cvs**. Por exemplo:

- `logar no sistema - cvs login`
- `baixar um projeto - cvs checkout projeto`

Cada comando do **cvs** será explicado em detalhes no decorrer deste capítulo.

## Versão

A versão do CVS documentada no guia é a *1.11.1*. As explicações aqui certamente serão compatíveis com versões posteriores deste programa.

## História

O CVS é uma substituição do sistema RCS (Revision Control System) ele possui mais recursos e foi criado sendo compatível com o RCS.

A história do CVS (extraída de sua info page) é que ele foi iniciado a partir de um conjunto de scripts shell escritos por *Dick Grune* que foram postados ao grupo de notícias `comp.sources.unix` no volume 6 de Dezembro de 1986. Na versão atual não estão mais presentes shell scripts porque muitos dos conflitos de resolução de algoritmos vem deles.

Em Abril de 1989, *Brian Berliner* fez o design e programou o CVS. Mais tarde, Jeff Polk ajudou Brian com o design do módulo CVS.

## Contribuindo com o CVS

Através da lista de discussão `info-cvs`. Para se inscrever envie uma mensagem com o subject "subscribe" para `info-cvs-request@gnu.org`. Outra alternativa é através do grupo de notícias (newsgroup) da Usenet `comp.software.config-mgm`.

## Características

Abaixo uma lista de características que tornam o CVS útil no gerenciamento de trabalhos em grupo:

- Gerenciamento de projeto em equipe
- Log de todas as alterações realizadas
- Lock de arquivos, permitindo que somente uma determinada pessoa modifique o arquivo durante o desenvolvimento do projeto.
- Histórico de todas as mudanças feitas, isto permite voltar a uma versão anterior em caso de problemas, e ver o que houve de errado com o código.
- Os projetos podem ser hospedados em repositórios.
- Podem ser criados diversas equipes de trabalho para cada repositórios, e definidos quem terá ou não acesso ao repositório individualmente. O desenvolvedor `gleydson`, por exemplo, podem ter acesso ao projeto `x_beta` e não ter acesso a projeto `secret_y`.
- Permissões de acesso individuais de leitura/gravação.
- É possível criar um usuário com acesso anônimo sem dar uma conta no sistema.
- Pode tanto utilizar o banco de dados de contas/senhas do sistema como um banco de dados de autenticação do próprio CVS.
- Permite utilizar diversos "métodos" de acesso ao servidor: *local*, *pserver*, *ext*, etc. Cada um destes métodos será descrito a seguir.
- Permite o acesso via ssh para usuários que já possuam conta na máquina servidora. Este método garante segurança no envio da senha criptografada (veja "Sniffer" para detalhes).
- Permite visualizar facilmente o que foi modificado entre duas versões de um arquivo.

**OBS:** O CVS possui algumas limitações e falhas, uma delas que mais me faz falta é um suporte a protocolo *pserver* via ssh que resolveria o problema de tráfego em texto plano e gerenciamento de grupos com permissões diferenciadas.

## Ficha técnica

Pacote:

Utilitários:

- `cvs` - Servidor/ferramenta cliente.
- `cvsbug` - Envia um bug sobre o CVS para a equipe de suporte.

- `rsc2log` - Converte arquivos de log do formato usado pelo **RCS** para o **CVS**. Utilizado na migração desta ferramenta para o **CVS**.
- `cvsconfig` - Usado pela **Debian** para ativar/desativar o servidor `pserver`. Pode também ser usado o `dpkg-reconfigure cvs` para desativar o servidor `pserver` e suas características.
- `cvs-makerepos` - Script da **Debian** que lê a lista de repositórios de `/etc/cvs-pserver.conf`, cria os repositórios no local apropriado, corrige as permissões do diretório e adiciona os repositórios no servidor `pserver`.
- `cvs-pserver` - Script da **Debian** responsável por fazer uma inicialização mais inteligente do servidor de **CVS** via `pserver`, leitura e processamento de repositórios, etc. Normalmente ele é chamado a partir do arquivo `/etc/inetd.conf`.

## Requerimentos de Hardware

Para executar o **CVS** é requerido pelo menos 3 vezes mais memória que o tamanho do maior arquivo usado pelo projeto (para realização de diffs entre as atualizações) e uma boa quantidade de espaço em disco.

Na realidade os requerimentos sobre o **CVS** dependem muito da aplicação que será desenvolvida. É recomendável que a máquina tenha memória suficiente para evitar o uso de swap, que degrada bastante a performance do sistema.

## Arquivos de log criados pelo CVS

Problemas na inicialização do **CVS** são registrados no arquivo `/var/log/daemon.log`. Os logs de modificações feitas nos arquivos de um projeto no **CVS** são armazenadas no formato `arquivo.extensão,v` (é adicionado o ",v" ao final do arquivo para indicar que é um arquivo de controle de modificações do **CVS**).

## Instalação

O **CVS** pode ser baixado de <http://www.cvshome.org/>.

Para pacotes **Debian** basta apenas executar o comando: `apt-get install cvs` e seguir as telas de configuração para ter o pacote **CVS** instalado e (opcionalmente) com o servidor sendo executado. Você poderá a qualquer momento reconfigurar o **CVS** executando: `dpkg-reconfigure cvs`.

Uma boa documentação de referência é encontrada no pacote .

## Iniciando o servidor/reiniciando/recarregando a configuração

A única configuração requerida é quando o **CVS** é executado via `pserver`. Para isto, é necessária a seguinte linha no arquivo `/etc/inetd.conf`

```
cvspserver stream tcp nowait.200 root /usr/sbin/tcpd /usr/sbin/c
```

Note que o parâmetro "200" indica quantas vezes o processo **CVS** poderá ser executado por minuto no sistema. Caso esse número seja excedido, o serviço será desabilitado e será necessário reiniciar o servidor **inetd** com o comando `killall -HUP inetd` para reativar o servidor **CVS** `pserver` (veja `"/etc/inetd.conf"` capítulo do **inetd** para detalhes). Ajuste este valor de forma adequada ao seu servidor!

Veja o script `cvs-pserver` sendo executado no final da linha. Ele foi desenvolvido para lidar de forma mais inteligente com a configuração do servidor CVS `pserver`.

## Opções de linha de comando

As seguintes opções são aceitas pelo CVS.

|                               |                                                                                                                                                                                                                                               |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-z [num]</code>         | Utiliza o <code>gzip</code> para fazer a transferência compactada dos arquivos. O valor especificado pode ser de 0 a 9, quanto maior o número maior o nível de compactação e uso da CPU.<br><br>Exemplo: <code>cvs -z 3 checkout teste</code> |
| <code>-q</code>               | Ocultar mensagens sobre recursão de diretório durante os comandos do CVS.                                                                                                                                                                     |
| <code>-d [repositório]</code> | Permite especificar o repositório através da linha de comando.                                                                                                                                                                                |
| <code>-e [editor]</code>      | Define qual é o editor de textos usado para registrar o texto de <code>commits</code> .                                                                                                                                                       |
| <code>-n</code>               | Executa o <code>cvs</code> em modo "simulação" não modificando qualquer arquivo do repositório.                                                                                                                                               |
| <code>-t</code>               | Mostra mensagens mostrando o processo de execução de comandos do CVS. É bastante útil para aprendizado do <code>cvs</code> usado junto com a opção <code>-n</code> .                                                                          |
| <code>-r</code>               | Torna os novos arquivos criados somente para leitura. É a mesma coisa que especificar a variável <code>CVSREAD</code> .                                                                                                                       |
| <code>-w</code>               | Torna os novos arquivos criados leitura/gravação que é o padrão.                                                                                                                                                                              |
| <code>-x</code>               | Utiliza criptografia para a transferência dos arquivos quando é utilizado em conjunto com o <b>Kerberos</b> .                                                                                                                                 |

Você pode obter detalhes sobre opções sobre um comando em especial do CVS (`commit`, `checkout`, etc) digitando: `cvs comando --help`. Veja "Criando projetos para serem usados no CVS" para exemplos sobre cada uma delas.

## Servidor de CVS - configurando métodos de acesso ao repositório

O CVS é uma aplicação cliente/servidor, possuindo diversas maneiras de fazer o acesso seu repositório (veja "Repositório" repositórios). Estes métodos são os seguintes:

- `local` ("local").
- `ext` ("ext").
- `pserver` ("pserver (password server)").
- `fork` ("fork").
- `GSSAPI` ("gssapi").

Eles são explicados em detalhes nas sub-seções a seguir.

## local

Acessa o diretório do repositório diretamente no disco local. A vantagem deste método é que não é requerido nem nome nem senha para acesso (você precisa apenas ter permissões para acesso aos arquivos que deseja trabalhar) e também não é preciso nenhuma conexão de rede.

Este método é ideal para trabalhar na máquina local ou com os arquivos administrativos do CVS existentes no diretório `CVSROOT` do repositório. É muito útil também para configurar outros métodos de acesso, como o `pserver`.

Para criar seu repositório, veja “Criando um repositório”.

## Configurando o método local

Para utilizar o método de acesso local, basta definir a variável `CVSROOT` da seguinte forma (assumindo que o repositório esteja instalado em `/var/lib/cvs`):

```
export CVSROOT=/var/lib/cvs
```

ou

```
export CVSROOT=local:/var/lib/cvs
```

Depois disso, basta utilizar os comandos normais do  `cvs`  sem precisar se autenticar no sistema. Veja os detalhes de utilização dos comandos de CVS após o login na seção “Clientes de CVS”.

## fork

Este método é semelhante ao local, mas ele "simula" uma conexão de rede com o servidor. É muito usado para fins de testes.

## Configurando o método fork

Para utilizar o método de acesso *fork*, basta definir a variável `CVSROOT` da seguinte forma (assumindo que o repositório esteja instalado em `/var/lib/cvs`):

```
export CVSROOT=fork:/var/lib/cvs
```

Depois disso, basta utilizar os comandos normais do  `cvs` , sem precisar se autenticar no sistema. Veja os detalhes de utilização dos comandos do CVS após o login em “Clientes de CVS”.

## ext

Este método de acesso lhe permite especificar um programa externo que será usado para fazer uma conexão remota com o servidor  `cvs` . Este programa é definido na variável `CVS_RSH` e caso não ela seja especificada o padrão é `rsh`.

Este método requer que o usuário possua um login/senha no banco de dados de autenticação `/etc/passwd` do servidor de destino. Suas permissões de acesso ao CVS (leitura/gravação) serão as mesmas definidas neste arquivo.

O uso do acesso criptografado via **ssh** é possível definindo o programa **ssh** na variável *CVS\_RSH*. Veja os exemplos a seguir em “Configurando o método ext”.

Para criar seu repositório, veja “Criando um repositório”.

## Configurando o método ext

Defina a variável *CVSROOT* da seguinte forma para utilizar este método de acesso (assumindo */var/lib/cvs* como repositório):

```
export CVSROOT=:ext:conta@servidor.org.br:/var/lib/cvs
cvs login
```

A "conta" é uma conta de usuário existente no servidor remoto (por exemplo, *gleydson*) seguido do nome do servidor remoto (separado por uma "@"). Por exemplo para acessar o servidor *cvs.cipsga.org.br* usando a conta *michelle*:

```
export CVSROOT=:ext:michelle@cvs.cipsga.org.br:/var/lib/cvs
cvs checkout
```

OBS: A senha via método de acesso "ext" será pedida somente uma vez quando for necessário o primeiro acesso ao servidor remoto. Veja os detalhes de utilização dos comandos de CVS após o login na seção “Clientes de CVS”. O uso mais freqüente do **ext** é para conexões seguras feitas via **ssh**, feita da seguinte forma:

```
export CVS_RSH=ssh
export CVSROOT=:ext:michelle@cvs.cipsga.org.br:/var/lib/cvs
cvs checkout
```

O acesso de leitura/gravação do usuário, é definido de acordo com as permissões deste usuário no sistema. Uma maneira recomendada é definir um grupo que terá acesso a gravação no CVS e adicionar usuários que possam fazer gravação neste grupo.

**OBS1:** O acesso via **ssh** traz a vantagem de que as senhas trafegarão de forma segura via rede, não sendo facilmente capturadas por sniffers e outros programas de monitoração que possam estar instalados na rota entre você e o servidor.

**OBS2:** É possível especificar a senha na variável *CVSROOT* usando a sintaxe semelhante a usada no ftp:

```
export CVSROOT=:ext:michelle:senha@cvs.cipsga.org.br:/var/lib/cvs
```

Entretanto isto não é recomendado, pois os processos da máquina poderão capturar facilmente a senha (incluindo usuários normais, caso a máquina não esteja com patches de restrições de acesso a processos configurada, que é o padrão em quase todas as distribuições de **Linux**).

## pserver (password server)

Este é um método de acesso remoto que utiliza um banco de dados de usuários senhas para acesso ao repositório. A diferença em relação ao método de acesso *ext* é que o *pserver* roda através de um servidor próprio na porta 2401. O acesso dos usuários (leitura/gravação) no repositório pode ser feita tanto através

do banco de dados de usuários do sistema (`/etc/passwd`) como através de um banco de dados separado por repositório.

A grande vantagem deste segundo método é que cada projeto poderá ter membros com acessos diferenciados; o membro `x` poderá ter acesso ao projeto `sgml` mas não ao projeto `focalinux`; ou o usuário `y` poderá ter acesso de gravação (para trabalhar no projeto `focalinux`) mas somente acesso de leitura ao projeto `sgml`.

Este é o método de acesso preferido para a criação de usuários anônimos (uma vez que o administrador de um servidor que hospede muitos projetos não vai querer abrir um acesso anônimo via `ext` para todos os projetos).

Também existe a vantagem que novos membros do projeto e tarefas administrativas são feitas por qualquer pessoa que possua acesso de gravação aos arquivos do repositório.

## Configurando um servidor pserver

### Ativando o servidor pserver

Para ativar o `pserver` (caso ainda não o tenha feito). Execute o comando `dpkg-reconfigure cvs` e selecione a opção `Ativar o servidor pserver`. Uma maneira de fazer isso automaticamente é modificando o arquivo `/etc/inetd.conf` adicionando a seguinte linha:

```
na Debian
cvspserver stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/c

em outras Distribuições
cvspserver stream tcp nowait root /usr/bin/cvs cvs -f --allow-root=/var/lib/c
```

Na **Debian**, o `cvs` é iniciado através do script `/usr/sbin/cvs-pserver` que checa os binários e executa o `cvs` para todos os repositórios especificados no arquivo `/etc/cvs-pserver.conf`.

Caso precise adicionar mais repositórios para acesso via `pserver` ou outro método de acesso, veja “Criando um repositório”.

Você também poderá executar o método `pserver` sob um usuário que não seja o `root`, para isto, modifique a entrada referente ao usuário.grupo no `inetd.conf` e tenha certeza que o daemon consegue fazer as operações de `suid/sgid` no diretório onde o repositório se encontra.

### Servidor pserver usando autenticação do sistema

Para usar o banco de dados de autenticação do sistema (`/etc/passwd`) para autenticar os usuários remotos, primeiro tenha certeza que o servidor `pserver` está ativado (como descrito em “Ativando o servidor pserver”). Repetindo o exemplo anterior, a usuária `Michelle` deverá ter uma conta em `/etc/passwd` para fazer acesso ao `cvs`:

```
export CVSR00T=:pserver:michelle@cvs.cipsga.org.br:/var/lib/cvs
cvs login
```

Será pedido a senha da usuária `michelle`. Entrando com a senha correta, o sistema retornará para o aviso de comando. Uma mensagem será mostrada caso a senha entrada seja incorreta. Daqui em diante, o resto da seção `CVS` é normal e você terá as permissões de acesso ao repositório de acordo com as suas permissões de acesso naquele diretório.

**OBS1:** A senha poderá ser passada junto com o login da mesma forma como o ftp. Veja a observação em “Configurando o método ext”.

**OBS2:** A desvantagem do método pserver padrão é que a seção é feita em texto plano, desta forma, alguns cuidados podem ser tomados para tornar o sistema um pouco mais seguro. Um deles é dar `/bin/false` como shell de usuário (para desativar o login no sistema) ou usar o método de acesso descrito em “Servidor pserver usando autenticação do sistema” em combinação com este. Tenha consciência das influências disso se a máquina for usada para outras tarefas, como um servidor "pop3" por exemplo.

## Servidor pserver com autenticação própria

Esta forma de acesso armazena os usuários em um banco de dados próprio, não requerendo a criação de contas locais no arquivo `/etc/passwd`. Para criar um servidor deste tipo siga os seguintes procedimentos:

1. Exporte a variável `CVSROOT` apontando para o repositório que deseja configurar o acesso. Como isto é uma configuração administrativa, assumo o método de acesso *local* sendo usada pelo usuário administrador do servidor: `export CVSROOT=/var/lib/cvs`.
2. Crie um diretório para trabalhar nos arquivos administrativos do repositório: `mkdir /tmp/repos`
3. Entre no diretório criado acima e execute o comando: `cvs checkout .`
4. Quando terminar de baixar os arquivos, entre no subdiretório `CVSROOT`, os arquivos de configuração do repositório se encontram lá (para detalhes sobre cada um destes arquivos, veja “Arquivos administrativos em CVSROOT”).
5. Edite o arquivo `config` e mude a variável `SystemAuth` para `no`. Isto diz ao servidor *pserver* não usar os arquivos de autenticação do sistema, mas a invés disso usar seu banco de dados próprio.

Em algumas instalações, caso exista o arquivo `passwd` no repositório, o *pserver* automaticamente o utiliza ao invés do `/etc/passwd`.

6. Crie um arquivo `passwd` no diretório `CVSROOT` o formato deste arquivo é:

```
usuario:senha:usuario_local
```

Onde:

|               |                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| usuario       | Nome da conta de usuário que fará acesso ao CVS.                                                                                                                                                                                                                                                                                                            |
| senha         | Senha que será usada pelo usuário. Ela deverá ser criptografada usando o algoritmo crypt. O comando <b>mkpasswd senha</b> pode ser usado para gerar a senha criptografada. Caso este campo seja deixado em branco, nenhuma senha de usuário será utilizada. O utilitário <b>mkpasswd</b> está presente no pacote na <b>Debian</b> .                         |
| usuario_local | Usuário local que terá suas permissões mapeadas ao usuário do CVS. Como a conta de usuário do <b>cvs</b> não existe no sistema, é necessário que o sistema tenha uma maneira de saber que nível de acesso este usuário terá. Caso não crie este usuário ou ele seja inválido, você terá erros do tipo "": no such user" no momento que fizer o "cvs login". |

Uma forma segura de se fazer isto, é criar uma conta de usuário *\*somente\** com acesso aos arquivos do CVS, sem shell e senha. Isto permitirá mapear a UID/

---

GID do usuário criado com o acesso do CVS sem comprometer a segurança do sistema de arquivos. Isto pode ser feito através do seguinte comando:

```
adduser --disabled-password --disabled-login usuario
```

É necessário especificar um diretório home do usuário, pois o servidor cvs precisa ter acesso ao arquivo `/home/do/cvs/.cvsignore`.

**OBS1:** Mais uma vez: Leve sempre em conta a forma que os outros serviços em sua máquina estão configurados (como eles fazem acesso, permissões de acesso, diretórios onde gravam arquivos, são algumas delas) antes de escolher como um serviço novo na máquina funcionará. Isto poderá modificar ou deixar vulnerável a segurança de sua instalação.

**OBS2:** Permita que os usuários **somente** tenham acesso a máquina via CVS.

**OBS3:** Certifique-se sempre que o dono/grupo do repositório seja `root.src` (ou outro grupo que tenha criado) adicione somente usuários de confiança no grupo `src` para criar novos projetos.

Exemplos:

```
gleydsonm:k32dk1234k:cvsuser
anonymous::pooruser
```

O usuário `cvs gleydsonm` quando logar no cvs, terá as permissões de acesso do usuário `cvsuser` do sistema.

**OBS1:** Certifique-se que o usuário local possui permissões de gravação no diretório do CVS, caso contrário ele não poderá fazer *commits*. Lembre-se que as permissões de leitura/gravação do usuário serão controladas através de arquivos do próprio `pserver`, mas também é necessária a permissão de gravação do usuário no repositório. Isto poderá ser feito através de grupos de sistema e garante uma dupla camada de segurança.

**OBS2:** Caso tenha preferido usar o `pserver` sob um usuário diferente de `root` e esteja obtendo a mensagem `setgid failed: Operation not permitted`, significa que o servidor CVS não consegue mudar para o grupo referente ao usado no diretório do repositório. Verifique se as permissões estão adequadas e se o grupo do usuário CVS no `/etc/passwd` é o mesmo que especificou para acesso ao repositório.

7. Para dar direito de leitura ao repositório, crie um arquivo chamado `readers` e adicione os nomes de usuários que terão acesso ao repositório (um por linha). O nome que deverá ser usado é o nome do usuário de CVS e não do sistema (usuário `gleydsonm`, segundo o exemplo).

Exemplo:

```
gleydsonm
anonymous
```

8. Para dar direito de gravação ao repositório, crie um arquivo chamado `writers`. Seu formato é idêntico ao arquivo `readers`.

Exemplo:

```
gleydsonm
macan
otavio
hmh
kov
```

9. Pronto, o acesso a CVS usando um banco de dados próprio está pronto! basta dar o commit nos arquivos, adicionar os arquivos `readers`, `writers` e `passwd` no repositório (veja “Adicionando um arquivo ao módulo CVS do servidor”) para o servidor de CVS para te-lo funcionando. Note que em versões mais novas do CVS, não é possível transferir o arquivo `passwd` via rede, então será necessário criá-lo manualmente dentro do repositório do servidor.

**OBS:** O arquivo `passwd` não é transferido pelo commit por motivos de segurança, pois ele contém senhas que podem ser capturadas e usada por pessoas maliciosas. Será necessário transferi-lo manualmente para o repositório do servidor remoto (você terá que ser o usuário `root` ou ter permissões adequadas). O recomendável é utilizar o `scp` (“scp”) para realizar transferências seguras. .

O método de acesso do CVS aos arquivos `readers` e `writers` é restritiva, portanto se um nome de usuário existir no arquivo `readers` e `writers` o que valerá será o menor nível de acesso. Vendo os exemplos acima, os usuários `gleydsonm` e `anonymous` terão somente acesso a leitura do repositório e `macan`, `otavio`, `hmh`, `kov` acesso de leitura e gravação.

## gssapi

Quando o CVS é compilado com o suporte a Kerberos 5, ele tenta estabelecer automaticamente uma conexão segura usando este método. Este método funciona somente se o CVS estiver compilado com o suporte a Kerberos (opção `--with-gssapi`).

## Criando projetos para serem usados no CVS

Esta seção descreve todos os passos necessários para colocação de um projeto para ser desenvolvido através do CVS, os comandos do `cvs`, considerações a respeito dos comandos e sua utilização através de exemplos didáticos.

## Repositório

Um repositório CVS é o local que armazena módulos e também os arquivos administrativos (que contém permissões, etc) são armazenados em um subdiretório chamado `CVSROOT`.

O acesso a um repositório é feito através de parâmetros especificados na variável `CVSROOT` ou pela opção `-d repositório` do `cvs`. Veja “Servidor de CVS - configurando métodos de acesso ao repositório” para ver exemplos de métodos de acesso.

O Repositório pode conter um ou mais módulos, cada módulo representa um projeto no servidor, criado após o uso do comando `import`. Segue um exemplo da estrutura de um repositório CVS:

```
var/lib
|
+- cvs
```

```
|- CVSROOT
|- projeto1
+- projeto2
```

O subdiretório `cv`s é o repositório (veja o subdiretório `CVSROOT` dentro dele) e os diretórios dentro dele `projeto1` e `projeto2` são os módulos criados através do comando `cv`s `import ...` (veja “Adicionando um novo projeto”).

Para acessar o projeto do CVS, então é definido o repositório que tem permissões de acesso na variável `CVSROOT` e então é executado um comando (`checkout`, `update`, `commit`, etc) no módulo que desejamos utilizar:

```
export CVSROOT=:ext:anonymous@servidor.org.br:/var/lib/cvs (<- Repositório "cv")
cv
```

Nas seções seguintes serão explicados cada um dos comandos usados para trabalhar com um projeto no `cv`s.

## Criando um repositório

Para adicionar um novo repositório no sistema, edite o arquivo `/etc/cvs-pserver.conf` e defina o nome de cada repositório na variável `CVS_PSERV_REPOS` separados por ":" (exemplo: `CVS_PSERV_REPOS="/var/lib/cvs:/var/lib/cvs2"`).

Feito isso execute o comando **cv**s-**makerepos** para que os diretórios especificados no arquivo `/etc/cvs-pserver.conf` sejam criados com as devidas permissões.

Para adicionar manualmente um repositório (`/var/lib/cvs`), execute os seguintes passos:

1. Execute o comando `cv`s `-d /var/lib/cvs init` (para criar o repositório e os arquivos administrativos que ficam armazenados dentro de `CVSROOT`).
2. Mude as permissões do diretório para `sgid` com: `chmod 2775 /var/lib/cvs`.
3. Mude o dono/grupo com o comando: `chown root.src /var/lib/cvs`
4. Opcional: caso utilize o método de acesso `pserver` será necessário adicionar a opção `--allow-root=/var/lib/cvs` na linha que inicia o servidor `pserver`. Este parâmetro deve ser usada para cada repositório adicionado no servidor.

A partir de agora, seu repositório já está pronto para ser utilizado.

## Logando no servidor de CVS via pserver

Quando é usado o método de acesso `pserver` (“`pserver` (password server)”), é necessário fazer para ter acesso ao repositório. Por exemplo, para acessar o repositório `/var/lib/cvs` no servidor `servidor.org.br`:

```
export CVSROOT=:pserver:anonymous@servidor.org.br:/var/lib/cvs
cv
```

ou

```
cv
```

Então será solicitada a senha para ter acesso ao sistema. Note que toda a seção de  `cvs`  ocorre por comandos interativos que logo após concluídos retornam para o interpretador de comandos. O restante desta seção descreverá estes comandos e como utiliza-los de maneira eficiente.

**OBS:** O uso da variável `CVSROOT` torna a utilização bastante prática, assim não precisamos especificar o repositório, método de acesso, etc. toda vez que usar um comando do  `cvs` .

## Encerrando uma seção de CVS

Embora que não seja necessário, após o uso do  `cvs`  é recomendável executar o logout do servidor para encerrar sua conexão com a máquina remota.

```
(assumindo que a variável CVSROOT está definida)
cvs logout
```

ou

```
cvs -d :pserver:anonymous@servidor.org.br:/var/lib/cvs logout
```

**OBS:** Para os paranóicos é importante encerrar uma seção de CVS, pois ele possui alguns bugs e um spoofing pode tornar possível o uso de uma seção deixada aberta.

## Baixando arquivos

O comando `checkout` (ou `co`) é usado para fazer isto. Para utilizá-lo seguindo os exemplos anteriores:

```
mkdir /tmp/cvs
cd /tmp/cvs
cvs checkout modulo
cvs -d :pserver:anonymous@servidor.org.br:/var/lib/cvs
```

Será criado um subdiretório chamado `modulo` que contém todos os arquivos do servidor de CVS remoto. É necessário apenas que tenha acesso de leitura ao servidor de CVS para executar este comando. Você pode usar a opção `-z [num]` para ativar a compactação na transferência dos arquivos, isso acelera bastante a transferência em conexões lentas: `cvs -z 3 checkout modulo`.

Também é possível especificar apenas subdiretórios de um módulo para baixa-lo via CVS e a estrutura de diretórios criada localmente será idêntica ao do servidor remoto.

## Adicionando um novo projeto

Use o comando `cvs import` para adicionar um novo projeto ao CVS. As entradas nos arquivos administrativos serão criadas e o projeto estará disponível para utilização dos usuários. A sintaxe básica do comando `import` é a seguinte:

```
cvs import [opções] [dir_modulo] [tag] start
```

Para adicionar o projeto `focalinux` que reside em `/usr/src/focalinux` ao cvs:

```
Primeiro exportamos o CVSROOT para dizer onde e qual repositório acessar
export CVSROOT=:ext:usuario@servidor.com.br:2401/var/lib/cvs
```

```
cd /usr/src/focalinux
cvs import documentos/focalinux tag_modulo start
```

Por padrão o **import** sempre utiliza a máscara `*` para fazer a importação dos arquivos do diretório atual. O projeto `focalinux` será acessado através de `$CVSROOT/documentos/focalinux` (`cvs checkout documentos/focalinux`), ou seja, `/var/lib/cvs/documentos/focalinux` no servidor CVS terá a cópia do `focalinux.tag_modulo` define o nome que será usado como identificador nas operações com os arquivos do CVS (pode ser usado "focalinux" em nosso exemplo). O parâmetro "start" diz para criar o módulo.

**OBS:** Por segurança, o diretório que contém os arquivos deverá ser sempre um caminho relativo na estrutura de diretórios, ou seja, você precisará entrar no diretório pai (como `/usr/src/projeto`) para executar o `cvs import`. Não é permitido usar `/` ou `..`, isto proíbe a descida em diretórios de nível mais altos e sérios incidentes de segurança em servidores CVS mal configurados pelo Administrador.

## Sincronizando a cópia remota com a cópia local

Este comando sincroniza a cópia remota do CVS (ou arquivo) com a cópia local que está trabalhando em sua máquina. Quando se trabalha nativamente no CVS em equipe é recomendado a utilização deste comando pois alguém pode ter modificado o arquivo antes de você, então uma incompatibilidade entre sua versão e a nova poderia causar problemas.

Supondo que tenha acabado de modificar o arquivo `main.c` do módulo `cvsproj`, então antes de fazer o commit ("Enviando as mudanças para o servidor remoto") use o `update`:

```
cvs update main.c
```

ou

```
cvs -d :ext:usuario@servidor.com.br:2401/var/lib/cvs update main.c
```

Após alguns segundos, sua cópia local ficará sincronizada com a cópia remota. Caso ele mostre alguma mensagem de saída, verifique o arquivo para solucionar qualquer conflito e então envie o arquivo para o servidor remoto ("Enviando as mudanças para o servidor remoto").

Você pode fazer o `update` de mais arquivos usando referências globais (`*`, `?` ou `[ ]`).

## Enviando as mudanças para o servidor remoto

O comando "commit" (ou "ci"), envia as mudanças feitas nos arquivos locais para o servidor remoto. Um exemplo de commit no arquivo `main.c`:

```
cvs commit main.c
```

```
cvs commit main.?
```

```
cvs commit *
```

O editor padrão do sistema será aberto e pedirá uma descrição das modificações para o commit. Esta descrição será usada como referência sobre as atualizações feitas em cada etapa do desenvolvimento. A mensagem também pode ser especificada usando a opção `"-m mensagem"`, principalmente quando o texto explicando as alterações é pequeno.

Para mudar o editor de texto padrão que será usado pelo `cv`s, altere a variável de ambiente `EDITOR` ou especifique o editor que deseja usar na linha de comando com a opção `"-e editor"`:

```
cv
```

s commit -e vi main.c

## Adicionando um arquivo ao módulo CVS do servidor

Após criar/copiar o arquivo para seu diretório de trabalho, use o comando `add` para fazer isto. O arquivo será enviado ao servidor, bastando apenas executar o `commit` para salvar o arquivo:

```
cv
```

s add main.h  
cvs commit main.h

## Adicionando um diretório ao módulo CVS do servidor

O método para adicionar um diretório com arquivos é semelhante ao de adicionar apenas arquivos ao `cv`s. O único ponto que deve ser seguido é que primeiro deve ser adicionado o diretório (com o `"cv`s add") salvar no servidor remoto (`"cv`s commit") e depois adicionar os arquivos existentes dentro dele (assim como descrito em “Adicionando um arquivo ao módulo CVS do servidor”). Para adicionar o diretório `teste` e seus arquivos no servidor `cv`s remoto:

```
cv
```

s add teste  
cvs commit -m "Adicionado" teste  
cvs add teste/\*  
cd teste  
cvs commit -m "Adicionados" .

Os dois primeiros comandos agendam o diretório `teste` e fazem o `commit` no diretório remoto. Os dois últimos, enviam os arquivos existentes dentro deste diretório para o servidor remoto.

## Removendo um arquivo do módulo CVS remoto

O comando para fazer isto é o `"remove"`. Primeiro use o `rm` para remover o arquivo/diretório de sua cópia local, depois execute o `remove` seguido de `commit` para confirmar a remoção do arquivo:

```
cv
```

s remove main.h  
cvs commit main.h

## Removendo um diretório do módulo CVS remoto

Para remover um diretório, primeiro remova todos os arquivos existentes dentro dele com o comando `rm` e salve para o servidor (seguindo os métodos descritos em “Removendo um arquivo do módulo CVS remoto”). O CVS não remove diretamente diretórios vazios, uma maneira de contornar isto é usar o `update` ou `commit` seguido da opção `-P` para ignorar diretórios vazios. Então a cópia remota do diretório será removida do servidor:

```
rm -f teste/*
cv
```

s remove teste/.  
cvs commit teste/.

```
cd ..
cvs checkout modulo
```

Depois do checkout, o subdiretório teste terá sido removido.

## Dizendo que o módulo atual não está mais em uso

O comando "release" faz esta função. Ele não é requerido, mas caso você tenha feito modificações que ainda não foram salvas no servidor de cvs (`commit`), ele alertará de arquivos modificados e perguntará se deseja continuar. Registrando também o abandono das modificações no histórico do cvs. O comando pode ser acompanhado de "-d" para remover o módulo anteriormente baixado com o "commit":

```
cvs release modulo

cvs release -d modulo
```

O `release` retorna os seguintes códigos quando verifica que as duas cópias (local e remota) não estão sincronizadas:

- |        |                                                                                                                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| U ou P | Existe uma versão nova do arquivo no repositório. Para corrigir isso, execute o comando "update".                                                                                                                       |
| A      | O arquivo não foi adicionado ainda ao repositório remoto. Se apagar o repositório local, este arquivo não será adicionado. Para corrigir isto, executa o comando "add" do cvs.                                          |
| R      | O arquivo foi removido localmente, mas não foi removido do servidor remoto. Use os procedimentos em "Removendo um arquivo do módulo CVS remoto" para corrigir a situação.                                               |
| M      | O arquivo está modificado localmente e não foi salvo ainda no servidor. Use os procedimentos em "Sincronizando a cópia remota com a cópia local" e "Enviando as mudanças para o servidor remoto" para salvar o arquivo. |
| ?      | O arquivo está em seu diretório de trabalho mas não tem referências no repositório remoto e também não está na lista de arquivos ignorados do CVS.                                                                      |

## Visualizando diferenças entre versões de um arquivo

Com o comando "diff" é possível visualizar que diferenças o arquivo que está sendo editado possui em relação ao arquivo do repositório remoto. Outra funcionalidade útil do "diff" é comparar 2 versões de arquivos do mesmo repositório CVS. Exemplos:

|                                               |                                                                                                                                  |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <code>cvs diff main.c</code>                  | Verifica as diferenças entre o arquivo <code>main.c</code> local e remoto.                                                       |
| <code>cvs diff -u -r 1.1 -r 1.2 main.c</code> | Mostra as diferenças em formato unificado para mostrar as diferenças entre as versões 1.1 e 1.2 do arquivo <code>main.c</code> . |

## Visualizando o status de versão de arquivos

O comando "status" permite verificar que versões do arquivo especificado está disponível localmente, remotamente, qual a versão inicial do arquivo no repositório, sticky tag. Exemplos:

|                                   |                                                                                                                                                      |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cvs status main.c</code>    | Verifica o status do arquivo <code>main.c</code> .                                                                                                   |
| <code>cvs status -v main.c</code> | Mostra o status do arquivo <code>main.c</code> , adicionalmente mostra também as tags existentes no arquivo (versão inicial, versão do repositório). |

## Outros utilitários para trabalho no repositório

Além dos comandos do `cv`s descritos aqui, existem comandos no pacote que auxiliam desde quem está aprendendo a utilizar o CVS (com o comando `cv`sdo para simular algumas operações de adição/remoção de arquivos) até profissionais que usam o programa no dia a dia (`cv`su, `cv`sco, `cv`schroot).

## Arquivos administrativos em CVSROOT

Esta seção descreve a função de cada um dos arquivos administrativos, isto pode ser útil na configuração e personalização do CVS e de seu repositório.

Para não alongar muito o capítulo, procurei colocar uma breve descrição da função de cada um deles, o comentários e exemplos existentes nos arquivos oferecem uma boa compreensão do seu conteúdo.

### config

Este arquivo segue os padrões dos arquivos de configuração e possui alguns parâmetros que controlam o comportamento do CVS. Segue uma lista deles:

*SystemAuth* Define se será utilizado a autenticação via `/etc/passwd` quando o método `pserver` for utilizado. Note que se o arquivo `passwd` for criado no CVSROOT, o *SystemAuth* será definido automaticamente para `no`.

Exemplo: `SystemAuth=yes`.

*LockDir* Especifica o diretório onde serão gravados os arquivos de lock. Caso não seja especificado, será usado o diretório do CVS.

Exemplo: `LockDir=/var/lock/cvs`

*TopLevelAdmin* Permite criar ou não um diretório chamado CVS no root do diretório de trabalho durante o `cv`s checkout.

*LogHistory* Utiliza opções para especificar o que será registrado nos arquivos de log do CVS.

- TOFEWGCMAR ou `all`

Registra todas as operações nos logs do `cv`s.

- TMAR

Registra todas as operações que modificam os arquivos " , v "

### modules

Especifica opções e programas externos que serão usados durante a execução de comandos no repositório CVS.

### cvswrappers

Este arquivo define ações de controle de características de arquivos, de acordo com seu nome.

Pode ser também definidas ações através de arquivos `.cvswrappers`.

## commitinfo

Define programas para fazer uma checagem baseada no diretório e dizer se o commit é permitido.

## verifysmsg

Especifica o programa usado para verificar as mensagens de log.

## loginfo

Programa que é executado após o commit. Ele pode ser usado para tratar a mensagem de log e definir onde ela será gravada/enviada, etc.

## cvsignore

Tudo que constar neste arquivo não será gravado (commit) no cvs. Referências globais podem ser usadas para especificar estes arquivos. Veja a info page do cvs para detalhes sobre seu formato.

Pode também ser especificado através de arquivos `.cvsignore`.

## checkoutlist

Especifica os arquivos que deseja manter sobre o controle do CVS que se encontram em `CVSROOT`. Se adicionar um script adicional, ou qualquer outro arquivo no diretório `CVSROOT` ele deverá constar neste arquivo.

## history

É usado para registrar detalhes do comando *history* do CVS.

# Clientes de CVS

Esta seção traz alguns programas cliente em modo texto/gráfico e visualizadores de repositórios via web. Eles facilitam o trabalho de controle de revisão por parte de iniciantes e flexibilidade para pessoas mais experientes, além de ter uma interface de navegação disponível para todos os interessados em fazer pesquisas no repositório.

## CVS

Este é o cliente **Unix** padrão, bastante poderoso e que opera em modo texto. As explicações neste capítulo do guia assumem este cliente de **cvs**, então as explicações sobre sua utilização se encontra em “Criando projetos para serem usados no CVS” e os parâmetros de linha de comando em “Opções de linha de comando”

É **altamente** recomendável a leitura caso deseje utilizar um cliente de **cvs** gráfico, pois os conceitos são os mesmos.

## gcvS - Linux

Este é um cliente CVS em GTK+Python para **Linux** que interage externamente com o cliente **cvs** externo, todas as opções do cvs estão disponíveis através de checkboxes nas telas de comando, incluindo suporte a compactação, visualizador gráfico da árvore de releases, histórico, diffs, etc.

---

Sua instalação é bastante fácil, instale o programa com `apt-get install gcvs` e execute o programa através do menu do sistema ou do terminal. Utilize os seguintes procedimentos para configurar e utilizar o programa:

1. Defina o repositório `CVSROOT` através do menu *Admin/Preferences*. Selecione o método de acesso, entre com o login, servidor e repositório.

Exemplo: `:pserver:anonymous@servidor:/var/lib/cvs`

O formato deve ser *EXATAMENTE* como o usado na variável `CVSROOT` no shell, incluindo os ":". Caso tenha erros de login, verifique o valor de `CVSROOT` cuidadosamente antes de contactar o administrador de sistemas!

2. Agora faça o login no sistema em: *Admin, Login*. Note que o status de todas as operações do **cv**s são mostradas na janela de status que fica na parte inferior da tela.
3. Crie um diretório que será usado para armazenar os fontes baixados do CVS, por exemplo: `mkdir ~/projetos`
4. Acesse o menu *Create, Checkout Module* para baixar o projeto do CVS para sua máquina local. Ele irá te pedir o nome de diretório para onde o código fonte do servidor CVS será baixado. Digite `~/projetos` ou outro diretório que foi criado no passo anterior.

**OBS:** Não utilize o nome "cvs" para o diretório local, pois o **gcvs** oculta automaticamente pois os arquivos administrativos de controle ficam neste local.

5. Altere o diretório padrão do **gcvs** para o diretório onde baixou o projeto (`~/projetos`) clicando no botão "set" no topo da coluna esquerda do **gcvs**.
6. Para enviar um arquivo modificado de volta ao servidor, selecione os arquivos, clique em *Modify, Commit Selection*, entre com a mensagem descrevendo as alterações e clique em "OK".

Note que os arquivos modificados serão identificados por um ícone vermelho e seu status será "Mod. File" (arquivo modificado).

7. Se desejar adicionar um novo projeto no CVS, entre em *Create, Import Module*, entre no diretório que deseja adicionar como um projeto no servidor de CVS. Após isto será feita uma checagem e mostrada uma tela de possíveis problemas que podem ocorrer durante a importação do novo projeto.

Na próxima tela, digite o nome do módulo e caminho no servidor remoto no primeiro campo, no segundo campo a mensagem de log para adicionar o projeto ao servidor. Em "Vendor tag" especifique o nome do projeto e sua versão logo abaixo. Clique em "OK" e aguarde a transferência dos arquivos para o servidor.

Para maiores detalhes sobre a criação de novos projetos no servidor de CVS, veja "Adicionando um novo projeto".

**OBS:** Você deverá ter permissão de gravação para criar um novo projeto no servidor CVS.

8. A partir de agora você poderá explorar as funções do programa e fazer uso das funções habituais do CVS. Todas as funções de operação e opções extras do CVS estão disponíveis na interface gráfica, basta se acostumar com sua utilização.

Após isto, explore bastante as opções do programa. Todas as funcionalidades do CVS estão organizadas entre os menus do programa. Caso não entenda bem as funções do programa, leia atentamente "Criando projetos para serem usados no CVS" e também não deixe de consultar detalhes na info page do cvs.

## WinCVS - Windows

Este é um cliente CVS em Python para **Windows** equivalente ao **gcvs** para **Linux**. Suas funcionalidades e recomendações são idênticas aos do **gcvs**. Este cliente pode ser baixado de: <http://telia.dl.sourceforge.net/sourceforge/cvsgui/WinCvs13b13.zip> e o Python para Windows de <http://starship.python.net/crew/mhammond/downloads/win32all-153.exe>.

Para sua utilização, as explicações em “gcvs - Linux” são totalmente válidas.

## MacCVS - Macintosh (PPC)

Idêntico ao **gcvs**, pode ser baixado de <http://telia.dl.sourceforge.net/sourceforge/cvsgui/MacCvsX-3.3a1-1.dmg>.

## viewcvs

Este é um visualizador de repositórios CVS via web, ele precisa apenas de um servidor web instalado com suporte a CGI. Para instalar, execute o comando `apt-get install` e siga os passos para configurar programa. Para adequar melhor o **viewcvs** ao seu sistema, edite o arquivo `/etc/viewcvs/viewcvs.conf`.

O **viewcvs** possui uma interface que se parece com a navegação de um diretório de ftp, recursos como a extração de diffs coloridos entre versões de um arquivo selecionado, visualização de commits (com data, log do commit, usuário, etc.), classificação da listagem exibida.

**OBS:**Leve em consideração as implicações de segurança impostas por aplicativos cgi sendo executados em seu sistema. Veja ??? para entender melhor o assunto.

## Exemplo de uma seção CVS

Nota: este exemplo é apenas didático, não foi feita nenhuma modificação real no conteúdo do repositório do **dillo** :-)

```
Definir o CVSROOT
export CVSROOT=:pserver:gleydson@ima.cipsga.org.br:/var/lib/cvs

entrar no servidor
gleydson@host:/tmp/teste$ cvs login
Logging in to :pserver:gleydson@ima.cipsga.org.br:2401/var/lib/cvs
CVS password: <password>

gleydson@oberon:/tmp/teste$

Pegar o módulo "dillo do cvs"
cvs -z 3 co dillo

cvs server: Updating dillo
cvs server: Updating dillo/CVSROOT
U dillo/CVSROOT/checkoutlist
U dillo/CVSROOT/commitinfo
U dillo/CVSROOT/config
U dillo/CVSROOT/cvswrappers
```

```
U dillo/CVSRROOT/editinfo
U dillo/CVSRROOT/loginfo
U dillo/CVSRROOT/modules
U dillo/CVSRROOT/notify
U dillo/CVSRROOT/rcsinfo
U dillo/CVSRROOT/taginfo
U dillo/CVSRROOT/verifymsg
cvs server: Updating dillo/CVSRROOT/Emptydir
cvs server: Updating dillo/dillo
U dillo/dillo/AUTHORS
U dillo/dillo/COPYING
U dillo/dillo/ChangeLog
U dillo/dillo/ChangeLog.old
U dillo/dillo/INSTALL
U dillo/dillo/Makefile.am
U dillo/dillo/Makefile.in
U dillo/dillo/NEWS
U dillo/dillo/README
U dillo/dillo/aclocal.m4
U dillo/dillo/config.h.in
U dillo/dillo/configure
U dillo/dillo/configure.in
U dillo/dillo/depcomp
U dillo/dillo/dillorc
U dillo/dillo/install-sh
U dillo/dillo/missing
U dillo/dillo/mkinstalldirs
U dillo/dillo/stamp-h.in
cvs server: Updating dillo/dillo/doc
U dillo/dillo/doc/Cache.txt
U dillo/dillo/doc/Cookies.txt
U dillo/dillo/doc/Dillo.txt
U dillo/dillo/doc/Dw.txt
U dillo/dillo/doc/DwImage.txt
U dillo/dillo/doc/DwPage.txt
...

Modifica o arquivo do projeto
cd /dillo/dillo/doc
vi Cache.txt

Update no arquivo para atualizar a cópia local com a remota
cvs update Cache.txt
M Cache.txt

gleydson@host:/tmp/teste

Damos o commit no arquivo
cvs commit Cache.txt

Saimos do sistema
cvs logout
```

---

# Capítulo 18. SAMBA

Este capítulo descreve a configuração, utilização, aplicação, integração de redes **Windows** e **Linux** através do **SAMBA**. Entre as explicações de cada opção, são passados detalhes importantes relacionados com seu funcionamento, performance e impactos de segurança sobre o servidor como um todo.

Uma seção foi especialmente separada para os mais paranóicos (como eu) conhecerem, combinar e aplicar as restrições de forma mais adequada a configuração da máquina.

## Introdução

O **SAMBA** é um servidor e conjunto de ferramentas que permite que máquinas **Linux** e **Windows** se comuniquem entre si, compartilhando serviços (arquivos, diretório, impressão) através do protocolo SMB (Server Message Block)/CIFS (Common Internet File System), equivalentes a implementação NetBEUI no Windows. O **SAMBA** é uma das soluções em ambiente UNIX capaz de interligar redes heterogênea.

Na lógica da rede Windows o **NetBEUI** é o protocolo e o **NETBIOS** define a forma com que os dados são transportados. Não é correto dizer que o **NetBIOS** é o protocolo, como muitos fazem.

Com o **SAMBA**, é possível construir domínios completos, fazer controle de acesso a nível de usuário, compartilhamento, montar um servidor WINS, servidor de domínio, impressão, etc. Na maioria dos casos o controle de acesso e exibição de diretórios no samba é mais minucioso e personalizável que no próprio Windows.

O guia Foca GNU/Linux documentará como instalar e ter as máquinas Windows de diferentes versões (**Win3.11**, **Win95**, **Win95OSR/2**, **Win98**, **XP**, **WinNT**, **W2K**) acessando e comunicando-se entre si em uma rede NetBEUI. Estas explicações lhe poderão ser indispensáveis para solucionar problemas, até mesmo se você for um técnico especialista em redes Windows e não tem ainda planos de implementar um servidor SAMBA em sua rede.

## Versão documentada

A versão do servidor samba documentada neste capítulo do guia é a 2.2.

## História

Andrew Tridgell - Desenvolveu o samba porque precisava montar um volume Unix em sua máquina DOS. Inicialmente ele utilizava o NFS, mas um aplicativo precisava de suporte NetBIOS. Andrew então utilizou um método muito avançado usado por administradores para detectar problemas: escreveu um sniffer de pacotes que atendesse aos requerimentos para ter uma única função: analisar e auxilia-lo a interpretar todo o tráfego NetBIOS da rede.

Ele escreveu o primeiro código que fez o servidor Unix aparecer como um servidor de arquivos Windows para sua máquina DOS que foi publicado mais ou menos em meados de 1992 quando também começou a receber patches. Satisfeito com o funcionamento de seu trabalho, deixou seu trabalho de lado por quase 2 anos. Um dia, ele resolveu testar a máquina Windows de sua esposa com sua máquina Linux, e ficou maravilhado com o funcionamento do programa que criou e veio a descobrir que o protocolo era documentado e resolveu levar este trabalho a fundo melhorando e implementando novas funções.

O SAMBA atualmente é um servidor fundamental para a migração de pequenos grupos de trabalho à grandes domínios com clientes mixtos. Nenhum servidor de rede NetBEUI conhecido proporciona tanta flexibilidade de acesso a clientes como o SAMBA para compartilhamento de arquivos/impressão em rede.

As funções de segurança que foram adicionadas ao SAMBA hoje garantem um controle mais rigoroso que a própria implementação usada no Windows NT, incluindo o serviços de diretórios, mapeamento entre IDs de usuários Windows com Linux, PDC, perfis móveis e uma coisa que inclusive apresenta problemas no Windows: compatibilidade total entre as diferentes implementações de versões do Windows.

Sua configuração pode receber ajustes finos pelo administrador nos soquetes TCP de transmissão, recepção, cache por compartilhamento, configurações físicas que afetam a performance de rede. Seu código vem sendo melhorado constantemente por hackers, obtendo excelente performance com hardwares mais obsoletos. O guia tem por objetivo abordar estes temas e permitir que você configure seu sistema com uma performance batendo a mesma alcançada em um servidor NT dedicado.

## Contribuindo

Para contribuir com o desenvolvimento do samba, veja os detalhes na página: <http://us1.samba.org/samba/devel/>

Caso encontre um bug no programa, ele poderá ser relatado se inscrevendo na lista de discussão [samba-technical-request@lists.samba.org](mailto:samba-technical-request@lists.samba.org). Após responder a mensagem de confirmação, envie um relato detalhado do problema encontrado no programa.

## Características

Segue abaixo algumas funcionalidades importantes de aplicações do samba e seu conjunto de ferramentas:

- Compartilhamento de arquivos entre máquinas **Windows** e **Linux** ou de máquinas **Linux** (sendo o servidor SAMBA) com outro SO que tenha um cliente NetBEUI (Macintosh, OS/2, LanManager, etc).
- Montar um servidor de compartilhamento de impressão no **Linux** que receberá a impressão de outras máquinas **Windows** da rede.
- Controle de acesso aos recursos compartilhados no servidor através de diversos métodos (compartilhamento, usuário, domínio, servidor).
- Controle de acesso leitura/gravação por compartilhamento.
- Controle de acesso de leitura/gravação por usuário autenticado.
- Possibilidade de definir contas de "Convidados", que podem se conectar sem fornecer senha.
- Possibilidade de uso do banco de dados de senha do sistema (`/etc/passwd`), autenticação usando o arquivo de dados criptografados do samba, LDAP, PAM, etc.
- Controle de cache e opções de tuning por compartilhamento.
- Permite ocultar o conteúdo de determinados diretórios que não quer que sejam exibidos ao usuário de forma fácil.
- Possui configuração bastante flexível com relação ao mapeamento de nomes DOS => UNIX e vice versa, página de código, acentuação, etc.
- Permite o uso de aliases na rede para identificar uma máquina com outro nome e simular uma rede NetBIOS virtual.
- O **samba** possibilita ajuste fino nas configurações de transmissão e recepção dos pacotes TCP/IP, como forma de garantir a melhor performance possível de acordo com suas instalações.

- Permite o uso do gerenciador de mensagem do Linux (**Linpopup**) para a troca de mensagens com estações **Windows** via **NetBIOS**. Com a flexibilidade do **samba** é possível até redirecionar a mensagem recebida via e-mail ou pager.
- Possui suporte completo a servidor WINS (também chamado de *NBNS - NetBios Name Service*) de rede. A configuração é bastante fácil.
- Faz auditoria tanto dos acessos a pesquisa de nomes na rede como acesso a compartilhamentos. Entre os detalhes salvos estão a data de acesso, IP de origem, etc.
- Suporte completo a controlador de domínio Windows (PDC).
- Suporte quase completo a backup do controlador de domínio (BDC). Até a versão 2.2 do **samba**, o suporte a BDC é parcial. Este código provavelmente estará estável até a versão 3.0.
- Permite montar unidades mapeadas de sistemas **Windows** ou outros servidores **Linux** como um diretório no **Linux**.
- Permite a configuração de recursos simples através de programas de configuração gráficos, tanto via sistema, como via web.
- Permite executar comandos no acesso ao compartilhamento ou quando o acesso ao compartilhamento é finalizado.
- Com um pouco de conhecimento e habilidade de administração de sistemas **Linux**, é possível criar ambientes de auditoria e monitoração até monitoração de acesso a compartilhamento em tempo real.
- Entre outras possibilidades.

## Ficha técnica

Pacote

Outros utilitários importantes para a operação do clientes **samba**.

- `smbclient` - Ferramenta para navegação e gerenciamento de arquivos, diretórios e impressoras compartilhados por servidores **Windows** ou **samba**.
- `smbfs` - Pacote que possui ferramentas para o mapeamento de arquivos e diretórios compartilhados por servidores **Windows** ou **samba** em um diretório local.
- `winbind` - Daemon que resolve nomes de usuários e grupo através de um servidor NT/SAMBA e mapeia os UIDs/GIDs deste servidor como usuários locais.

## Requerimentos de Hardware

Processador 386 ou superior, 15 MB de espaço em disco (não levando em conta os logs gerados e espaço para arquivos de usuários, aplicativos, etc.), 8 MB de memória RAM.

## Arquivos de log criados

O daemon `nmbd` salva seus logs em `/var/log/daemon.log` (dependendo da diretiva de configuração `syslog` do arquivo `smb.conf`) e em `/var/log/samba/log.nmbd`. Os detalhes de acesso a compartilhamento são gravados no arquivo `/var/log/samba/log.smbd` (que pode ser modificado de acordo com a diretiva `log file` no `smb.conf`, "Log de acessos/serviços").

## Instalação

Digite `apt-get install samba smbclient smbfs` para instalar o conjunto de aplicativos samba. O pacote é o servidor samba e os pacotes e fazem parte dos aplicativos cliente. Caso deseje apenas mapear compartilhamentos remotos na máquina Linux, instale somente os 2 últimos pacotes.

## Iniciando o servidor/reiniciando/recarregando a configuração

O servidor samba pode ser executado tanto via `inetd` como `daemon`:

`inetd` No modo `inetd`, o servidor de nomes `nmbd` será carregado assim que for feita a primeira requisição de pesquisa e ficará residente na memória. No caso de acesso a um compartilhamento, o `smbd` será carregado e lerá a configuração em `smb.conf` a cada acesso do cliente a um compartilhamento. Quando o **samba** opera via `inetd`, ele não usa o controle de acesso dos arquivos `hosts.allow` e `hosts.deny`. Veja “Restringindo o acesso por IP/rede” e “Restringindo o acesso por interface de rede” para detalhes de como fazer o controle de acesso.

Para reiniciar o samba digite `killall -HUP nmbd`. Não é necessário reiniciar o serviço `smbd`, conforme foi explicado acima.

`daemon` Quando opera no modo `daemon`, ambos os daemons **nmbd** e **smbd** são carregados. No caso de um acesso a compartilhamento, é criado um processo filho do **smbd** que é finalizado assim que o compartilhamento não for mais usado.

Para iniciar o samba em modo `daemon` digite: `/etc/init.d/samba start`, para interromper o samba: `/etc/init.d/samba stop` e para reiniciar: `/etc/init.d/samba restart`.

Se desejar mudar do modo `daemon` para `inetd` ou vice versa, edite o arquivo `/etc/default/samba` e modifique o valor da linha `RUN_MODE=` para `daemons` ou `inetd`. Uma forma de fazer isso automaticamente é executando o `dpkg-reconfigure samba`.

**OBS:** Como praticamente não existe diferença entre os modos de operação `inetd` e `daemon` para o **SAMBA**, é aconselhável que execute sempre que possível via `inetd`, pois isto garantirá uma disponibilidade maior do serviço caso algo aconteça com um dos processos.

## Opções de linha de comando

Opções de linha de comando usadas pelo **nmbd**:

- H [arquivo\_lmhosts] Quando especificado, o servidor samba fará a procura de nomes primeiro neste arquivo e depois usando a rede.
- s [arquivo\_cfg] Especifica uma nova localização para o arquivo de configuração do samba. Por padrão o `/etc/samba/smb.conf` é usado.
- d [num] Especifica o nível de depuração do `nmbd`, que podem ir de 0 a 10. O valor padrão é 0.
- l [diretório] Especifica a localização do diretório onde o `nmbd` gravará o arquivo de log `log.nmbd`. O valor padrão é `/var/log/samba`

-n [nomeNetBIOS]

Permite utilizar o nome NetBIOS especificado a invés do especificado no arquivo `smb.conf` para identificar o computador na rede.

## Conceitos gerais para a configuração do SAMBA

Este capítulo documenta como configurar o seu servidor **SAMBA** permitindo o acesso a compartilhamento de arquivos e impressão no sistema.

### Nome de máquina (nome NetBios)

Toda a máquina em uma rede NetBEUI é identificada por um nome, este nome deve ser único na rede e permite que outras máquinas acessem os recursos disponibilizados ou que sejam enviadas mensagens para a máquina. Este nome é composto de 16 caracteres, sendo 15 que identificam a máquina e o último o tipo de serviço que ela disponibiliza. O tipo de serviço é associado com o nome da máquina e registrado em servidores de nomes confirme a configuração da máquina (você verá isto mais adiante).

O nome de máquina é especificado nas diretivas *netbios name* e *netbios aliases* (veja “Nomes e grupos de trabalho”) para detalhes.

### Grupo de trabalho

O grupo de trabalho organiza a estrutura de máquinas da rede em forma de árvore, facilitando sua pesquisa e localização. Tomemos como exemplo uma empresa grande com os departamentos comunicação, redes, web, rh, as máquinas que pertencem ao grupo de redes serão agrupadas no programa de navegação:

```
redes
 gleydson
 tecnico
 marcelo
 henrique
 michelle

rh
 mrpaoduro

web
 web1
 web2
 web3

comunicacao
 comunic1
 comunic2
 comunic3
```

A segurança no acesso a arquivos e diretórios na configuração de *grupo de trabalho* é controlada pela própria máquina, normalmente usando segurança a nível de compartilhamento. Esta segurança funciona definindo um usuário/senha para acessar cada compartilhamento que a máquina possui. O **Lan Manager**, **Windows for Workgroups**, **Windows 95**, **Windows 98**, **XP Home Edition** usam este nível de acesso

por padrão. Se deseja configurar uma rede usando o nível de grupo de trabalho, veja “Configuração em Grupo de Trabalho” para detalhes passo a passo e exemplos práticos.

Os programas para navegação na rede NetBIOS são mostrados em “smbclient”, “nmblookup” e “Programas de navegação gráficos”.

## Domínio

O funcionamento é semelhante ao grupo de trabalho, com a diferença que a segurança é controlada pela máquina central (PDC) usando diretivas de acesso e grupos. O PDC (Primary Domain Controller) deverá ter todas as contas de acesso que serão utilizadas pelo usuário para acessar os recursos existentes em outras máquinas, script de logon que poderá ser executado em cada máquina para fazer ajustes, sincronismo, manutenção ou qualquer outra tarefa programada pelo administrador do sistema.

Estas características para configuração de máquinas em domínio são documentadas passo a passo em “Uma breve introdução a um Domínio de rede”.

## Compartilhamento

Um compartilhamento é um recurso da máquina local que é disponibilizado para acesso via rede, que poderá ser *mapeada* (veja “Mapeamento”) por outra máquina. O compartilhamento pode ser um diretório, arquivo, impressora. Além de permitir o acesso do usuário, o compartilhamento pode ser protegido por senha, e ter controle de acesso de leitura/gravação, monitoração de acessos, diretórios ocultos, autenticação via PDC (domínio) e outras formas para restringir e garantir segurança na disponibilização dos dados (veja “Controle de acesso ao servidor SAMBA” para aprender os métodos de como fazer isto).

Um compartilhamento no SAMBA pode ser acessível publicamente (sem senha) ou estar rigidamente protegido tendo que passar por diversas barreiras para chegar ao seu conteúdo, como senhas, endereço de origem, interfaces, usuário autorizados, permissões de visualização, etc.

O guia *Foca Linux* abordará estes assuntos com detalhes e explicará didaticamente como tornar seguro seu servidor samba e garantir um minucioso controle de acesso a seus compartilhamentos.

## Mapeamento

Mapear significa pegar um diretório/arquivo/impressora compartilhado por alguma máquina da rede para ser acessada pela máquina local. Para mapear algum recurso de rede, é necessário que ele seja compartilhado na outra máquina (veja “Compartilhamento”). Por exemplo, o diretório `/usr` compartilhado com o nome `usr`, pode ser mapeado por uma máquina **Windows** como a unidade `F:`, ou mapeado por uma máquina **Linux** no diretório `/mnt/samba`.

O programa responsável por mapear unidades compartilhadas no **Linux** é o `smbmount` e `smbclient` (veja “Linux”).

## Navegação na Rede e controle de domínio

Esta função é controlada pelo `nmbd` que fica ativo o tempo todo disponibilizando os recursos da máquina na rede, participando de eleições NetBIOS (“Níveis de sistema para eleição de rede”), fazer logon de máquinas no domínio (“Uma breve introdução a um Domínio de rede”), etc.

A função de navegação na rede é feita utilizando o compartilhamento `IPC$`. Este compartilhamento possui acesso público somente leitura e utiliza o usuário “guest” para disponibilização de seus. Como deve ter percebido, é necessário especificar esta ID de usuário através do parâmetro `guest account`

(“Descrição de parâmetros usados em compartilhamento”), ou a navegação de recursos no sistema (ou na rede, dependendo da configuração do SAMBA) não funcionará.

**OBS:** A função de navegação (browsing) poderá não funcionar corretamente caso a máquina não consiga resolver nomes NetBIOS para endereços IP.

## Arquivo de configuração do samba

Toda a configuração relacionada com nomes, grupo de trabalho, tipo de servidor, log, compartilhamento de arquivos e impressão do **samba** é colocada no arquivo de configuração `/etc/samba/smb.conf`. Este arquivo é dividido em *seções* e *parâmetros*.

Uma seção no arquivo de configuração do **samba** (`smb.conf`) é definido por um nome entre “[ ]”. As seções tem o objetivo de organizar os parâmetros pra que tenham efeito somente em algumas configurações de compartilhamento do servidor (exceto os da seção `[global]` que não especificam compartilhamentos, mas suas diretivas podem ser válidas para todas os compartilhamentos do arquivo de configuração). Alguns nomes de seções foram reservados para configurações específicas do **samba**, eles são os seguintes:

- `[global]` Define configurações que afetam o servidor samba como um todo, fazendo efeito em todos os compartilhamentos existentes na máquina. Por exemplo, o grupo de trabalho, nome do servidor, página de código, restrições de acesso por nome, etc. Veja “Seção `[global]`”.
- `[homes]` Especifica opções de acesso a diretórios homes de usuários. O diretório home é disponibilizado somente para seu dono, após se autenticar no sistema. Veja “Seção `[homes]`”.
- `[printers]` Define opções gerais para controle das impressoras do sistema. Este compartilhamento mapeia os nomes de todas as impressoras encontradas no `/etc/printcap`. Configurações especiais podem ser feitas separadamente. Veja “Seção `[printers]`”.
- `[profile]` Define um perfil quando o servidor **samba** é usado como PDC de domínio. Veja “Configurando perfis de usuários”.

Qualquer outro nome de `[seção]` no arquivo `smb.conf` que não sejam as acima, são tratadas como um compartilhamento ou impressora.

Um *parâmetro* é definido no formato *nome = valor*. Para um exemplo prático, veja um exemplo de arquivo `smb.conf` em “Exemplos de configuração do servidor SAMBA”. Na configuração de booleanos, a seguinte sintaxe pode ser usada:

- 0 ou 1
- yes ou no
- true ou false

Assim, as seguintes configurações são equivalentes

```
master browse = 0
master browse = no
master browse = false
```

Todos significam "NÃO ser o navegador principal de domínio". A escolha fica a gosto do administrador. Durante a configuração, você notará o poder da flexibilidade oferecida pelo samba na configuração de um servidor SMB :-)

Linhas iniciadas por # ou ; são tratadas como comentário. Quebras de linha pode ser especificadas com uma \ no final da linha.

## Seção [global]

Os parâmetros especificados nesta seção tem efeito em todo o servidor **samba** incluindo os compartilhamentos. Caso o parâmetro seja novamente especificado para o compartilhamento, o valor que valerá é o do compartilhamento.

Por exemplo, se `guest user = nobody` for usado na seção [global] e o `guest user = foca` for usado no compartilhamento [focalinux], o usuário que fará acesso público a todos os compartilhamentos do servidor será o `nobody`, exceto para o compartilhamento [focalinux], que será feito pelo usuário `foca`. Veja “Compartilhamento de arquivos e diretórios” para obter uma lista e descrição dos principais parâmetros de compartilhamentos existentes. Uma lista completa pode ser obtida na página de manual do `smb.conf`.

Irei descrever alguns parâmetros utilizados nesta seção, organizado de forma didática e simplificada.

## Nomes e grupos de trabalho

`netbios name = [nome do servidor]` Especifica o nome NetBIOS primário do servidor samba. Caso não seja ajustado, ele usará o `hostname` de sua máquina como valor padrão.

Ex.: `netbios name = focasamba`.

`workgroup = [grupo de trabalho/ domínio]` Diz qual o nome do grupo de trabalho/domínio que a máquina samba pertencerá.

Ex.: `workgroup = focalinux`.

`netbios aliases = [nomes alternativos ao sistema]` Permite o uso de nomes alternativos ao servidor, separados por espaços.

Ex.: `testel teste2`.

`server string = [identificação]` Identificação enviada do servidor samba para o ambiente de rede. A string padrão é `Samba %v` (%v é substituída pela versão do samba, para maiores detalhes, veja “Variáveis de substituição”).

Ex: `server string = Servidor Samba versão %v`.

`name resolve order = [ordem]` Define a ordem de pesquisa para a resolução de nomes no samba. A ordem padrão é: `lmhosts host wins bcast`, que é a melhor para resolução rápida e que tente gerar menos tráfego broadcast possível. Veja “Resolução de nomes de máquinas no samba” para uma explicação mais detalhada.

## Caracteres e página de código

Uma das partes essenciais após colocar o **SAMBA** em funcionamento, é configurar a página de código para que os caracteres sejam gravados e exibidos corretamente no cliente. A primeira coisa que precisa verificar é se seu kernel possui o suporte a página de código local. Caso não tenha, baixe o fonte do kernel e siga os seguintes passos na configuração:

- Dentro da opção "File Systems", "Network File Systems" defina como "Default Remote NLS Option" a iso8859-1. Esta opção permite ao **smbmount** montar os volumes locais usando os caracteres corretos.
- Entre na opção "File Systems", "Native Language Support". Na opção "Default NLS Option" coloque "iso8859-1". Ative também o suporte as páginas de código 437, 850 e 860 e também ao conjunto de caracteres iso8859-1 e UTF8.

Note que esta ordem pode variar dependendo da versão do seu kernel, basta que as entenda para fazer as modificações apropriadas. Em caso de dúvidas sobre a compilação do kernel, veja ???.

|                                          |                                                                                                                                                                                                                                                                                     |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| character set =<br>[conjunto_caracteres] | Seleciona o conjunto de caracteres dos arquivos exibidos pelo servidor samba. Para os idiomas de língua latina, sempre utilize iso8859-1.<br><br>Ex.: <code>character set = iso8859-1.</code>                                                                                       |
| client code page =<br>[pagina_de_codigo] | Seleciona a página de código do servidor samba para tratar os caracteres. Para os idiomas de língua latina, sempre utilize 850.<br><br>Ex.: <code>client code page = 850.</code>                                                                                                    |
| preserve case =                          | Seleciona se arquivos com nomes extensos criados serão criados com os caracteres em maiúsculas/minúsculas definidos pelo cliente (no) ou se será usado o valor de <i>default case</i> (caso seja especificado yes).                                                                 |
| short preserve case =                    | Seleciona se os arquivos com nomes curtos (formato 8.3) serão criados com os caracteres mixtos enviados pelo cliente (no) ou se será usando o valor de <i>default case</i> (caso seja especificado yes).                                                                            |
| default case = [lower/upper]             | Define se os arquivos criados terão seus nomes todos em minúsculas (lower) ou maiúsculas (upper).                                                                                                                                                                                   |
| valid chars = [caracteres]               | Define caracteres válidos nos nomes de arquivos: <code>valid chars =ã:Ã é:É í:Í ó:Ó ú:Ú â:Â ê:Ê ô:Ô ã:Ã õ:Õ à:À ò:Ò</code> . Este parâmetro <b>DEVERÁ</b> ser sempre especificado depois do <code>client code page</code> , pois caso contrário, eles serão substituídos por estes. |

## Restrições de acesso/mapeamento de usuários

|                                      |                                                                                                                                                                                                                                                                    |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>guest account = [conta]</code> | Define a conta local de usuário que será mapeada quando um usuário se conectar sem senha (usuário guest). Veja mais detalhes em "Descrição de parâmetros usados em compartilhamento".                                                                              |
| <code>invalid users</code>           | Define uma lista de usuários que não terão acesso aos recursos do servidor ou compartilhamento. É seguro restringir o acesso samba a usuários com grande poder no sistema (como o <code>root</code> ). Veja mais detalhes em "Restringindo o acesso por usuários". |
| <code>valid users</code>             | Semelhante a opção <code>invalid users</code> mas permite que somente os usuários especificados tenham acesso ao sistema. Veja mais detalhes em "Restringindo o acesso por usuários".                                                                              |
| <code>default service = nome</code>  | Caso o serviço que o usuário deseja se conectar não for encontrado no servidor, o SAMBA mapeará o serviço especificado nesta                                                                                                                                       |

diretiva como alternativa. A variável "%S" e o caracter "\_" podem ser interessantes em algumas alternativas de configuração. A opção `default` é um sinônimo para esta opção. Caso utilize esta opção, crie o compartilhamento em modo somente leitura e com acesso público, caso contrário (dependendo do planejamento de partições e segurança do sistema de arquivos) a máquina poderá ser derrubada sem dificuldades.

`username map = [arquivo]` Especifica um arquivo que faz o mapeamento entre nomes fornecidos por clientes e nomes de contas Unix locais. Veja "Mapeamento de nomes de usuários" para mais detalhes de como configurar este recurso.

`obey pam restrictions = yes` Indica se as restrições do usuário nos módulos PAM terão efeito também no SAMBA.

## Níveis de autenticação

(esta seção contém algumas explicações que dependem do resto do conteúdo do guia, caso não entenda de imediato a fundo as explicações, recomendo que a leia novamente mais tarde).

Define o nível de segurança do servidor. Os seguintes valores são válidos:

- `share` - Usada principalmente quando apenas a senha é enviada por compartilhamento acessado para o servidor, caso muito típico em sistemas **Lan Manager** e **Windows for Workgroups**.

Mesmo assim o samba tenta mapear para um UID de usuário local do sistema usando os seguintes métodos (retirado da página de manual do samba):

1. Se o parâmetro `guest only` é usado no compartilhamento junto com o `guest ok`, o acesso é imediatamente permitido, sem verificar inclusive a senha.
2. Caso um nome de usuário seja enviado junto com a senha, ele é utilizado para mapear o UID e aplicar as permissões deste usuário (como acontece no nível de segurança `user`).
3. Se ele usou um nome para fazer o logon no **Windows** este nome será usado como usuário local do SAMBA. Caso ele seja diferente, você deverá usar o mapeamento de nomes para associar o nome remoto do nome local (veja "Mapeamento de nomes de usuários")
4. O nome do serviço é tentado como nome de usuário.
5. O nome da máquina NetBios é tentada como nome de usuário
6. Os usuários especificados na opção `user` do compartilhamentos são utilizados (veja "Descrição de parâmetros usados em compartilhamento").
7. Caso nenhum destes métodos acima for satisfeito, o acesso é NEGADO.

Hoje em dia, o uso do nível de acesso `share` é raramente usado, porque todos os sistemas a partir do **Windows 95** e acima enviam o nome de usuário ao acessar um compartilhamento (caindo na segunda checagem do nível `share`), sendo equivalente a usar o nível `user`. Entretanto, o nível de segurança `share` é recomendado para servidores onde TODO o conteúdo deve ter acesso público (seja leitura ou gravação) e o parâmetro `guest shares` também funciona nativamente.

As senhas criptografadas (`encrypt passwords = 1`) **NÃO** funcionarão no nível `share`, lembre-se deste detalhe.

- `user` - Este é o padrão. O usuário precisa ter uma conta de usuário no **Linux** para acessar seus compartilhamentos. A mesma conta de usuário/senha deverá ser usada no **Windows** para acessar seus recursos ou realizado um mapeamento de nomes de usuários (veja “Mapeamento de nomes de usuários”). Este é o padrão do SAMBA. No nível de acesso `user` o usuário precisa ser autenticado de qualquer forma, inclusive se for usado o parâmetro `guest only` ou `user`. Os seguintes passos são usados para autorizar uma conexão usando o nível `user` (retirado da documentação do SAMBA):
  - É tentada a validação usando o nome/senha passados pelo cliente. Se tudo estiver OK, a conexão é permitida.
  - Caso já tenha se autenticado anteriormente para acessar o recurso e forneceu a senha correta, o acesso é permitido.
  - O nome NetBIOS da máquina do cliente e qualquer nome de usuário que foi usado é novamente tentado junto com a senha para tentar permitir o acesso ao recurso compartilhado.
  - Caso o cliente tenha validado o nome/senha com o servidor e o cliente enviou novamente o token de validação, este nome de usuário é usado.
  - É tentada a checagem com o parâmetro `user` no compartilhamento (veja “Descrição de parâmetros usados em compartilhamento”).
  - É verificado se o serviço é público, então a conexão é feita usando o usuário `guest account` e ignorando a senha (veja “Criando um compartilhamento para acesso sem senha”).
- `domain` - Neste nível, o acesso só será permitido quando a máquina for adicionada ao domínio com o `smbpasswd` (“Linux”). Neste nível de acesso, a conta de usuário será validada em um servidor PDC (controlador de domínio) e o acesso aos recursos das máquinas que fazem parte do domínio será feito a partir do PDC. Veja “Linux” para detalhes.
- `server` - A máquina samba tentara autenticar o usuário em outro servidor NT (ou samba). No caso da autenticação falhar, será usado o nível de acesso `user` na base de usuários local (será necessário o arquivo de senhas criptografado do samba para que a autenticação local funcione, veja “Ativando o suporte a senhas criptografadas”). Este nível é bastante usado quando configuramos um servidor de perfis de usuários ou logon separado do PDC.

## Log de acessos/serviços

`log file=` [arquivo]

Define a localização e nome do arquivo de log gerado pelo samba. As variáveis de expansão podem ser usadas caso o administrador queira ter um melhor controle dos logs gerados (veja “Variáveis de substituição”).

Ex.: `/var/log/samba/samba-log-%m`.

**OBS:** Se possível coloque uma extensão no arquivo de log gerado pelo **SAMBA** (como `.log`). O motivo disto é porque se estes logs forem rotacionados pelo **logrotate** você terá problemas de recompactação múltiplas caso utilize um coringa `samba-log-*`, gerando arquivos como `.gz.gz.gz...`, lotando a tabela de arquivos do diretório e deixando sua máquina em um loop de compactação.

`max log size =` [tamanho]

Especifica o tamanho máximo em Kb do arquivo de log gerado pelo samba. O valor padrão é 5000Kb (5MB).

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| debug pid = [valor]          | Este processo adiciona a pid aos logs gerados pelo processo <b>smbd</b> . Isto é útil para depuração caso existam múltiplos processos rodando. O valor padrão é <i>no</i> e a opção <i>debug timestamp</i> deve ser <i>yes</i> para esta opção ter efeito.                                                                                                                                                                                                                                                                                                                    |
| debug timestamp = [valor]    | Ativa ou desativa a gravação de data/hora nos arquivos de log gerados pelo samba. O valor padrão é <i>yes</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| debug level = [valor]        | Aumenta o nível de depuração dos daemons do SAMBA de 0 a 9. Um nível de depuração interessante e que produz uma quantidade razoável de dados para configuração de um <b>logrotate</b> só para o SAMBA é o 2, produzindo a lista de todos os compartilhamentos acessados, quem acessou, data/hora (dependendo das outras opções de depuração). Isto permite ao administrador saber exatamente o que está sendo acessado e por quem, quais as tentativas de acesso. Assim terá certeza que o conteúdo não está sendo acessado indevidamente. O nível de depuração 0 é o padrão. |
| debug uid = [valor]          | Este parâmetro inclui o euid, egid, uid, gid nos arquivos de log. O valor padrão é <i>no</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| lock directory = [diretório] | Define onde serão gravados os arquivos de lock gerados pelo samba.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## Navegação no servidor/tipo de servidor

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| os level=[num]          | Especifica o nível do sistema operacional. Este número é usado para as eleições netbios para definir o navegador de grupo local e controlador de domínio (veja “Níveis de sistema para eleição de rede” para detalhes). O valor pode ser de 0 a 255, o padrão é 32.                                                                                                                                                                                                                                                                                         |
| announce as = [sistema] | Selecione o nome que o <b>samba (nmbd)</b> se anunciará na lista de pesquisa de rede. Os seguintes nomes podem ser usados: <ul style="list-style-type: none"> <li>• <i>NT Server</i> (ou <i>NT</i>) - Anuncia como <i>Windows NT Server</i>. Este é o padrão.</li> <li>• <i>NT Workstation</i> - Anuncia-se como um <i>NT Workstation</i>.</li> <li>• <i>Win95</i> ou <i>wfW</i> - Anuncia-se na rede como uma estação <i>Windows 9x</i>, <i>Windows for Workgroups</i>, <i>Windows NT Server</i> e <i>Windows NT Workstation</i> de uma só vez.</li> </ul> |
| domain master = [valor] | Diz se o servidor tentará se tornar o navegador principal de domínio. Os valores que podem ser especificados são: <i>yes</i> , <i>no</i> e <i>auto</i> . O valor padrão é <i>auto</i> . Veja “Domain Master Browser”.                                                                                                                                                                                                                                                                                                                                       |
| local master = [valor]  | Diz se o servidor participará ou não das eleições para navegador local do grupo de trabalho (workgroup). Os valores que podem ser especificados são: <i>yes</i> , <i>no</i> . O valor padrão é <i>yes</i> . Para vencer a eleição, o samba precisa ter o valor de <i>os level</i> maior que os demais.                                                                                                                                                                                                                                                      |

Note também que o Windows NT não aceita perder as eleições e convoca uma nova eleição caso ele perca. Como esta eleição é feita via broadcasting, isso gera um tráfego grande na rede. Desta forma,

se tiver um computador NT na rede configure este valor para "no". Veja "Local Master Browser".

preferred master = [valor]

Diz se o servidor samba terá ou não vantagens de ganhar uma eleição local. Se estiver configurado para "yes", o servidor samba pedirá uma eleição e terá vantagens para ganha-la. O servidor poderá se tornar garantidamente o *navegador principal do domínio* se esta opção for usada em conjunto com *domain master = 1*. Os valores especificados podem ser *yes*, *no* e *auto*, o padrão é *auto*.

Antes de ajustar este valor para *yes*, verifique se existem outros servidores NetBIOS em sua rede que tem preferência para se tornar o master principal, pois poderá ocorrer um tráfego alto de broadcasting causado pelas eleições solicitadas pelas outras máquinas.

## Outros parâmetros de configuração

`include` Inclui um outro arquivo de configuração na porção atual do arquivo de configuração. Você pode utilizar variáveis de substituição, exceto %u, %P e %S (veja "Variáveis de substituição").

## Seção [homes]

Esta seção tem a função especial de disponibilizar o diretório home do usuário. Quando o usuário envia seu nome de login como compartilhamento é feita uma busca no arquivo `smb.conf` procurando por um nome de compartilhamento que confira. Caso nenhum seja encontrado, é feita uma busca por um nome de usuário correspondente no arquivo `/etc/passwd`, se um nome conferir e a senha enviada também, o diretório de usuário é disponibilizado como um compartilhamento com o mesmo nome do usuário local. O diretório home do usuário poderá ser modificado com o uso de mapeamento de nomes, veja "Mapeamento de nomes de usuários". Quando o caminho do compartilhamento não for especificado, o SAMBA utilizará o diretório home do usuário (no `/etc/passwd`).

Para maior segurança da instalação, principalmente porque o diretório home do usuário não é um requerimento para a autenticação de usuário, recomendo usar a variável de substituição %S apontando para um diretório com as permissões apropriadas configuradas em seu sistema, por exemplo:

```
[homes]
comment = Diretórios de Usuários
path=/pub/usuarios/%S
```

Você apenas terá o trabalho extra de criar os diretórios de usuários que farão acesso ao sistema. Isto não será nenhum problema após você programar um shell script simples que verifique os nomes de contas em `/etc/passwd` e crie os diretórios com as permissões/grupos adequados (isso não será abordado por este capítulo do guia, embora não seja complicado). Se deseja, existem exemplos em "Exemplos de configuração do servidor SAMBA" sobre a seção `[homes]` no arquivo de configuração.

Os parâmetros aceitos em `[homes]` aqui são os mesmos usados para compartilhamentos normais (veja "Descrição de parâmetros usados em compartilhamento"). Abaixo segue mais um exemplo de seção `[homes]`:

```
[homes]
comment = Diretório home de usuários
```

```
writable = yes
public = no
invalid users = root nobody @adm
follow symlinks = no
create mode = 0640
directory mode = 0750
```

A explicação de cada um dos parâmetros podem ser encontradas em “Descrição de parâmetros usados em compartilhamento”. O guia está com os parâmetros bem organizados em seções específicas, apenas de uma olhada para entender com o capítulo do SAMBA foi organizado e não terá dificuldades de se localizar.

**OBS1:**Caso nenhum caminho de compartilhamento seja utilizado, o diretório home do usuário será compartilhado.

**OBS2:**Não utilize o parâmetro *public yes* na seção *guest*, caso contrário todos os diretórios de usuários serão lidos por todos. Veja “Considerações de segurança com o uso do parâmetro "public = yes"” para maiores detalhes.

## Seção [printers]

Esta seção tem a função de disponibilizar as impressoras existentes no sistema (*lp*, *lp1*, *lp2*, etc) existentes no `/etc/printcap` como compartilhamento de sistemas Windows. O método que os nomes de impressoras são pesquisados é idêntico a forma feita para a seção *[homes]*: Primeiro o nome do compartilhamento é pesquisado como um nome de serviço, depois se ele é um nome de usuário (tentando mapear o serviço disponibilizado em *[homes]*), depois será verificado a seção *[printers]*.

Ao invés de usar este recurso, se preferir você poderá compartilhar as impressoras individualmente. Para detalhes, veja “Configurando o Linux como um servidor de impressão Windows”.

**OBS:**É importante lembrar que a seção *[printers]* **DEVE** ser definida como *printable* usando o parâmetro *printable = yes* para funcionar. O utilitário **testparm** poderá ser usado para verificar problemas no arquivo `cd` configuração do SAMBA (veja “Buscando problemas na configuração”).

## Buscando problemas na configuração

Durante o processo de configuração do SAMBA, é comum cometer erros de digitação, usar parâmetros em lugares indevidos, etc. É recomendável o uso do **testparm** para checar a configuração do SAMBA sempre que houver modificações para ter certeza nada comprometa o funcionamento que planejou para sua máquina.

Para usar o **testparm** é só digitar `testparm`. Logo após executa-lo, analise se existem erros nas seções de configuração e te pedirá para pressionar <ENTER> para ver um dump do arquivo:

```
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[cdrom]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

A saída acima indica que está tudo OK com todas as configurações que foram realizadas no servidor. É possível especificar um outro arquivo de configuração do SAMBA usando `testparm /etc/samba/smb2.conf`.

Também é permitido simular o nome NetBIOS que fará acesso a máquina usando o parâmetro `-L nome` (que será substituído na variável `%L`).

## Níveis de sistema para eleição de rede

Para selecionar qual sistema NetBIOS será o local master browse ou domain master browse, é usado um método bastante interessante: o de eleições.

Quando uma nova máquina entra na rede NetBIOS, ela solicita quem é o Local Master Browser, caso nenhuma resposta, ela força uma eleição na rede através de uma requisição Broadcasting especial. Vence a eleição quem tiver o **\*\*\*maior número\*\*\***, chamado de OS Level (nível de sistema operacional). Caso duas máquinas empatem, o desempate é feito usando outros critérios.

Se você for a única máquina de um workgroup, automaticamente você será o Local Master Browser. De meia em meia hora uma nova eleição é feita, forçando mais tráfego broadcasting na rede. Durante este novo processo de eleição, a lista de computadores é atualizada; as novas máquinas são adicionadas e as desligadas saem da lista após 36 minutos. Este é o motivo porque as máquinas Windows continuam aparecendo no ambiente de rede por algum tempo mesmo depois que desligadas ou porque elas não aparecem de imediato.

O OS Level é um número que é característico de cada sistema operacional ficando entre 0 (mais baixo) e 255. Os níveis de acessos dos sistemas operacionais são os seguintes:

|                                         |    |
|-----------------------------------------|----|
| Windows for Workgroups                  | 1  |
| Windows 95                              | 1  |
| Windows 98                              | 2  |
| Windows 98 Second Edition               | 2  |
|                                         |    |
| Windows 2000 Server (standalone)        | 16 |
| Windows 2000 Professional               | 16 |
| Windows NT 4.0 Wks                      | 17 |
| Windows NT 3.51 Wks                     | 16 |
|                                         |    |
| Windows NT 3.51 Server                  | 32 |
| Windows NT 4.0 Server                   | 33 |
| Windows 2000 Server (Domain Controller) | 32 |
|                                         |    |
| SAMBA                                   | 32 |

O valor padrão do OS Level do SAMBA é 32, entretanto ele é bastante flexível para permitir sua mudança através do parâmetro "os level" (veja "Navegação no servidor/tipo de servidor"), isto garante que o SAMBA sempre vença as eleições da rede sobre qualquer outro sistema operacional.

No caso de um servidor que estiver configurado para ser o navegador de rede, assim que for iniciado ele solicitará uma eleição de rede. As regras são as mesmas, vence o que tiver o **\*maior\*** número. Este número pode ser configurado facilmente no SAMBA para que ele sempre vença as eleições de rede, tomando conta da lista de máquinas. Isto é especialmente interessante por causa da estabilidade do servidor **Linux**, quando migramos de servidor NT ou para fornecer mais serviços de navegação, como servidor WINS.

**OBS:** Nunca deixe um servidor NT configurado para ser o Local Browser ou Domain Master Browser competir com o SAMBA. Mesmo que o SAMBA ganhe, o NT é um péssimo perdedor e convoca uma nova eleição para tentar novamente se eleger, gerando um **\*extremo\*** tráfego broadcasting em redes grandes.

## Variáveis de substituição

Esta seção foi baseada nos dados da página de manual do samba, com adições que não estavam presentes na versão original e exemplos. Existem variáveis especiais que podem ser usadas no arquivo de configuração do samba e são substituídas por parâmetros especiais no momento da conexão do usuário. Um exemplo de utilização de variáveis de substituição seria mudar a localização do diretório home do usuário:

```
[homes]
comment = Diretório home do usuário
path = /home/usuarios/%u
```

Cada uma das variáveis são descritas em detalhes abaixo:

|    |                                                                                                                                                                                                                              |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %S | O nome do serviço atual, se existir. Seu uso é interessante, principalmente no uso de diretórios homes.                                                                                                                      |
| %P | O diretório raiz do serviço atual, se existir.                                                                                                                                                                               |
| %u | O nome de usuário do serviço atual, se aplicável. Esta variável é bastante útil para programação de scripts e também para criar arquivos de log personalizados, etc.                                                         |
| %g | O grupo primário do usuário %u.                                                                                                                                                                                              |
| %U | O nome de usuário da seção (o nome de usuário solicitado pelo cliente, não é uma regra que ele será sempre o mesmo que ele recebeu).                                                                                         |
| %G | O nome do grupo primário de %U.                                                                                                                                                                                              |
| %H | O diretório home do usuário, de acordo com %u.                                                                                                                                                                               |
| %v | A versão do Samba.                                                                                                                                                                                                           |
| %h | O nome DNS da máquina que está executando o Samba.                                                                                                                                                                           |
| %m | O nome NetBIOS da máquina do cliente. Isto é muito útil para log de conexões personalizados e outras coisas úteis.                                                                                                           |
| %L | O nome NetBIOS do servidor. Como o servidor pode usar mais de um nome no samba (alias), você poderá saber com qual nome o seu servidor está sendo acessado e possivelmente torna-lo o nome primário de sua máquina.          |
| %M | O nome DNS da máquina cliente.                                                                                                                                                                                               |
| %N | O nome do seu servidor de diretórios home NIS. Este parâmetro é obtido de uma entrada no seu arquivo auto.map. Se não tiver compilado o SAMBA com a opção <code>--with-automount</code> então este valor será o mesmo de %L. |
| %p | O caminho do diretório home do serviço, obtido de uma entrada mapeada no arquivo <code>auto.map</code> do NIS. A entrada NIS do arquivo <code>auto.map</code> é dividida na forma "%N:%p".                                   |
| %R | O nível de protocolo selecionado após a negociação. O valor retornado pode ser CORE, COREPLUS, LANMAN1, LANMAN2 ou NT1.                                                                                                      |

|                 |                                                                                                                                                                                                                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| %d              | A identificação de processo do processo atual do servidor.                                                                                                                                                                                                                                                     |
| %a              | A arquitetura da máquina remota. Somente algumas são reconhecidas e a resposta pode não ser totalmente confiável. O <b>samba</b> atualmente reconhece <i>Samba, Windows for Workgroups, Windows 95, Windows NT e Windows 2000</i> . Qualquer outra coisa será mostrado como "UNKNOWN" ( <i>desconhecido</i> ). |
| %I              | O endereço IP da máquina do cliente.                                                                                                                                                                                                                                                                           |
| %T              | A data e hora atual.                                                                                                                                                                                                                                                                                           |
| %(var_ambiente) | Retorna o valor da <i>variável de ambiente</i> especificada.                                                                                                                                                                                                                                                   |

## Compartilhamento de arquivos e diretórios

Esta seção documenta como disponibilizar arquivos e impressoras com o SAMBA e os parâmetros usados para realizar restrições de compartilhamento, modo que os dados serão disponibilizados e itens de performance. A maior parte destes parâmetros são empregados em serviços do SAMBA, mas nada impede que também sejam colocado na seção *[global]* do arquivo de configuração, principalmente quando isto é válido para diversos serviços compartilhados (veja “Seção *[global]*”).

## Descrição de parâmetros usados em compartilhamento

Abaixo o guia traz algumas das opções que podem ser usadas para controlar o comportamento do compartilhamento de arquivos por *serviços* no servidor SAMBA:

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| path          | <p>Indica o diretório que será compartilhado. Lembre-se que o usuário terá as permissões de acesso que ele teria caso estivesse logado no sistema como um usuário UNIX normal, exceto se estiver fazendo mapeamento para outros nomes de usuários (veja “Mapeamento de nomes de usuários”).</p> <p>Ex: path=/pub - Compartilha o diretório local /pub.</p> <p><b>OBS:</b> Quando não é definido um path, o diretório /tmp é usado como padrão.</p> |
| comment       | <p>Descrição do compartilhamento que será mostrada na janela de procura de rede ou no smbclient -L maquina.</p> <p>Ex: comment=Pasta de conteúdo público do sistema.</p>                                                                                                                                                                                                                                                                           |
| browseable    | <p>Define se o compartilhamento será ou não exibido na janela de procura de rede. Mesmo não sendo exibido, o compartilhamento poderá ser acessado. Veja “Criando um compartilhamento invisível” para uma explicação mais detalhada.</p> <p>Ex: browseable=yes - Lista o compartilhamento na janela de pesquisa de servidores.</p>                                                                                                                  |
| guest account | <p>Conta que será usada para fazer acesso sem senha (convidado) quando o parâmetro guest ok ou public forem usados em um compartilhamento. Por padrão ela é mapeada para o usuário nobody. É importante especificar uma nome de usuário guest (convidado), principalmente porque seu UID será usado para fazer várias operações no</p>                                                                                                             |

SAMBA, como exibir os recursos disponíveis na máquina para a rede. Por motivos claros, é recomendável que este usuário **não** tenha acesso login ao sistema.

Caso não tenha a intenção de ocultar o SAMBA na lista de máquinas da rede (fazendo apenas acesso direto aos recursos), especifique um valor para esta opção.

Ex: `guest account = sambausr` - Mapeia os usuário se conectando sem senha para o usuário `sambausr`, desde que o acesso `guest` seja permitido pela opção `public`.

`public` Permite aos usuários usuários se conectarem ao compartilhamento sem fornecer uma senha usando o usuário `guest`. O UID que o usuário `guest` será mapeado é especificado pelo parâmetro `guest account`). Veja “Criando um compartilhamento para acesso sem senha”. O parâmetro `guest ok` é equivalente a `public`.

Ex: `public = no` - Não permite

`guest only` Permite somente conexões `guest` ao recurso. O UID do usuário é mapeado para `guest`, mesmo que forneça uma senha correta. O valor padrão é `no`.

Ex: `guest only = no`.

`write list` Lista de usuários separados por espaço ou vírgula que poderão ler e gravar no compartilhamento. Caso o nome for iniciado por "@", o nome especificado será tratado como um grupo UNIX (`/etc/group`) e todos os usuários daquele grupo terão acesso de gravação. O uso deste parâmetro ignora o `read only = yes`. Veja “Excessão de acesso na permissão padrão de compartilhamento” para mais detalhes.

Ex: `write list = gleydson, @usuarios` - Permite acesso gravação somente do usuário `gleydson` e todos os usuários pertencentes ao grupo `@usuarios`.

**OBS:** - O significado de "@" nos parâmetros "invalid users"/"valid users" é diferente das opções `write list` e `read list`.

`read list` Lista de usuários separados por espaço ou vírgula que poderão apenas ler o compartilhamento. O caracter "@" pode ser especificado para fazer referência a grupos, como no `write list`. O uso deste parâmetro ignora o `read only = no`. Veja “Excessão de acesso na permissão padrão de compartilhamento” para mais detalhes.

Ex: `read list = nobody, system, operador, @usuarios` - Permite acesso de leitura somente do usuário `nobody`, `system`, `operador` e todos os usuários pertencentes ao grupo `@usuarios`.

`user` Especifica um ou mais nomes de usuários ou grupos (caso o nome seja seguido de "@") para checagem de senha. Quando o cliente somente fornece uma senha (especialmente na rede **Lan Manager, Windows for Workgroups** e primeira versão do **Windows 95**) ela será validada no banco de dados de senhas usando o usuário especificado nesta opção.

Ex: `user = john @usuariosrede`

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| only user       | <p>Especifica se somente serão permitidas conexões vindas de usuários da diretiva user. O padrão é no. Caso deseje restringir o acesso a determinados usuários, o certo é fazê-lo usando <code>valid users</code> e <code>invalid users</code> (veja “Restringindo o acesso por usuários”). O uso de <code>only user</code> é apropriado quando é necessário um controle específico de acesso sobre a diretiva user.</p> <p>Ex: <code>only user = no</code>.</p>                                                                                                                                                                                                                                                                 |
| locking         | <p>Permite ao SAMBA fazer um lock real de arquivo ou apenas simular. Caso seja especificado como "0", o arquivo não é bloqueado para acesso exclusivo no servidor mas uma resposta positiva de lock é retornada ao cliente. Se definido como "1", um lock real é feito. O padrão é yes.</p> <p>Ex: <code>locking = yes</code></p>                                                                                                                                                                                                                                                                                                                                                                                                |
| available       | <p>Faz o SAMBA ignorar o compartilhamento (como se tivesse retirado do servidor). O valor padrão é "no".</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| follow symlinks | <p>Permite o uso de links simbólicos no compartilhamento (veja também a opção <code>wide links</code>). A desativação desta opção diminui um pouco a performance de acesso aos arquivos. Como é restrita a compartilhamento, o impacto de segurança depende dos dados sendo compartilhados. O valor padrão desta opção é "YES".</p> <p>Ex: <code>follow symlinks = yes</code></p>                                                                                                                                                                                                                                                                                                                                                |
| wide links      | <p>Permite apontar para links simbólicos para fora do compartilhamento exportada pelo SAMBA. O valor padrão esta opção é "YES".</p> <p>Ex: <code>wide links = yes</code>.</p> <p><b>OBS:</b> - A desativação desta opção causa um aumento na performance do servidor SAMBA, evitando a chamada de funções do sistema para resolver os links. Entretanto, diminui a segurança do seu servidor, pois facilita a ocorrência de ataques usando links simbólicos.</p> <p>Lembre-se mais uma vez que a segurança do seu sistema começa pela política e uma instalação bem configurada, isso já implica desde a escolha de sua distribuição até o conhecimento de permissões e planejamento na implantação do servidor de arquivos.</p> |
| dont descend    | <p>Não mostra o conteúdo de diretórios especificados.</p> <p>Ex: <code>dont descend = /root, /proc, /win/windows, "/win/Arquivos de Programas", "/win/program files"</code>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| printable       | <p>Especifica se o compartilhamento é uma impressora (yes) ou um compartilhamento de arquivo/diretório (no). O padrão é "no".</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| read only       | <p>Especifica se o compartilhamento é somente para leitura (yes) ou não (no) para todos os usuários. O parâmetro <code>writable</code> é um antônimo equivalente a este parâmetro, só que utiliza as opções invertidas. Por segurança, o valor padrão é somente leitura. Veja uma explicação mais detalhada em “Criando um compartilhamento com acesso somente leitura”.</p> <p>Ex: <code>read only = yes</code>.</p>                                                                                                                                                                                                                                                                                                            |

---

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| create mask         | <p>Modo padrão para criação de arquivos no compartilhamento. O parâmetro "create mode" é um sinônimo para este. O modo de arquivos deve ser especificado em formato octal. Veja ???).</p> <p>Ex: <code>create mask = 0600</code>.</p>                                                                                                                                                                                       |
| directory mask      | <p>Modo padrão para a criação de diretórios no compartilhamento. O parâmetro "directory mode" é um sinônimo para este. O modo de diretório deve ser especificado em formato octal.</p> <p>Ex: <code>directory mask = 0700</code>.</p>                                                                                                                                                                                       |
| getwd cache         | <p>Permite utilizar um cache para acesso as requisições getwd, diminuindo o número de ciclos de processamento para acesso a arquivos/diretórios. O valor padrão é "Yes".</p>                                                                                                                                                                                                                                                |
| write cache size    | <p>Tamanho do cache de leitura/gravação do compartilhamento. Este valor é especificado em bytes e o padrão é "0". Veja "Melhorando a performance do compartilhamento/servidor" para detalhes sobre seu uso.</p> <p>Ex: <code>write cache size = 384000</code>.</p>                                                                                                                                                          |
| inherit permissions | <p>Permite herdar permissões de arquivos/diretórios do diretório pai quando novos arquivos/diretórios são criados, isto inclui bits SGID (set group ID). O padrão é NÃO herdar permissões. O uso desta opção substitui as opções fornecidas por <code>create mask</code>, <code>directory mask</code>, <code>force create mask</code> e <code>force directory mask</code>.</p> <p>Ex: <code>inherit permissions</code>.</p> |
| preexec             | <p>Executa um comando antes a abertura de um compartilhamento. O parâmetro <code>exec</code> é um sinônimo para este. Veja "Executando comandos antes e após o acesso ao compartilhamento".</p>                                                                                                                                                                                                                             |
| postexec            | <p>Executa um comando depois da utilização do compartilhamento. Veja "Executando comandos antes e após o acesso ao compartilhamento".</p>                                                                                                                                                                                                                                                                                   |
| preexec close       | <p>Fecha imediatamente o compartilhamento caso o valor do comando executado pela opção <code>preexec</code> seja diferente de 0. O uso desta opção só faz sentido em conjunto com <code>preexec</code>. O valor padrão é "no". Veja "Executando comandos antes e após o acesso ao compartilhamento".</p> <p>Exemplo: <code>preexec close = yes</code>.</p>                                                                  |
| volume = nome       | <p>Retorna o nome de volume especificado quando é feito o acesso ao compartilhamento. Isto é muito útil para instalações onde o serial do CD, disquete ou HD é verificado durante o acesso. Isto acontece com frequência em produtos de fabricantes proprietários como forma de evitar a execução ilegal do programa.</p>                                                                                                   |

## Configuração em Grupo de Trabalho

A configuração *grupo de trabalho* é o método mais simples para compartilhar recursos em uma rede e também é indicado quando se possui uma rede pequena (até 30 máquinas) pois o gerenciamento não é tão

complicado. Acima deste número, é recomendada a utilização da configuração de domínio para definição de políticas de acesso mais precisas pelo administrador e para manter o controle sobre os recursos da rede (veja “Configurando um servidor PDC no SAMBA”).

A configuração do nível de acesso por grupo de trabalho tem como características principais essa simplicidade na configuração e o controle de acesso aos recursos sendo feito pela máquina local através de senhas e controle de IP.

Quanto ao método de senhas, você pode optar tanto por usar senhas criptografadas (“Ativando o suporte a senhas criptografadas”) ou senhas em texto limpo (“Ativando o suporte a senhas em texto plano”).

Veja abaixo um exemplo explicado de configuração do **SAMBA** para grupo de trabalho:

```
[global]
netbios name = servidor
workgroup = focalinux
security = user
obey pam restrictions = yes
encrypt passwords = no
os level = 30
guest account = nobody
server string = servidor da rede
local master = true
domain master = false

[homes]
comment = Diretórios de usuários
create mask= 0700
directory mask = 0700
browseable = no

[tmp]
path = /tmp
comment = Diretório temporário do sistema
read only = yes
valid users = gleydson
public = no
```

Agora, verifique se existem erros na configuração com o comando **testparm** (“Buscando problemas na configuração”) e reinicie o **SAMBA** (“Iniciando o servidor/reiniciando/recarregando a configuração”). O nome do grupo de trabalho que a máquina pertencerá é `focalinux` (`workgroup = focalinux`). O nível de acesso usado neste exemplo é de usuário (`security = user`), para mais detalhes sobre este método, veja “Níveis de autenticação”. O parâmetro `local master` foi definido para `yes` para o **SAMBA** tentar ser o navegador local do grupo de trabalho (veja “Local Master Browser”).

Para testar se o servidor está funcionando, digite o seguinte comando:

```
smbclient -L servidor -U usuario
```

Digite a senha de usuário quando solicitado. O comando deverá listar os recursos da máquina, indicando que a configuração está funcionando corretamente. Se você é paranóico e está preocupado com a segurança da máquina, recomendo ler a “Controle de acesso ao servidor SAMBA”.

---

# Resolução de nomes de máquinas no samba

O Samba pode utiliza os seguintes métodos para resolução de nomes de máquinas na rede (“Nome de máquina (nome NetBios)”). Eles estão listados em ordem de prioridade do mais para o menos recomendável:

- `lmhosts` - Pesquisa primeiro o arquivo `/etc/samba/lmhosts` (veja “Arquivo `/etc/samba/lmhosts`” para detalhes sobre este arquivo).
- `host` - Faz a pesquisa no arquivo `/etc/hosts` e no DNS em busca do nome da máquina.
- `wins` - Pesquisa no servidor WINS especificado pelo parâmetro `wins server` do `smb.conf` (veja “WINS”).
- `bcast` - Envia um pacote para o endereço de broadcast de sua configuração de rede. Este geralmente deve ser o último método por gerar tráfego excessivo em uma rede com um considerável número de computadores.

A ordem que a resolução de nomes é feita pelo samba, pode ser modificada usando o parâmetro "name resolve order = [ordem]" no arquivo de configuração do samba (ex. `name resolve order = lmhosts host wins bcast`).

## Arquivo `/etc/samba/lmhosts`

Este arquivo é um banco de dados que mapeia o endereço IP com o nome NetBIOS de uma máquina, semelhante ao formato do `/etc/hosts`. Este arquivo é útil quando temos servidores que são acessados com frequência, quando servidores de rede estão em segmentos separados e não temos um servidor WINS entre os dois pontos para resolução de nomes, para definir máquinas WINS que serão acessados pela internet, etc. Para ter certeza da localização do arquivo `lmhosts` em sua máquina, digite `smbclient -d 3 -L localhost` e veja o diretório de pesquisa deste arquivo. Veja um exemplo de arquivo `lmhosts` em “Exemplo de `lmhosts` do UNIX”.

O uso do arquivo `lmhosts` evita o excesso de broadcasting na rede, pois a ordem padrão usada para a resolução de nomes do **samba**, procura primeiro resolver o nome procurando em arquivos `lmhosts`, depois usando `dns`, `wins` e `broadcast`. Dependendo do projeto de sua rede e como as máquinas resolvem os nomes, ele pode ser uma camada a mais de segurança contra um simples hijacking de servidor através de NetBEUI ou WINS (isso é evitado com o uso de domínios, veja “Configurando um servidor PDC no SAMBA”).

**OBS:** Note que em clientes **Windows** que estejam em outra subrede, é necessário o arquivo `\windows\lmhosts` apontando para um servidor PDC mesmo que ele esteja apontando para o servidor WINS, caso contrário, a máquina não efetuará o logon.

O formato do arquivo `lmhosts` do **Windows** é mais complexo do que o do **Linux** pois o sistema precisa de mais detalhes para resolver os nomes e tipos de máquinas no domínio. Veja o modelo `lmhosts.sam` em seu sistema **Windows** para compreender seu funcionamento.

## Exemplo de `lmhosts` do UNIX

O exemplo abaixo mapeia o endereço IP das máquinas (primeira coluna) com o respectivo nome de máquina (segunda coluna):

```
172.16.0.34 servarq
172.16.0.30 serverdom
```

```
192.168.5.2 servwins
172.16.0.3 servpdc
172.16.0.1 gateway
```

## Exemplo de `lmhosts` do Windows

O arquivo possui uma sintaxe idêntica a do `lmhosts` do UNIX, mas alguns parâmetros especiais são especificados para ajudar o **Windows** resolver algumas coisas que não consegue fazer sozinho (principalmente com relação a identificação de função de máquinas em redes segmentadas):

```
192.168.0.5 servarq
192.168.0.1 serverpdc #PRE #DOM:dominio
192.168.0.2 "serverwins \0x1e" #PRE
#INCLUDE \\serverpdc\lmhosts
```

A primeira entrada do arquivo é a tradicional, onde o nome da máquina NetBIOS é associada ao IP. A segunda utiliza dois parâmetros adicionais:

- #PRE - Faz a entrada ser carregada logo na inicialização e se tornando uma entrada permanente no cache NetBIOS.
- #DOM - Especifica que a máquina é um controlador de domínio. A máquina deverá ter sido configurada para a função de domínio, pois caso contrário isso simplesmente não funcionará.

Note que ambos #PRE e #DOM devem ser especificados em maiúsculas. O terceiro exemplo faz uma referência permanente (#PRE) a máquina servidora WINS *serverwins*. Neste exemplo é usada uma característica especial para especificar a ID hexadecimal da máquina na rede *1e*. O quarto utiliza um include para associar outro arquivo ao atual, útil quando temos um compartilhamento que distribui um arquivo `lmhosts` para diversas máquinas na rede. De preferência, utilize sempre uma diretiva #PRE para todas as máquinas especificadas na diretiva #INCLUDE em seu arquivo de configuração.

Para a especificação de ID de serviço manual, é necessário manter os 15 caracteres no nome da máquina (preenchendo os restantes com espaços, caso seja preciso). O último caracter é o código hexadecimal que identifica o serviço de rede (veja “nmblookup” para ver a lista de serviços e sua respectiva função).

**OBS:** Caso crie este arquivo em um editor de textos do **Linux**, não se esqueça de converter o arquivo para que contenha o CR+LF no final das linhas.

## WINS

Este é um serviço de resolução de nomes que funciona de forma semelhante ao DNS, só que voltado para o NetBIOS. Quando uma máquina cliente NetBIOS entra na rede, o servidor WINS pega seu nome e IP e inclui em uma tabela para futura consulta pelos clientes da rede.

Esta tabela consultada toda vez que um cliente NetBIOS solicita um nome de máquina, componentes do grupo de trabalho ou domínio na rede. Uma outra aplicação importante de um servidor WINS é permitir a resolução de nomes em pontos de redes que requerem roteamento, a simplicidade de um protocolo não roteável como o NetBIOS fica limitada a simplicidade das instalações de rede. Um servidor WINS pode ser instalado em cada ponta da rede e eles trocarem dados entre si e atualizar suas tabelas de nomes/grupos de trabalhos/IPs.

A resolução de nomes de máquinas será feita consultando diretamente a máquina WINS ao invés de broadcasting (que geram um tráfego alto na rede).

## Configurando o servidor WINS

Para ativar o servidor WINS no **samba**, inclua as seguinte linha na seção `[global]` do seu arquivo `/etc/samba/smb.conf`:

```
[global]
wins support = yes
wins proxy = no
dns proxy = no
max wins ttl = 518400
```

O parâmetro `wins proxy` pode ser necessário para alguns clientes antigos que tenham problemas no envio de suas requisições WINS. O `dns proxy` permite que o servidor WINS faça a pesquisa no DNS para localização de nomes de máquinas caso não exista no cache. Ambas as opções `wins support`, `wins proxy` e `dns proxy` tem como valor padrão não. Pronto, seu servidor samba agora suporta WINS. Fácil, prático e rápido :-)

Se estiver configurando uma subrede com `masquerade` para acesso a um PDC ou um servidor WINS, você terá que mexer no gateway central para apontar uma rota para o gateway `masquerade`. O motivo disto é porque o `masquerade` do **Linux** atua somente nos cabeçalhos, mas o IP da estação é enviada e processada pelo PDC para retornar uma resposta. Da mesma forma, este IP é registrado no servidor WINS para uso das estações de trabalho. Isto só vai ser resolvido quando for escrito um módulo de `contrack` para conexões SAMBA (até o lançamento do kernel 2.4.22, isso ainda não ocorreu).

**OBS1:** NUNCA configure mais de um servidor WINS em uma mesma rede.

**OBS2:** NÃO especifique o parâmetro `wins server` caso esteja usando o suporte a WINS.

## Configurando o Cliente WINS

Para os clientes da rede (**Linux, Windows, OS/2, etc.**) fazer uso das vantagens da resolução de nomes usando o WINS, é necessário configurar para que eles o utilizem para resolver os nomes de máquinas. Isto é feito da seguinte forma em cada um dos sistemas operacionais:

**Linux**            Adicione a linha `wins server = ip_do_servidor_WINS` na seção global do arquivo `/etc/samba/smb.conf`:

```
[global]
wins server = 192.168.1.1
```

Após isto, reinicie o servidor **samba**. Caso esteja executando o servidor via `inetd`, digite: `killall -HUP nmbd`. Se estiver rodando através de `daemons: /etc/init.d/samba restart`. Não é necessário reiniciar o computador!

**Windows 9x**      Clique com o botão direito sobre o ícone *Ambiente de Rede* e selecione propriedades. Na janela de configuração de rede clique na aba *Configuração*. Na lista que aparece selecione o protocolo TCP/IP equivalente a sua placa de rede local e clique em *Propriedades*.

Na tela de *Propriedades TCP/IP* clique em *Configurações WINS* e marque a opção *Ativar resolução WINS*. Digite o endereço do servidor WINS e clique em *Adicionar*.

**OBS:** Se utilizar um servidor DHCP em sua rede local e o endereço do servidor WINS também é oferecido através dele, você poderá marcar a opção *Usar DHCP para*

*resolução WINS*. Note que esta opção somente estará disponível se escolher a opção *Obter um endereço IP automaticamente* na tab *Endereços IP*. Clique em OK até fechar todas as telas e reinicie quando o computador perguntar :-)

## Servidor de data/hora

O samba pode atuar como um servidor de data/hora ajustando o horário de suas estações de trabalho com o servidor da rede.

As estações clientes poderão executar o comando **net** para sincronizar seu relógio durante a inicialização do Windows, ou durante o logon da rede através do script de logon, caso tenha configurado o servidor **samba** para logon em domínios NT.

## Configuração do serviço de data/hora no SAMBA

Para configurar o samba para atuar como servidor de data/hora de sua rede, adicione o seguinte parâmetro na seção global do arquivo de configuração `/etc/samba/smb.conf`:

```
[global]
time server = yes
```

Para sincronizar a data/hora das estações de trabalho usando o servidor samba, veja “Sincronizando a data/hora no Cliente”. Caso o seu servidor SAMBA também seja o servidor de autenticação PDC da rede, a melhor forma de se fazer isto é colocar o comando `net time \\servidor_SAMBA /set /yes` em um script que será executado pela estação.

**OBS** É recomendável instalar um cliente ntp para manter o relógio do servidor sempre atualizado, conseqüentemente mantendo a data/hora das estações também em sincronismo . .

## Sincronizando a data/hora no Cliente

Na estação cliente Windows, use o seguinte comando:

```
NET TIME \\SERVIDOR /WORKGROUP:GRUPO /SET /YES
```

Um local interessante para colocação deste comando é na pasta Iniciar da estação Windows, pois todos os comandos que estejam nesta pasta são executados quando o sistema é iniciado.

Exemplos:

- `net time \\linux /set /yes` - Sincroniza a hora com o servidor "\\linux" e não pede confirmação (/yes).
- `net time \\linux /WORKGROUP:pinguim /set /yes` - Sincroniza a hora com o servidor "\\linux" do grupo de trabalho pinguim (/WORKGROUP:pinguim) e não pede confirmação (/yes).

## Configuração em Domínio

Esta seção descreve todos os passos necessários para configurar um servidor de domínio PDC (*Primary Domain Control*) com perfis móveis e outros recursos que tornam úteis e seguras a administração de uma rede NetBEUI.

## Uma breve introdução a um Domínio de rede

Um domínio de rede consiste em uma máquina central chamada de PDC, que mantém o controle de todas as contas de usuários/grupos e permissões para acesso a rede NetBEUI. O acesso desta forma é centralizado, como vantagem disto você pode usar o nível de acesso por usuários nas máquinas, definindo quais usuários ou grupos terão acesso de leitura/gravação.

É permitido criar scripts de logon, assim comandos programados pelo administrador serão executados nas máquinas clientes durante o logon no domínio (veja “Criando Scripts de logon”).

O nome da máquina é protegido contra hijacking através de contas de máquinas que fazem parte do domínio (veja “Contas de máquinas de domínio”). Isto só é possível em clientes **Linux**, **Windows NT**, **Windows 2000** e **Windows XP**.

Você poderá usar perfis móveis, copiando todas as personalizações do seu desktop para qualquer máquina na rede que você faça o logon. Para o administrador, ele poderá definir políticas com o **Poledit** e outros programas que serão salvas junto com o perfil do usuário, valendo para qualquer máquina que ele se autentique na rede (veja “Criando Scripts de logon”).

Se você deseja iniciar logo a configuração do seu domínio, siga até “Configurando um servidor PDC no SAMBA”.

## Local Master Browser

É a máquina que ganhou a eleição no segmento local de rede (veja “Níveis de sistema para eleição de rede”). Logo que é declarada o *local master browser*, ela começa a receber via broadcasting a lista de recursos compartilhados por cada máquina para montar a lista principal que será retornada para outras máquinas do grupo de trabalho ou outras subredes que solicite os recursos compartilhados por aquele grupo.

Uma nova eleição é feita a cada 36 minutos ou quando a máquina escolhida é desligada.

## Domain Master Browser

Quando o local master browse é eleito no segmento de rede, uma consulta é feita ao servidor WINS para saber quem é o Domain Master Browse da rede para enviar a lista de compartilhamentos. A máquina escolhida como Local Master Browse envia pacotes para a porta UDP 138 do Domain Master e este responde pedindo a lista de todos os nomes de máquinas que o local master conhece, e também o registra como local master para aquele segmento de rede.

Caso tenha configurado sua máquina para ser o domain master browser da rede (também chamado de *controlador principal de domínio* ou PDC), ela tentará se tornar a máquina que terá a lista completa de recursos enviados pelos locais master browsers de cada segmento de rede. Um PDC também é o local master browse de seu próprio segmento de rede.

É possível ter mais de um domain master browse, desde que cada um controle seu próprio domínio, mas não é possível ter 2 domain master browsers em um mesmo domínio. Caso utilize um servidor WINS em sua rede, o PDC fará consultas constantes em sua base de dados para obter a lista de domínios registrados. O domínio é identificado pelo caracter *Ib* na rede (veja “nmblookup”).

**OBS:** O Windows NT configurado como PDC sempre tenta se tornar o domain master browser em seu grupo de trabalho. Não sendo possível retirar o Windows NT configurado como PDC do domínio (por alguma outra razão), a única forma será deixar ele ser o domain master browser. Se este for o caso, você poderá continuar lendo este documento para aprender mais sobre NetBIOS e talvez ainda mudar de idéia sobre manter o NT na rede após ver as características do SAMBA ;-)

## Configurando um servidor PDC no SAMBA

Esta é a parte interessante do guia, a prática. Para os administradores que conhecem através da experiência própria os problemas e definições do SAMBA, grande parte do guia foi apenas uma revisão (por favor, se faltou algo que acha interessante, me notifiquem que incluirei na próxima versão e colocarei uma nota no lançamento e na página com os devidos créditos :-))

Para configurar uma máquina para ser o PDC (*Controladora Principal de Domínio* ou *Primary Domain Control*), siga esta seqüência:

- Habilite o suporte a senhas criptografadas. Caso ainda não tenha feito isso, leia a seção “Ativando o suporte a senhas criptografadas”.
- Na seção *[global]*, insira/modifique os seguintes parâmetros:

```
; Identificação da máquina e domínio
netbios name = gleydson
workgroup = focalinux

;níveis de acesso e funções do servidor
security = user
domain master = yes
prefered master = yes
local master = yes

; senhas criptografadas
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd.db
```

Onde os parâmetros significam:

- `netbios name = gleydson` - Nome do computador. Este também será o nome usado pelas outras máquinas clientes quando for configurar o PDC (*controlador de domínio*).
- `workgroup = focalinux` - Nome do domínio que está criando. Todas as máquinas que pertencerem a este domínio, terão o nível de acesso definido pelo PDC. Note que o parâmetro `workgroup` também é usado ao especificar o nome do grupo de trabalho quando se é usado a configuração *grupo de trabalho* (“Configuração em Grupo de Trabalho”).
- `security = user` - Requerido para controle de acesso por domínio, já que é utilizado o controle de acesso local usando usuários e grupos locais.
- `domain master = yes` - Especifica se esta máquina está sendo configurada para ser o PDC da rede.

**OBS:** Por favor, certifique-se que não existe outro PDC no domínio. Veja “Domain Master Browser”. `prefered master = yes` - Força uma eleição com algumas vantagens para seu servidor ser eleito sempre como o controlador de domínio. Isto garante que a máquina **SAMBA** sempre seja o PDC. Veja “Navegação no servidor/tipo de servidor”. `local master = yes` - Define se a máquina será o controlador principal do grupo de trabalho local que ela pertence.

Pronto, agora teste se existem erros em sua configuração executando o comando **testparm** (“Buscando problemas na configuração”) e corrija-os se existir. Resta agora reiniciar o servidor **nmbd** para que todas as suas alterações tenham efeito. Para adicionar seus clientes a um domínio, veja “Contas de máquinas de domínio” e “Configurando clientes em Domínio”.

## Contas de máquinas de domínio

Uma conta de máquina de domínio garante que nenhum outro computador possa utilizar o mesmo nome de uma máquina confiável e assim utilizar os compartilhamentos que ela tem permissão. Os clientes **Windows NT**, **Windows XP** e **Windows 2000** precisam de uma conta de máquina para ter acesso ao domínio e seus recursos. A criação de uma conta de máquina é bastante semelhante a criação da conta de um usuário normal no domínio.

Existe uma coisa que precisa sempre ter em mente quando estiver configurando uma conta de máquina de domínio: Quando você cria uma conta para a máquina, ela entra e altera sua senha no próximo logon usando um "segredo" entre ela e o PDC, este segredo a identifica sempre como dona daquele nome NetBIOS, ou seja, até o primeiro logon no NT, outra máquina com o mesmo nome NetBIOS poderá ser a dona do netbios naquele domínio caso faça o logon no domínio. A única forma de se evitar isto é logar imediatamente no domínio NT assim que criar as contas de máquinas.

Existem duas formas para criação de contas de máquinas: manual e automática.

### Criando contas de máquinas manualmente

Para criar uma conta de domínio para a máquina master, siga estes 2 passos:

1. Crie uma conta de máquina no arquivo `/etc/passwd`:

```
useradd -g domainmac -c "Maquina de Dominio" -s /bin/false -d /dev/null master$
```

O comando acima cria uma conta para a máquina `master$` e torna ela parte do grupo `domainmac`. É necessário especificar o caracter `$` após o nome da máquina para criar uma conta de máquina no domínio, caso contrário o próximo passo irá falhar. Acredito que nas próximas versões do SAMBA seja desnecessário o uso do arquivo `/etc/passwd` para a criação de contas de máquina.

2. Crie uma conta de máquina no arquivo `/etc/samba/smbpasswd`:

```
smbpasswd -m -a master
```

Isto cria uma conta de máquina para o computador `master` no arquivo `/etc/samba/smbpasswd`. Note que a criação de uma conta de máquina é muito semelhante a criação de um usuário apenas precisa adicionar a opção `-m`. Quando for criar uma conta com o **smbpasswd** Não é necessário especificar `$` no final do nome da máquina.

O mais importante: Entre **IMEDIATAMENTE** no domínio após criar a conta de máquina usando a conta de administrador de domínio criada no SAMBA (veja "Criando uma conta de administrador de domínio")! como a máquina ainda não se autenticou pela primeira vez, qualquer máquina que tenha o mesmo nome e entre no domínio, poderá alocar o nome recém criado. A única forma de resolver este problema, é apagando a conta de máquina e criando-a novamente no domínio. Siga os passos de acordo com o sistema operacional em "Configurando clientes em Domínio" para colocar seus clientes em domínio.

**OBS1:** Como segurança, recomendo desativar a conta de máquina no `/etc/passwd` usando o comando `passwd -l conta`. Esta conta NUNCA deverá ser usada para login, isto deixa nossa configuração um pouco mais restrita.

**OBS2:** A localização do arquivo de senhas criptografadas do SAMBA pode ser modificado através da opção `smb passwd file` na seção `[global]` do arquivo `smb.conf`.

**OBS3:** Os que tem experiência com NT e Windows 2000 devem ter notado que este método é semelhante ao do *Server Manager* das ferramentas de gerenciamentos de servidores existentes no Windows.

## Criando contas de máquinas automaticamente

Através deste método, as máquinas clientes terão sua conta criada automaticamente assim que seja feita a entrada no domínio usando a conta do administrador de domínio no SAMBA. Este é o método recomendável de colocação de máquinas no domínio por ser mais prática ao invés do método manual. Note que normalmente isto funciona para o WinXP e Win2000 mas não funciona em redes com o NT4, devendo ser criadas contas de máquinas usando o método manual.

Para fazer a configuração automática, coloque a seguinte linha no arquivo `smb.conf` na seção `[global]`:

```
add user script = useradd -g domainmac -c "Maquina de Dominio" -s /bin/false -d /d
```

Assim, a conta de máquina será automaticamente criada quando o administrador fizer sua configuração no domínio (veja “Criando uma conta de administrador de domínio”). No SAMBA 3.0, a opção `add machine script` deverá ser usada no lugar de `add user script` para adicionar uma máquina no domínio.

## Criando uma conta de administrador de domínio

A conta de administrador do domínio é a conta que tem permissões para realizar operações de manutenção e administração de máquinas que compõem o domínio de rede. Com ela é possível, entre outras coisas, adicionar e remover máquina que compõem o domínio. Para especificar que contas de usuários do arquivo `/etc/samba/smbpasswd` que terão poderes administrativos, utilize a opção `domain admin group` ou `admin users` na seção `[global]` do arquivo `/etc/samba/smb.conf`.

O parâmetro `admin users` permite que todas as operações realizadas pelo usuário sejam feitas com poderes de usuário `root`. Isto é necessário porque o arquivo `smbpasswd` (usado para ajustar as contas de máquinas) normalmente tem permissões de leitura/gravação somente para `root`. O `domain admin group` permite que usuários específicos ou usuários do grupo especificado sejam parte do grupo de administradores do domínio para adicionar máquinas, etc. Por exemplo, para tornar o usuário `gleydson` com privilégios para adicionar/remover máquinas no domínio:

```
[global]
...
admin users = gleydson
ou
domain admin group = @admins gleydson
```

Isto permite que o usuário `gleydson` possa adicionar/remover máquinas do domínio NT (veja “Configurando clientes em Domínio”) entre outras tarefas. Por segurança, recomendo que coloque esta conta no `invalid users` de cada compartilhamento para que seja utilizada somente para fins de gerenciamento de máquinas no domínio, a menos que deseje ter acesso total aos compartilhamentos do servidor (nesse caso, tenha consciência do nível de acesso que esta conta possui e dos problemas que pode causar caso caia em mãos erradas).

**OBS1:** Tenha SEMPRE bastante cuidado com quem dará poderes de administrador de domínio, pois toda sua rede poderá ficar vulnerável caso os cuidados de administração não estejam em boas mãos.

**OBS2:** Em versões antigas do SAMBA, somente o usuário `root` tem poderes para adicionar máquinas no domínio usando o parâmetro `domain admins group`, devendo ser também adicionado no arquivo

`smbpasswd` para que possa fazer isto e obviamente não deverá estar listado em `invalid users`. Mesmo assim, existem outras formas explicadas no guia de se contornar o risco causado pela liberação de acesso do usuário `root`.

## Criando Scripts de logon

Uma dos recursos mais úteis em um domínio é a possibilidade de se executar comandos nas máquinas cliente quando fazem o logon no domínio. Desta forma, é possível instalar programas, executar anti-vírus, mapear compartilhamentos automaticamente no clientes, etc. A programação de scripts de logon é feita usando a linguagem em lote do DOS, com possibilidades de usar variáveis de ambiente, cópia de arquivos entre servidores, etc. O guia não irá abordar a programação em linguagem de lote, mas isto é simples de se encontrar na internet e mesmo a documentação que acompanha o próprio **Windows** é útil.

Para habilitar o recurso de scripts de logon na máquina, adicione os seguintes parâmetros no arquivo `smb.conf`:

```
[global]
domain logons = yes
logon script = logon.cmd

[netlogon]
 path = /pub/samba/netlogon
 read only = yes
 write list = ntadmin
```

Segue a descrição de cada parâmetro com detalhes importantes para a configuração e funcionamento do recurso de logon:

- `domain logons` - Deve ser definido para `yes` para ativar o recurso de logon scripts do SAMBA.
- `logon drive` é a unidade de disco que terá o homedir do usuário mapeado. Isto somente é usado por máquinas NT/2000/XP.
- `logon script` - Define qual é o script que será executado na máquina cliente quando fizer o logon. Ele deve ser gravado no diretório especificado pela opção `path` do compartilhamento `[netlogon]` (`/pub/samba/netlogon` no exemplo). Os scripts de logon podem ser tanto em formato `.bat` ou `.cmd`. Se for programar um script universal, é recomendável o uso do formato `.bat` por ser compatível tanto com **Win9X** e **WinNT**.

Um detalhe que deve ser lembrado durante a programação do script de logon é que ele **DEVE** seguir o formato DOS, ou seja, ter os caracteres `CR+LF` como finalizador de linhas. Para utilizar editores do UNIX para escrever este script, será necessário executar o programa **flip** (`flip -m -b arquivo`) ou **unix2dos** no arquivo para converte-lo em formato compatível com o DOS.

Segue abaixo um exemplo de script de logon que detecta quando o cliente é Windows 95/NT, ajusta a hora com o servidor e mapeia 2 unidades de disco:

```
@echo off
cls
rem Logon Script desenvolvido por Gleydson Mazioli
rem da Silva como modelo para o guia Foca GNU/Linux
rem
rem Este script pode ser utilizado para fins didáticos
```

```

rem e distribuído livremente de acordo com os termos
rem da GPL
rem
echo "Aguarde enquanto sua máquina efetua"
echo "o logon na rede do domínio focalinux."
rem
if %OS%==Windows_NT goto NT-2000
rem
echo "-----"
echo "SO: %OS%"
echo "Usuário: %USERNAME%"
echo "Grupo de Trabalho: %LANGROUP%"
echo "Servidor: %DOMINIO%"
echo "-----"
echo "Recuperando compartilhamentos"
rem mapeia o compartilhamento publico definido no servidor
net use e: \\gleydson\publico
echo "Sincronizando data/hora"
rem sincroniza a data/hora com o servidor
net time \\gleydson /set /yes
goto fim
rem
rem
:NT-2000
echo "-----"
echo "SO: %OS%"
echo "Usuário: %USERNAME%"
echo "Windows: %windir%"
echo "Logon de domínio: %LOGONSERVER%"
echo "-----"
echo "Recuperando compartilhamentos"
net use e: \\gleydson\publico /persistent:yes
echo "Sincronizando data/hora"
net time \\gleydson /set /yes
rem
rem
goto fim
rem
:fim

```

Note no exemplo acima que não podem haver linhas em branco, você deverá utilizar a palavra *rem* (comentário em arquivos em lote) em seu lugar. Note que existem diferenças entre o comando **net** do Windows 9x/ME e do NT, as variáveis também possuem um significado diferente entre estes 2 sistemas, isto explica a necessidade de se incluir um bloco separado detectando a existência de qual sistema está sendo efetuado o logon.

A lista completa de variáveis disponíveis para cada sistema operacional pode ser obtida colocando-se `set >c:\vars.txt` que gravará uma lista de variáveis disponíveis durante o logon no arquivo `c:\vars.txt` da máquina cliente.

**OBS:** Caso especifique um computador que contém o script de login, lembre-se de fazê-lo sempre com `\` ao invés de `/` para não ter incompatibilidade com o **Windows 95/3.11**.

**ATENÇÃO:** Lembre-se que copiar e colar pode não funcionar para este script. Leia novamente esta seção do guia se estiver em dúvidas.

## Configurando perfis de usuários

Os perfis permitem que os clientes utilizem o mesmo perfil em qualquer máquina que ele se autentique na rede. Isto é feito após a autenticação copiando os arquivos que contém os dados de personalização de usuários (`user.dat`, `NTuser.dat`) para a máquina local. Este processo também inclui a cópia de papéis de parede, links da área de trabalho, cache do IE, etc. Para configurar o recurso de perfis móveis no domínio, é necessário adicionar os seguintes parâmetros no seu arquivo `smb.conf`:

```
[global]
security = user
encrypt passwords = yes
domain logons = yes
logon drive = H:
logon path = \\%N\profilesNT\%u
logon home = \\%N\profiles\%u
preserve case = yes
short preserve case = yes
case sensitive = no

[profiles]
path = /pub/profiles
read only = no
create mask = 0600
directory mask = 0700

[profilesNT]
path = /pub/profilesNT
read only = no
create mask = 0600
directory mask = 0700
```

Segue a descrição dos parâmetros de detalhes para seu funcionamento:

- O parâmetro `domain logons = yes` especifica que o servidor será usado para fazer logons no domínio. Quando este parâmetro é definido para `yes`, a máquina automaticamente tentará ser o PDC.
- `logon path` e `logon home` definem (respectivamente) o diretório de logon do `/pub/profilesNT/usuario (NT)` e `/pub/profiles/usuario (Win95)` respectivamente. Durante o logon, a variável `%N` será substituída pelo nome do servidor (ou servidor de diretórios, se for o caso) e a variável `%u` pelo nome do usuário.

O sistema operacional de origem é detectado no momento da conexão. Isto significa que o usuário poderá ter 2 perfis diferentes, de acordo com o tipo de sistema operacional cliente que estiver conectando.

- O diretório `home` do usuário será mapeado para a unidade `H:` (`logon drive = h:`). O parâmetro `logon drive` somente é usado pelo NT/2000/XP.
- As opções `preserve case`, `short preserve case` e `case sensitive` permite que os nomes dos arquivos/diretórios tenham as letras maiúsculas/minúsculas mantidas, isto é requerido para os perfis.

O compartilhamento dos 2 perfis pode ser feito sem tantos traumas, mas isto não será explicado profundamente no guia pois o procedimento segue o mesmo padrão do NT sendo bastante documentado na internet.

Note que é possível definir um servidor separado para servir os perfis para um domínio modificando a variável %N para apontar direto para a máquina. Na máquina que armazenará os perfis, basta definir o nível de segurança por servidor (`security = server`) e o endereço IP do servidor de senhas (`password server = IP`).

**OBS1:** Os perfis só funcionam caso o servidor de perfis contenha a opção `security = user` e `encrypt passwords = yes` ou `security = server` e `password server = endereço_IP`. Caso tenha problemas, verifique se uma destas alternativas está correta.

**OBS2:** Quando utiliza o SAMBA com o Windows 2000 SP2, é necessário adicionar a opção `nt acl support = no` no compartilhamento [`profiles`], caso contrário, ele retornará um erro de acesso ao compartilhamento.

## Modificações de permissões de acesso pelos clientes do domínio

Um usuário do Windows NT (ou versões baseadas neste) pode modificar as permissões dos arquivos/diretórios que tem acesso através da caixa de diálogo de listas de acesso do NT, lembrando que estas permissões nunca substituirão as definidas pelo administrador local.

A opção "nt acl support" deverá estar definida para "yes" na seção [`global`] do arquivo de configuração, caso contrário você não terá acesso para mudar as permissões através de caixas de diálogo do NT. \

## Gerenciamento de senhas

Esta seção contém explicações sobre o gerenciamento de senhas no SAMBA.

### Ativando o suporte a senhas criptografadas

O uso de senhas criptografadas é um requisito quando você deseja configurar o SAMBA para ser um servidor PDC ou um cliente de um domínio. Quando utiliza senhas criptografadas, elas trafegam em formato seguro através da rede, dificultando a captura por outras pessoas. Em versões mais recentes do Windows (a partir da OSR/2 e NT 4 service pack3) o suporte a senhas criptografadas vem habilitado como padrão para login e utilização de serviços da rede. Não é recomendável desativar o uso de senhas criptografadas, mas se mesmo assim for necessário veja "Senhas criptografadas ou em texto puro?".

Quando usamos senhas criptografadas, elas são armazenadas no arquivo `/etc/samba/smbpasswd` ao invés do `/etc/passwd`, isto permite que possamos controlar as permissões de usuários separadamente das do sistema e diferenciar os logins do domínio dos logins do sistema (usuários que possuem shell). Caso tenha um servidor que já possua muitas contas de usuários acessando em texto plano, recomendo ler "Migrando de senhas texto plano para criptografadas" para facilitar o processo de migração de contas.

O utilitário **smbpasswd** é o programa utilizado para gerenciar este arquivo de senhas e também o status de contas de usuários/máquinas do domínio. Siga estes passos para ativar o uso de senhas criptografadas no SAMBA:

1. Edite o arquivo `/etc/samba/smb.conf` e altere as seguintes linhas na seção [`global`] para adicionar o suporte a senhas criptografadas:

```
[global]
encrypt passwords = true
smb passwd file = /etc/samba/smbpasswd
```



(muito útil em scripts). Apenas tenha cuidado para que esta senha não seja divulgada em seus arquivos/processos.

4. Reinicie o processo do SAMBA (veja “Iniciando o servidor/reiniciando/recarregando a configuração”).
5. Verifique se o suporte a senhas criptografadas está funcionando com o comando:

```
smbclient -L localhost -U gleydson
```

Substitua localhost pelo IP do servidor e gleydson pelo usuário. Caso obtenha a mensagem `session setup failed: NT_STATUS_LOGON_FAILURE` significa que a senha informada está incorreta.

## Migrando de senhas texto plano para criptografadas

No SAMBA, é possível fazer um processo de migração de senhas em texto plano de usuários para criptografadas sem que eles deixem de acessar o servidor durante esta mudança. Caso este seja seu caso, insira o parâmetro

```
update encrypted = yes
```

na seção `[global]` do seu arquivo de configuração `smb.conf`. A senha criptografada é definida assim que o usuário se logar usando sua senha em texto plano. Não se esqueça de desativar esta opção ou removê-la após o prazo necessário para que todas as senhas sejam trocadas.

## Adicionando usuários no smbpasswd

A adição de um usuário no `smbpasswd` segue duas etapas: primeiro é necessário adicioná-lo no sistema com o `adduser` e depois no samba com o `smbpasswd`. Você deve estar se perguntando qual a vantagem de se ter um arquivo separado de usuários se ainda é preciso criar o login nos dois arquivos; O **SAMBA** para fazer o controle de acesso aos arquivos utiliza além dos mecanismos tradicionais do NT, o controle de permissões a nível UNIX para manter os arquivos ainda mais restritos. Além disso, será necessário usuários e grupos para criação e acesso ao sistema.

1. Adicione um usuário no sistema com o comando:

```
useradd -g grupo-dominio -c "Usuário de Domínio" -s /bin/false -d /dev/null joa
```

Este comando adiciona o usuário "joao" no grupo `grupo-dominio` e não define nem uma shell, diretório home nem senha para este usuário. Isto mantém o sistema mais seguro e não interfere no funcionamento do SAMBA, pois somente é necessário para fazer o mapeamento de UID/GID de usuários com as permissões do sistema UNIX.

É interessante padronizar os usuários criados no domínio para um mesmo grupo para pesquisa e outras coisas.

2. Crie o usuário "joao" no SAMBA:

```
smbpasswd -a joao
```

Será solicitada a senha do usuário.

## Removendo usuários do `smbpasswd`

Utilize o comando `smbpasswd -x usuario` para remover um usuário do arquivo `smbpasswd`. Se desejar, você pode manter o usuário no `/etc/passwd` ou remove-lo com o `userdel`.

**OBS:** Removendo um usuário deste arquivo fará que ele não tenha mais acesso ao SAMBA. Utilize o comando `smbpasswd -a teste`

## Desabilitando uma conta no `smbpasswd`

Como administrador, pode ser necessário que precise desativar temporariamente uma conta de usuário por alguma situação qualquer (má utilização de recursos, dúvida se a conta está sendo usada, para que ele ligue reclamando de autenticação para ter aquela desejada conversa (hehe), etc.). Remover uma conta e novamente adiciona-la então não é uma situação muito prática. Utilize então o seguinte comando para desativar uma conta de usuário:

```
smbpasswd -d usuario
```

Quando a conta de usuário é desativada, uma flag "D" é adicionada às opções do usuário (junto com as opções "UX"). Veja "Habilitando uma conta no `smbpasswd`" para reativar a conta.

## Habilitando uma conta no `smbpasswd`

Uma conta desativada com o uso do comando `smbpasswd -d` pode ser novamente ativada usando:

```
smbpasswd -e usuario
```

## Alterando a senha de um usuário

O utilitário `smbpasswd` pode ser usado tanto para alterar a senha de usuários locais do **SAMBA** ou de uma conta em um servidor remoto (seja **SAMBA**, **NT**, **W2K**). Para alterar a senha de um usuário local, digite:

```
smbpasswd -U usuario
```

Lhe será pedida a antiga senha, a nova senha e a confirmação. Caso seja o usuário `root`, somente a nova senha e a confirmação. Isto é mecanismo de proteção para usuários que esquecem a senha ;-)

Para alterar a senha de um usuário remoto, utilize:

```
smbpasswd -r servidor -U usuario
```

Note que apenas foi adicionada a opção `-r servidor` comparado com a opção anterior. A diferença é que a senha antiga do usuário sempre será solicitada para troca (pois o `root` das 2 máquinas pode não ser o mesmo).

## Definindo acesso sem senha para o usuário

Para fazer um usuário acessar sem senha, use o comando:

```
smbpasswd -n usuario
```

Isto é completamente desencorajado e necessita que a opção `null passwords` da seção `[global]` no arquivo `smb.conf` esteja ajustada para `yes` (**que NÃO é o padrão**).

## Ativando o suporte a senhas em texto plano

Esta forma de autenticação é enviada por implementações NetBIOS antigas, como a encontrada no **Lan Manager**, **Windows for Workgroups** e **Windows 95 OSR1**. As versões mais novas destas implementações enviam a senha em formato criptografado, sendo necessário também usar o formato criptografado no SAMBA para que possa se autenticar (veja “Ativando o suporte a senhas criptografadas”).

Em “Senhas criptografadas ou em texto puro?” é feita uma comparação entre o uso de autenticação usando senhas em texto plano e senhas criptografadas. Em geral, o administrador prefere a utilização da autenticação usando texto plano quando deseja usar o `/etc/passwd` para autenticação e está usando grupos de trabalho é necessário usar senhas criptografadas para autenticação).

Para configurar o **SAMBA** para utilizar senhas em texto, modifique o parâmetro `encrypt passwords` para no:

```
[global]
encrypt passwords = no
```

Reinicie o **SAMBA** (“Iniciando o servidor/reiniciando/recarregando a configuração”) e a partir de agora, ele usará o `/etc/passwd` para autenticação.

**OBS:** Tenha certeza de não estar participando de um domínio ou que sua máquina seja o PDC antes de fazer esta modificação.

## Configurando o acesso de clientes para uso de senhas em texto plano

Esta seção descreve como configurar clientes para acessar o servidor **SAMBA** usando autenticação em texto plano. Atualmente o guia cobre os seguintes clientes:

- “Lan Manager”
- “Windows for Workgroups”
- “Windows 95 / Windows 95A”
- “Windows 95B”
- “Windows 98/98SE”
- “Windows ME”
- “Windows NT Server/WorkStation”
- “Windows 2000”
- “Linux”

Em cada seção, também é explicado como habilitar novamente a autenticação usando senhas criptografadas (se suportado pelo cliente).

### Lan Manager

Cliente NetBIOS para DOS. Ele trabalha somente com senhas em texto plano.

## Windows for Workgroups

Este é o padrão de autenticação do **Windows for Workgroups** caso não tenha feito nenhuma alteração específica (mas desconheço algo que faça-o trabalhar com senhas criptografadas).

## Windows 95 / Windows 95A

O **Windows 95** até a release "A", utiliza texto plano como padrão para autenticação (veja qual a release clicando com o botão direito em *Meu Computador e Propriedades*).

## Windows 95B

Copie o seguinte conteúdo para um arquivo chamado `win95-textoplano.reg`:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

Após isto, execute no **Windows 95** o seguinte comando: `regedit win95-textoplano.reg` e reinicie o computador para fazer efeito.

Para voltar a utilizar criptografia, apenas altere o valor `dword` para `00000000` no arquivo e execute novamente o **regedit**.

## Windows 98/98SE

O procedimento é idêntico ao "Windows 95B".

## Windows ME

O procedimento é idêntico ao "Windows 95B".

## Windows NT Server/WorkStation

Copie o seguinte conteúdo para um arquivo chamado `winNT-textoplano.reg`:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001
```

Após isto, execute no **Windows NT** o seguinte comando: `regedit winNT-textoplano.reg` e reinicie o computador para fazer efeito.

Para voltar a utilizar criptografia, apenas altere o valor `dword` para `00000000` no arquivo e execute novamente o **regedit**.

## Windows 2000

Copie o seguinte conteúdo para um arquivo chamado `win2000-textoplano.reg`:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation\Parameters
```

```
"EnablePlainTextPassword"=dword:00000001
```

Após isto, execute no **Windows 2000** o seguinte comando: `regedit win2000-textoplano.reg` e reinicie o computador para fazer efeito.

Para voltar a utilizar criptografia, apenas altere o valor `dword` para `00000000` no arquivo e execute novamente o **regedit**.

## Linux

Inclua/modifique a linha `encrypt passwords = no` no arquivo `smb.conf` e reinicie o **SAMBA**. Para voltar a utilizar criptografia, veja “Ativando o suporte a senhas criptografadas”.

## Mapeamento de usuários/grupos em clientes

O mapeamento de usuários do servidor remoto com a máquina local é usado quando você deseja controlar o acesso aos arquivos/diretórios a nível de usuário. No **Windows** isto permite que cada arquivo/diretório tenha o acesso leitura/gravação somente para os usuários definidos e autenticados no controlador de domínio. No **Linux** as permissões de arquivos e diretórios podem ser definidas para o usuário do PDC, garantindo o mesmo nível de controle de acesso.

Esta seção explica como configurar o mapeamento de UID/GID entre o servidor PDC SAMBA e seus clientes NetBIOS Windows e Linux.

## Mapeamento de usuários/grupos domínio em Windows

Para o **Windows** utilizar os usuários remotos do servidor para fazer seu controle de acesso por nível de usuário, siga os seguintes passos:

Windows 9X      Entre no *Painel de Controle/Propriedades de Rede* e clique na tab *Controle de Acesso*. Marque a opção *Controle de acesso a nível de usuário* e coloque o nome da máquina PDC na caixa de diálogo de onde os usuários/grupos serão obtidos. Você também pode colocar o nome do grupo de trabalho, neste caso a máquina fará uma busca pelo PDC ou outra máquina de onde pode obter os nomes de usuários/grupos.

**OBS:** Para fazer isto, você deverá estar autenticado no domínio.

## Mapeamento de usuários/grupos domínio em Linux

A associação de UIDs de usuários de um domínio com usuários locais no **Linux** é feita pelo programa **winbind**. Ele utiliza o mecanismo **nsswitch** para obter outras fontes de dados de usuários e os associa nas ferramentas de gerenciamento de contas existentes no sistema. Siga estes passos para fazer sua instalação e configuração do **Winbind** em um servidor **Linux**:

- Instale o programa **winbind**: `apt-get install winbind`.
- Modifique o arquivo `smb.conf` adicionando as seguintes linhas na seção `[global]`:

```
winbind separator = +
winbind cache time = 30
winbind uid = 10000-15000
winbind gid = 10000-12000
winbind enum users = yes
winbind enum groups = yes
template homedir = /home/winbind/%D/%U
```

template shell = /bin/false

Onde

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| winbind separator   | Separador usado para separar o nome dos grupos do nome de domínio. Este parâmetro somente tem sentido quando usado em conjunto com um PDC Windows ou quando os módulos <code>pam_winbind.so</code> e <code>nss_winbind.so</code> estão sendo utilizados.                                                                                                                                                                                                                                                                                   |
| winbind cache time  | Define a quantidade de tempo em segundos que um nome/grupo permanecerá no cache local para não ser feita uma nova consulta no servidor PDC.                                                                                                                                                                                                                                                                                                                                                                                                |
| winbind uid         | Especifica o intervalo que será usado para mapear os nomes de usuários remotos como UIDs locais. Você precisará ter certeza que nenhum UID nesse intervalo é usado no sistema, como pelo LDAP, NIS ou usuários normais. Por padrão, os IDS de usuários normais na maioria dos sistemas Linux, começam por 1000. No exemplo serão usados os UIDs de 10000 a 15000 para mapeamento e UIDs dos usuários do domínio para usuários locais.                                                                                                      |
| winbind gid         | Especifica o intervalo de GIDs que será usado para mapear os nomes de grupos remotos do domínio como GIDs locais. Como no parâmetro <code>winbind uid</code> , você deverá ter certeza que esta faixa de GIDs não está sendo usada em seu sistema.<br><br><b>OBS:</b> Atualmente <b>SAMBA</b> não possui suporte a grupos globais, apenas para usuários globais, desta forma os grupos da máquina remota não serão trazidos para o sistema. Uma forma de contornar isto, é utilizando o LDAP ou o NIS no PDC e nos clientes <b>Linux</b> . |
| winbind enum users  | Permite enumerar usuários do winbind para retornarem dados quando solicitados. A não ser que possua uma instalação parecida em todas as máquinas (como com o uso de LDAP e NIS) responda "yes" para não ter problemas.                                                                                                                                                                                                                                                                                                                     |
| winbind enum groups | Permite enumerar grupos do winbind para retornarem dados quando solicitados. A não ser que possua uma instalação parecida em todas as máquinas (como com o uso de LDAP e NIS) responda "yes" para não ter problemas.                                                                                                                                                                                                                                                                                                                       |
| template homedir    | Quando o sistema cliente for um Windows NT ou baseado, este diretório será retornado como diretório de usuário para o sistema. O parâmetro <code>%D</code> será substituído pelo nome do domínio e <code>%U</code> pelo nome de usuário durante a conexão.                                                                                                                                                                                                                                                                                 |
| template shell      | Este será o shell enviado para máquinas NT ou baseadas nele como shell usado para login. O valor usado foi <code>/bin/false</code> pois desabilita os logons, mas você poderá usar <code>/bin/sh</code> (ou algum outro shell) para efetuar conexões do comando <b>net</b> ou outras ferramentas NetBEUI ao servidor.                                                                                                                                                                                                                      |

- Reinicie o servidor **SAMBA**
- Edite o arquivo `/etc/nsswitch.conf` alterando a ordem de pesquisa de nomes de usuários e grupos do sistema local para a seguinte:

```
passwd: files winbind
group: files winbind
shadow: compat
```

- Agora, inicie o daemon **winbind** local com o comando: `/etc/init.d/winbind restart`.
- Entre no domínio com o comando: `smbpasswd -j domínio -r nome_do_PDC -U usuario` (veja "Linux" para aprender como entrar no domínio em caso de dúvidas).
- Agora faça o teste para obter a listagem dos grupos e usuários do domínio do PDC digitando:

```
wbinfo -u
wbinfo -g
getent passwd
getent group
```

Caso isto não aconteça, revise suas configurações e veja os logs procurando por erros quando o **winbind** tenta obter a lista de usuários/grupos do domínio.

Agora você deve ser capaz de criar diretórios/arquivos locais usando os nomes de usuários/grupos do domínio. Lembre-se de reiniciar sempre o **winbind** quando reiniciar o SAMBA por alguma modificação for feita (ao mesmo que saiba que não afeta o **winbind**), assim como entrar novamente no domínio, caso contrário o mapeamento deixará de funcionar.

**OBS:** Atualmente, o **winbind** não oferece suporte a restrições por data/hora de logon para estações de trabalho. Isto deverá ser implementado em uma futura versão

## Compartilhamento de impressão no servidor SAMBA

Este capítulo documenta como configurar o seu servidor samba para permitir o acesso a compartilhamento de arquivos e impressão no sistema.

### Configurando o Linux como um servidor de impressão Windows

Será necessário ter o pacote instalado e adicionar as seguintes linhas no seu arquivo `/etc/samba/smb.conf`:

```
[hp-printer]
path = /tmp
printer name=HP DeskJet 690C
printable = yes
print command = lpr -r -h -P %p %s
valid users = winuser winuser2
create mode = 0700
```

O compartilhamento acima tornará disponível a impressora local "lp" as máquinas Windows com o nome "HP DeskJet 690C". Uma impressora alternativa pode ser especificada modificando a opção `-P` da linha de comando do **lpr**. Note que somente os usuários "winuser" e "winuser2" poderão usar esta impressora.

Os arquivos de spool (para gerenciar a fila de impressão) serão gravados em /tmp (path = /tmp) e o compartilhamento [hp-printer] será mostrado como uma impressora (printable = yes).

Agora será necessário instalar o driver desta impressora no Windows (HP 690C) e escolher impressora instalada via rede e seguir os demais passos de configuração.

## Controle de acesso ao servidor SAMBA

Este capítulo documenta o controle de acesso ao servidor samba e restrições.

### Nível de acesso de usuários conectados ao SAMBA

Quando acessa um compartilhamento, o usuário do samba é mapeado com o UID respectivo de usuário do sistema ou o usuário guest (especificado pela opção "guest account") no caso de um acesso público. Quando isto ocorre, um processo filho do **smbd** é executado sobre o UID e GID deste usuário. Isto significa que em nenhuma ocasião o **SAMBA** dará mais permissões que as necessárias para o usuário (com exceção de quando é usado o parâmetro `admin users`, veja "Criando uma conta de administrador de domínio").

### Restringindo o acesso por IP/rede

Esta restrição pode ser feita pelos parâmetros `allow hosts` e `deny hosts` tanto em serviços individuais ou em todo o servidor. Os parâmetros `hosts allow` e `hosts deny` são equivalentes a estes acima. O `allow hosts` permite o acesso a máquinas especificadas como argumento. São permitidos os seguintes métodos para permitir o acesso a uma máquina/rede:

- 192.168.1.1 - IP da máquina
- servidor - Nome da máquina
- 192.168.1.0/255.255.255.0 - IP com máscara de rede
- 192.168.1.0/24 - IP com máscara de rede octal
- 192.168.1. - Porção de rede sem o host (como no `hosts.allow` e `hosts.deny`).
- @nome - Pesquisa por máquinas no grupo NIS.

É permitido usar mais de um endereço IP separando-os por vírgulas ou espaços. A palavra chave *EXCEPT* pode ser usada para fazer exceção de um ou mais endereços IPs, por exemplo:

```
hosts allow = 192.168.1. EXCEPT 192.168.1.20
```

Que permite o acesso a toda as máquinas da faixa de rede 192.168.1.0/24 exceto para a 192.168.1.20.

O `deny hosts` possui a mesma sintaxe do `allow hosts` mas bloqueia o acesso das máquinas especificadas como argumento. Quando o `allow hosts` e `deny hosts` são usados juntos, as máquinas em `allow hosts` terão prioridade (processa primeiro as diretivas em `allow hosts` e depois em `deny hosts`).

**OBS:** O endereço de loopback (127.0.0.1) nunca é bloqueado pelas diretivas de acesso. Provavelmente deve ter notado porque o endereço de loopback não pode ser bloqueado e as consequências disto para o SAMBA.

Se você está executando o SAMBA via `inetd`, os arquivos `hosts.allow` e `hosts.deny` são verificados antes do controle e acesso `allow hosts` e `deny hosts` para controle de acesso ao **smbd**. Caso

estiver usando o SAMBA *via* *inetd* e deseja restringir o acesso usando TCP Wrappers, veja “O mecanismo de controle de acessos tcpd”.

**OBS:** Lembre-se de usar o **testparm** para verificar a sintaxe do arquivo `smb.conf` sempre que desconfiar de problemas (veja “Buscando problemas na configuração”).

## Testando a restrição de Acesso por IP/Redes

Um método interessante e útil para testar se a nossa configuração vai bloquear o acesso a serviços é usando o **testparm** da seguinte forma:

```
testparm /etc/samba/smb.conf IP/host
```

Você precisará dizer para o **testparm** qual é o arquivo de configuração que está usando e o endereço IP/nome de host que fará a simulação de acesso. Este método não falsifica o endereço IP para testes, apenas usa os valores em *allow hosts* e *deny hosts* para checagem. Por exemplo, para verificar o acesso vindo do IP 192.168.1.50:

```
testparm /etc/samba/smb.conf 192.168.1.50
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[cdrom]"
Loaded services file OK.
Allow connection from /etc/samba/smb.conf (focalinux) to homes
Allow connection from /etc/samba/smb.conf (focalinux) to printers
Allow connection from /etc/samba/smb.conf (focalinux) to tmp
Allow connection from /etc/samba/smb.conf (focalinux) to cdrom
```

## Restringindo o acesso por interface de rede

Esta restrição de acesso permite que façamos o SAMBA responder requisições somente para a interfaces indicadas. O método de segurança descrito em “Restringindo o acesso por IP/rede” serão analisadas logo após esta checagem.

Para restringir o serviço SAMBA a interfaces, primeiro será necessário ativar o parâmetro `bind interfaces only` usando `1`, `yes` ou `true` (o padrão é desativado). Depois, definir que interfaces serão servidas pelo samba com o parâmetro *interfaces*. Os seguintes formatos de interfaces são permitidos:

- `eth0`, `sl0`, `plip0`, `etc` - Um nome de interface local. É permitido o uso de `*` para fazer o SAMBA monitorar todas as interfaces que iniciam com aquele nome (por exemplo, `eth*`).
- `192.168.1.1`, `192.168.1.2`, `etc` - Um endereço IP de interface local.
- `192.168.1.2/24`, `192.168.1.2/255.255.255.0` - Um par de endereço/máscara de rede.

Mais de uma interface pode ser usada separando-as com vírgula ou espaços. A escolha do uso de nome da interface ou do IP é feita de acordo com a configuração da máquina. Em uma máquina DHCP por exemplo, é recomendado o uso do nome da interface. Quando *bind interfaces only* estiver ativado, o padrão é esperar conexões em todas as interfaces que permitem broadcast exceto a loopback.

Exemplo:

```
bind interfaces only = 1
interfaces = loopback eth0
```

Permite o recebimento de requisições de acesso ao **SAMBA** somente da interface `loopback` (desnecessário, pois como notou durante a leitura, sempre é permitida a conexão) e `eth0`.

## Restringindo o acesso por usuários

Permite que você controle quem poderá ou não acessar o compartilhamento da máquina. Este controle é feito pelos parâmetros `valid users` e `invalid users`.

O `invalid users` lista de usuário que **NÃO** terão acesso ao compartilhamento. Se o nome for iniciado por "+" o parâmetro será tratado como um nome de grupo UNIX (`/etc/group`). O caracter "&" faz ele pesquisar o nome de grupo no banco de dados NIS. O caracter "@" permite fazer a busca do grupo primeiro no banco de dados NIS e caso ele não seja encontrado, no arquivo de grupos do sistema (`/etc/group`).

É possível usar a combinação de caracteres "+&" e "&+" para alternar a ordem de busca entre o `/etc/group` e o `NIS`.

Exemplos:

|                                                        |                                                                                                                                                                                                                                                 |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>invalid users = junior, marcio, +badusers</code> | Não permite que os usuários especificados e os usuários do grupo <code>+badusers</code> tenham acesso ao compartilhamento.                                                                                                                      |
| <code>invalid users = &amp;;semacesso</code>           | Bloqueia o acesso de todos os usuários NIS que pertençam ao grupo <code>semacesso</code> .                                                                                                                                                      |
| <code>invalid users = bruno, henrique, +@users,</code> | Bloqueia o acesso dos usuários <code>bruno</code> , <code>henrique</code> e de todos os usuários que pertençam ao grupo <code>users</code> . A pesquisa de grupo é feita primeiro no <code>/etc/group</code> e em seguida no <code>NIS</code> . |
| <code>invalid users = @semacesso</code>                | Bloqueia o acesso dos usuários que pertencem ao grupo "semacesso". A pesquisa é feita primeiro no <code>NIS</code> e depois no <code>/etc/group</code> (equivalente ao uso de "&+").                                                            |

O `valid users` possui a mesma sintaxe de funcionamento do `invalid users`, mas permite somente o acesso para os usuários/grupos listados. Caso a opção `valid users` não seja especificada ou a lista esteja vazia, o acesso é permitido. Se um mesmo nome de usuário estiver na lista `valid users` e `invalid users`, o padrão é ser mais restritivo, negando o acesso.

```
valid users = gleydson, michelle, geo
```

A segurança deste método de acesso depende muito da forma de autenticação dos nomes antes de passar o controle para o **SAMBA**, pois uma autenticação fraca põe em risco a segurança da sua máquina.

## Evite o uso do parâmetro *hosts equiv!*

Este parâmetro permite que máquinas tenham acesso sem senha a um servidor. Isto pode se tornar um **\*ENORME\*** buraco na segurança do seu sistema, pois mesmo usando uma senha inválida, a máquina poderá ter acesso a todos os recursos do compartilhamento e não é complicado fazer um ataque usando `DNS spoofing`.

Se realmente deseja fazer isto, tenha em mente os dados que poderão ser acessados daquela máquina, se realmente não existe nenhuma outra forma de disponibilizar o acesso de forma que mantenha o controle

de restrições (usando todos os outros métodos), restrinja o acesso usando MAC Address com o **iptables** ou o **arp** (veja “Restrições por MAC Address/IP”). O padrão é não usar nenhum arquivo `hosts.equiv`.

## Evite o uso de senhas em branco!

O parâmetro `null passwords` é usado na seção `[global]` permitindo que contas de usuários sem senha tenham acesso permitido ao servidor. **ISTO É TOTALMENTE INSEGURO** e deve ser sempre evitado. Caso você tenha feito uma bela restrição em sua máquina e deseja que o seu shell script de cópia de arquivos funcione usando este método, você está jogando toda a segurança do seu sistema por ralo abaixo.

Não existe motivo para usar senhas em branco em um controle de acesso por usuário, a não ser que precise testar algo realmente temporário e que depurando algo no SAMBA.

## Criando um compartilhamento para acesso sem senha

Em algumas situações (mesmo em instalações seguras) é preciso tornar um compartilhamento acessível publicamente, exemplos disto incluem um diretório que contém drivers de impressoras, arquivos comuns, um diretório temporário, etc.

Para configurar um acesso público utilizamos a opção `public = yes` ou `guest ok = yes` (que é um sinônimo para o último comando). O UID utilizado no acesso público é especificado pelo parâmetro `guest account`, portanto ele deverá ser um usuário válido do sistema. Caso você queira somente definir acesso `guest` a um compartilhamento, especifique a opção `guest only` para o serviço, desta forma, mesmo que o usuário tenha acesso, ele será mapeado para o usuário `guest`.

Uma boa medida de segurança é usar o usuário `nobody` pois a maioria das distribuições de **Linux** seguras adotam-o como padrão como usuário que não é dono de quaisquer arquivos/diretórios no sistema, não possui login, senha ou sequer um diretório home.

Veja um exemplo disponibilizando o compartilhamento `[download]` para acesso público com acesso a gravação:

```
[global]
guest account = nobody
..
..

[download]
path = /downloads
comment = Espaço público para abrigar downloads de Usuários
guest ok = yes (aqui poderá ser também "public = yes").
writable = yes
follow symlinks = false
```

O parâmetro `guest account` também poderá ser especificado no compartilhamento, isto é útil quando não quiser que o usuário que acesse o compartilhamento não seja o mesmo usado na diretiva `[global]`.

Caso seu servidor somente disponibiliza compartilhamentos para acesso público, é mais recomendado utilizar o nível `security = share` pra diminuir a carga máquina, pois o usuário `guest` será o primeiro a ser checado pelas regras de acesso (ao contrário do nível `user`, onde o acesso `guest` é o último checado).

**OBS:** Lembre-se que o compartilhamento funciona de modo recursivo, ou seja, todos os arquivos e subdiretórios dentro do diretório que compartilhou serão disponibilizados, portanto tenha certeza da

importância dos dados que existem no diretório, verifique se existem links simbólicos que apontam para ele, etc. Recomendo dar uma olhada rápida em “Considerações de segurança com o uso do parâmetro `public = yes`”.

## Criando um compartilhamento com acesso somente leitura

Esta proteção é útil quando não desejamos que pessoas alterem o conteúdo de um compartilhamento. Isto pode ser feito de duas formas: negando o acesso de gravação para todo o compartilhamento ou permitindo leitura somente para algumas pessoas. O parâmetro usado para fazer a restrição de acesso somente leitura é o `read only = yes` ou seu antônimo `writable = no`. Abaixo seguem os dois exemplos comentados:

```
[teste]
comment = Acesso a leitura para todos
path = /tmp
read only = yes
public = yes
```

No exemplo acima, o diretório `/tmp` (`path = /tmp`) foi compartilhado com o nome `teste` (`[teste]`), de forma pública (acesso sem senha - `public = yes`), e todos podem apenas ler seu conteúdo `read only = yes`.

```
[teste]
comment = Acesso a gravação para todos com excessões
path = /tmp
read only = no
read list = @users, gleydson
invalid users = root
```

Neste, o mesmo compartilhamento `teste` (`[teste]`) foi definido como acesso leitura/gravação para todos (`read only = no`), mas os usuários do grupo `@users` e o usuário `gleydson` terão sempre acesso leitura (`read list = @users, gleydson`). Adicionalmente foi colocada uma proteção para que o superusuário não tenha acesso a ele (`invalid users = root`). Esta forma de restrição é explicada melhor em “Excessão de acesso na permissão padrão de compartilhamento”.

## Criando um compartilhamento com acesso leitura/gravação

Esta forma de compartilhamento permite a alteração do conteúdo do compartilhamento dos usuários que possuem as permissões de acesso apropriadas. Este controle pode ser feito de duas formas: Acesso total de gravação para os usuários e acesso de gravação apenas para determinados usuários. Este controle é feito pela opção `read only = no` e seu antônimo equivalente `writable = yes`. Abaixo dois exemplos:

```
[teste]
comment = Acesso de gravação para todos.
path = /tmp
writable = yes
public = yes
```

No exemplo acima, o diretório `/tmp` (`path = /tmp`) foi compartilhado com o nome `teste` (`[teste]`), de forma pública (acesso sem senha - `public = yes`) e todos podem ler/gravar dentro dele (`writable = yes`).

```
[teste]
comment = Acesso a leitura para todos com excessões
path = /tmp
writable = no
write list = @users, gleydson
```

Neste, o mesmo compartilhamento teste (*[teste]*) foi definido como acesso de leitura para todos (*writable = no*), mas os usuários do grupo *@users* e o usuário *gleydson* serão os únicos que terão também acesso a gravação (*write list = @users, gleydson*). Esta forma de restrição é explicada melhor em “Excessão de acesso na permissão padrão de compartilhamento”.

## Excessão de acesso na permissão padrão de compartilhamento

É possível alterar o nível de acesso para determinados usuários/grupos em um compartilhamento, para entender melhor: Caso tenha criado um compartilhamento somente leitura e queira permitir que apenas alguns usuários ou grupos tenham acesso a gravação, isto é possível e será explicado nesta seção. Este comportamento é controlado por duas opções: *read list* e *write list*. Veja alguns exemplos:

```
[temporario]
comment = Diretório temporário
path = /tmp
writable = yes
read list = gleydson, root
browseable = no
available = yes
```

Neste exemplo, disponibilizamos o diretório */tmp* (*path = /tmp*) como compartilhamento de nome temporario (*[temporario]*), seu acesso padrão é leitura/gravação para todos (*writable = yes*), exceto para os usuários *root* e *gleydson* (*read list = root, gleydson*). Em adição, tornamos o compartilhamento *invisível* (veja “Criando um compartilhamento invisível”) no “Ambiente de Rede” do Windows (*browseable = no*) e ele será lido e disponibilizado pelo SAMBA (*available = yes*).

```
[temporario]
comment = Diretório temporário
path = /tmp
writable = no
write list = gleydson, @operadores
browseable = yes
```

Neste exemplo, disponibilizamos o diretório */tmp* (*path = /tmp*) como compartilhamento de nome temporario (*[temporario]*), seu acesso padrão é apenas leitura para todos (*writable = no*), exceto para o usuário *gleydson* e usuários do grupo Unix *operadores*, que tem acesso a leitura/gravação (*write list = gleydson, @operadores*). Tornamos o compartilhamento *visível* no “Ambiente de Rede” do Windows (*browseable = yes* - que é o padrão).

## Restringindo o IPC\$ e ADMIN\$

É seguro restringir os serviços IPC\$ e ADMIN\$ para acesso somente pelas faixas de rede de confiança. Isto pode ser feito através da mesma forma que a restrição em outros compartilhamentos. Os efeitos desta

restrição serão que somente as redes autorizadas possam obter a lista de máquinas, se autenticar no domínio e realizar tarefas administrativas gerais:

```
[IPC$]
read only = yes
allow from 192.168.1.0/24
```

```
[ADMIN$]
read only = yes
allow from 192.168.1.0/24
```

O exemplo acima permite que os serviços IPC\$ e ADMIN\$ sejam acessados de qualquer máquina na faixa de rede 192.168.1.0/24. Para forçar a autenticação para acesso a estes serviços:

```
[IPC$]
invalid users = nobody
valid users = gleydson michelle
read only = yes
allow from 192.168.1.0/24
```

```
[ADMIN$]
invalid users = nobody
valid users = gleydson michelle
read only = yes
allow from 192.168.1.0/24
```

Os exemplos acima são similares ao de antes, mas o acesso a listagem dos compartilhamentos é restringida (*invalid users = nobody*), pois o usuário *nobody* (usado para mostrar o compartilhamento) tem o acesso negado. Somente os usuários *gleydson* e *michelle* (*valid users = gleydson michelle*) podem listar seu conteúdo.

**OBS:** Mesmo que estejam restritos, os serviços IPC\$ e ADMIN\$ sempre poderão ser acessados de 127.0.0.1, ou teríamos problemas com o funcionamento do SAMBA. Assim não é necessário colocar 127.0.0.1 na lista de IPs autorizados.

## Criando um compartilhamento invisível

Para não exibir um compartilhamento da lista de compartilhamentos das máquinas, utilize o parâmetro `browseable = no`. Por exemplo:

```
[teste]
path = /tmp
comment = Diretório temporário
read only = yes
browseable = no
```

Neste exemplo, o diretório `/tmp` (*path = /tmp*) foi compartilhado através de `teste` (*[teste]*) com acesso somente leitura (*read only = yes*) e ele não será mostrado na listagem de compartilhamentos do ambiente de rede do Windows (*browseable = no*).

Note que o compartilhamento continua disponível, porém ele poderá ser acessado da estação Windows, especificando a `\\maquina\compartilhamento`. Para acessar o compartilhamento do exemplo acima:

```
Clique em Iniciar/Executar e digite:
\\nome_do_servidor_samba\teste
```

Ao contrário das máquinas **Windows** onde é necessário adicionar um "\$" do nome de compartilhamento para criar um compartilhamento oculto (como teste\$) o SAMBA cria um compartilhamento **realmente** oculto, não aparecendo mesmo na listagem do **smbclient**.

## Executando comandos antes e após o acesso ao compartilhamento

Este recurso oferece uma infinidade de soluções que podem resolver desde problemas de praticidade até segurança usando as opções `preexec` e `postexec`. Por exemplo, imagine que esteja compartilhando 4 unidades de CD-Rom de um servidor na rede, e deseje que estes CDs estejam sempre disponíveis mesmo que algum operador engraçadinho tenha ejetado as gavetas de propósito, podemos fazer a seguinte configuração:

```
[cdrom]
path = /cdrom
comment = Unidade de CD-ROM 1
read only = yes
preexec = /bin/mount /cdrom
preexec close = yes
postexec = /bin/umount /cdrom
```

Na configuração acima, o CD-ROM será compartilhado como `cdrom` (*[cdrom]*), somente leitura (*read only = yes*), quando o usuário acessar o compartilhamento ele "fechará" a gaveta do CD (*preexec = /bin/mount /cdrom*) e desmontará o drive de CD assim que o compartilhamento for fechado (*postexec = /bin/umount /cdrom*). Adicionalmente, caso o comando **mount** da opção `preexec` tenha retornado um valor diferente de 0, a conexão do compartilhamento é fechada (*preexec close = yes*).

A UID do processo do `preexec` e `postexec` será o mesmo do usuário que está acessando o compartilhamento, por este motivo ele deverá ter permissões para montar/desmontar o CD-ROM no sistema. Caso precise executar comandos como usuário `root`, utilize a variante `root preexec` e `root postexec`. Apenas tenha consciência que os programas sendo executados são seguros o bastante para não comprometer o seu sistema.

Usando a mesma técnica, é possível que o sistema lhe envie e-mails alertando sobre acesso a compartilhamentos que em conjunto com um `debug level 2` e logs configurados independentes por máquina, você possa ver o que a máquina tentou acessar (e foi negado) e o que ela conseguiu acesso.

Como bom administrador, você poderá criar scripts que façam uma checagem de segurança no compartilhamento e encerre automaticamente a conexão caso seja necessário, montar um "honey pot" para trojans, etc.

Como deve estar notando, as possibilidades do SAMBA se estendem além do simples compartilhamento de arquivos, se integrando com o potencial dos recursos do sistema UNIX.

## Considerações de segurança com o uso do parâmetro "public = yes"

Este parâmetro permite que você acesso um compartilhamento sem fornecer uma senha, ou seja, que o usuário não esteja autenticado. **NÃO** utilize o parâmetro "public = yes" (ou um de seus sinônimos) no

compartilhamento [homes], pois abrirá brechas para que possa acessar o diretório home de qualquer usuário e com acesso a gravação (que é o padrão adotado pelos administradores para permitir o acesso ao seu diretório home remoto).

Recomendo utilizar o parâmetro `public = yes` somente em compartilhamentos onde é realmente necessário, como o [netlogon] ou outras áreas de acesso público onde as permissões do sistema de arquivos local estejam devidamente restritas. Outra medida é não utilizar a opção `follow symlinks`, que poderá lhe causar problemas com usuários mal intencionados que tenham acesso shell.

**OBS:** Tenha em mente todas as considerações de segurança abordadas neste capítulo, bem como as permissões de acesso ao sistema Unix e como elas funcionam. A disponibilidade de arquivos em uma rede é simples, simples também pode ser o acesso indevido a eles caso não saiba o que está fazendo.

## Senhas criptografadas ou em texto puro?

Como regra geral, prefira sempre utilizar senhas criptografadas. Aqui alguns motivos:

- A senha é enviada de uma forma que dificulta sua captura por pessoas maliciosas.
- O NT não permite que você navegue no ambiente de rede em um sistema SAMBA com nível de acesso por usuário autenticando usando senhas em texto plano.
- Será solicitada sempre a senha para reconexão em cada compartilhamento da máquina.
- Todas as versões de Windows NT 4 a partir SP3 e Windows 95 OSR/2 utilizam senhas criptografadas como padrão. É possível fazê-lo utilizar senhas em texto plano modificando chaves no registro das máquinas clientes (veja “Ativando o suporte a senhas em texto plano” para detalhes).

As vantagens da utilização da autenticação usando texto plano:

- A senha utilizada será a mesma do `/etc/passwd` (servindo para ftp, login, etc)
- O servidor PDC pode ser usado para logon desde que os clientes estejam usando senhas em texto plano.
- Elas não são armazenadas no disco da estação cliente.
- Você não será perguntado por uma senha durante cada reconexão de recurso.

Antes de optar por utilizar um sistema de senhas em texto plano, leve em consideração estes pontos. Se você já utiliza telnet ou ftp, provavelmente a utilização de autenticação usando texto plano no SAMBA não trará problemas mais graves para você.

**OBS:** Caso seu NT ou versão derivada não navegue no ambiente de rede (só aceitando conexões especificando diretamente o “\\servidor\compartilhamento”) modifique sua configuração do SAMBA para autenticar usando senhas criptografadas (veja “Ativando o suporte a senhas criptografadas”) para detalhes de como fazer isto.

## Mapeamento de nomes de usuários

Este recurso faz a mapeamento (tradução) de nomes de usuários usados no momento do acesso para contas de acesso locais, bastante útil quando o nome de usuário enviado pela máquina não confere com NENHUMA conta local do sistema (um exemplo é quando o login do usuário no **Windows** é diferente de seu Login no **Linux**). Outra vantagem de seu uso é permitir que uma categoria de usuários utilizem um mesmo nível de acesso no sistema.

Seu formato é o seguinte: `username map = arquivo`.

As seguintes regras são usadas para construir o arquivo de mapeamento de nomes:

- Um arquivo de múltiplas linhas onde o sinal de "=" separa os dois parâmetros principais. O arquivo é processado linha por linha da forma tradicional, a diferença é o que o processamento do arquivo continua mesmo que uma condição confira. Para que o processamento do resto do arquivo seja interrompido quando um mapeamento confira, coloque o sinal "!" na frente do nome local.
- O parâmetro da esquerda é a conta Unix local que será usada para fazer acesso ao compartilhamento. Somente uma conta Unix poderá ser utilizada.
- O parâmetro da direita do sinal de "=" pode conter um ou mais nomes de usuários separados por espaços que serão mapeados para a conta Unix local.

O parâmetro "@grupo" permite que usuários pertencentes ao grupo Unix local sejam mapeados para a conta de usuário do lado esquerdo. Outro caracter especial é o "\*" e indica que qualquer usuário será mapeado.

Você pode utilizar comentários na mesma forma que no arquivo de configuração `smb.conf`. Alguns exemplos:

```
Mapeia o usuário "gleydson mazioli" com o usuário local gleydson
gleydson = gleydson mazioli

Mapeia o usuário root e adm para o usuário nobody
nobody = root adm

Mapeia qualquer nome de usuário que pertença ao grupo smb-users para o usuário
samba.
samba = @smb-users

Utiliza todos os exemplos anteriores, se nenhum usuário conferir, ele será
mapeado para o usuário nobody (como o usuário root e adm já são mapeados
para "nobody", este exemplo terá o mesmo efeito).
!gleydson = gleydson mazioli
!samba = @smb-users
nobody = *
```

## Melhorando a performance do compartilhamento/servidor

Esta seção trará algumas formas de otimização do servidor SAMBA que fazem diferença quando os valores adequados são utilizados: A primeira é a ativação de um cache de gravação/leitura de arquivos. Este cache é feito pela opção `write cache size` e funciona fazendo o cache dos arquivos que serão lidos/gravados. Ele é esvaziado assim que o arquivo for fechado ou quando estiver cheio. O valor especificado nesta opção é em bytes e o padrão é "0" para não causar impacto em sistemas com pouca memória (ou centenas de compartilhamentos). Exemplo:

```
[publico]
path = /pub
comment = Diretório de acesso público
read only = yes
```

```
public = yes
write cache size = 384000
```

Compartilha o diretório /pub (*path = /pub*) como compartilhamento de nome público (*[publico]*), seu acesso será feito como somente leitura (*read only = yes*) e o tamanho do cache de leitura/gravação reservado de 384Kb (*write cache size = 384000*).

Deixar a opção para seguir links simbólicos ativada (*follow symlinks*) garante mais performance de acesso a arquivos no compartilhamento. A desativação da opção *wide links* em conjunto com o uso de cache nas chamadas *getwd* (*getwd cache*) permite aumentar a segurança e tem um impacto perceptível na performance dos dados.

A desativação da opção global *nt smb support* também melhora a performance de acesso dos compartilhamentos. Esta é uma opção útil para detectar problemas de negociação de protocolo e por padrão, ela é ativada.

Caso utilize um valor de depuração de log muito alto (*debug level*), o sistema ficará mais lento pois o servidor sincroniza o arquivo após cada operação. Em uso excessivo do servidor de arquivos, isso apresenta uma degradação perceptível de performance.

A opção *prediction* permite que o SAMBA faça uma leitura adiante no arquivo abertos como somente-leitura enquanto aguarda por próximos comandos. Esta opção associada com bons valores de *write cache size* pode fazer alguma diferença. Note que o valor de leitura nunca ultrapassa o valor de "read size".

A opção *read size* permite obter um sincronismo fino entre a leitura e gravação do disco com o envio/recebimento de dados da rede. O valor é dependente da instalação local, levando em consideração a velocidade de disco rígido, rede, etc. O valor padrão é 16384.

Em casos onde um NFS montado ou até mesmo leitura em discos locais é compartilhada, o parâmetro *strict locking* definido para *yes* pode fazer alguma diferença de performance. Note que nem todos os sistemas ganham performance com o uso desta opção e não deve ser usada em aplicativos que não requisitam o estado do lock de arquivo ao servidor.

Caso você possua aplicativos que fazem o lock corretamente de arquivos, você poderá usar o *share modes = no*, isto significa que futuras aberturas de arquivo podem ser feitas em modo leitura/gravação. Caso utilize um aplicativo muito bem programado que implementa de forma eficiente de lock, você poderá desativar esta opção.

O uso de *oplocks yes* em compartilhamentos aumenta a performance de acesso a arquivos em até 30%, pois utiliza um código de cache no cliente. Tenha certeza do que está fazendo antes de sair usando *oplocks* em tudo que é lugar. A desativação de *kernel oplocks* é necessária para que isto funcione.

A opção *read raw* e *write raw* devem ter seus valores experimentados para ver se faz diferença na performance da sua rede, pois é diretamente dependente do tipo de cliente que sua rede possui. Alguns clientes podem ficar mais lentos em modo de leitura raw.

O tipo de sistema de arquivos adotado na máquina e suas opções de montagem tem um impacto direto na performance do servidor, principalmente com relação a atualização de status dos arquivos no sistema de arquivos (hora de acesso, data, etc).

O cache de leitura adiante de abertura de arquivos em modo somente leitura aumenta a performance com o uso do *oplocks* nível 2. Para isto, ajuste a opção *level2 oplocks* para *yes*. A recomendação deste tipo de *oplock* é o mesmo do nível 1.

Como o SAMBA faz o transporte NetBEUI via TCP/IP, ajustes no socket fazem diferença nos dados que trafegam na rede. Como isso é dependente de rede você precisará usar técnicas de leitura/gravação para

determinar quais são as melhores que se encaixam em seu caso. A opção *socket options* é usada para fazer tais ajustes, por exemplo:

```
socket options = SO_SNDBUF=2048 IPTOS_THROUGHPUT=1
```

Em especial, a opção `TCP_NODELAY` apresenta uma perceptível melhoria de performance no acesso a arquivos locais.

**OBS::** Não use espaços entre o sinal de "=" quando especificar as opções do parâmetro `socket options`.

## Configuração de Clientes NetBEUI

Este capítulo documenta a configuração de máquinas clientes NetBEUI, requerimentos de cada configuração e documenta os passos necessários para ter o cliente se comunicando perfeitamente com o seu servidor. Serão explicadas tanto a configuração de *grupo de trabalho* como de *domínio* e como a configuração é compatível entre **Linux** e **Windows**, estas explicações são perfeitamente válidas para configurar clientes que acessem servidores **Windows**.

## Considerações sobre o Windows for Workgroups e LanManager

Sistemas com implementações NetBIOS mais antigos, como o **Windows for Workgroups** (Windows 3.11) e o **Lan Manager** (DOS), enviam somente a senha para acesso ao compartilhamento, desta forma, para o acesso ser autorizado pelo **samba**, você deverá especificar a diretiva `user = usuario` para que a senha confira com o usuário local do sistema. A senha enviada também é em formato texto plano. Este problema não ocorre no Windows 95 e superiores, que enviam o nome de usuário que efetuou o logon junto com a respectiva senha.

Se a segurança do seu samba depende de senhas criptografadas, será necessário utilizar a diretiva `"include = outro_arquivo_de_configuração.%m"` para definir configurações específicas de acesso para estas máquinas.

Outro detalhe que deve ser lembrado é que o **Windows for Workgroups** envia sempre a senha em MAIÚSCULAS, então é preciso configurar o SAMBA para tentar combinações de maiúsculas/minúsculas usando o parâmetro `mangle case` e `default case` na seção global do `smb.conf`.

## Configurando clientes em Grupo de Trabalho

Para configurar o cliente para fazer parte de um *grupo de trabalho*, é necessário apenas que tenha em mãos o nome do grupo de trabalho (workgroup) que os clientes farão parte e o nome de uma outra máquina que faz parte do mesmo grupo (para testes iniciais). Com estes dados em mãos, selecione na lista abaixo o nome do cliente que deseja configurar para incluir no grupo de trabalho:

- “Windows 9X”
- “Windows XP Home Edition”
- “Windows XP Professional Edition”
- “Windows XP Server Edition”
- “Windows NT WorkStation”

- “Windows NT Server”
- “Windows 2000 Professional”
- “Windows 2000 Server”
- “Linux”

## Windows 9X

Estas configurações são válidas para clientes Windows 95, Windows 95OSR/2, Windows 98. Caso utilize o **Windows 95** (qualquer uma das séries) é aconselhável atualizar a stack TCP/IP e NetBEUI para corrigir alguns problemas que podem deixar sua máquina vulnerável na versão que acompanha o WinSock do Windows 95.

Para tornar uma máquina parte do grupo de trabalho, siga os seguintes passos:

- Entre nas propriedades de rede no Painel de Controle
- Instale o Cliente para redes Microsoft (caso não esteja instalado).
- Instale o Protocolo TCP/IP. Você também pode instalar o protocolo NetBIOS, mas utilizaremos o suporte NetBIOS sobre TCP/IP que é o usado pelo **SAMBA** além de ter um melhor desempenho, permitir integração com servidores WINS, etc.
- Clique em "Protocolo TCP/IP" e em Propriedades. Clique na tab "NetBIOS" e marque a opção "Desejo ativar o NetBIOS através do TCP/IP". Caso esta caixa esteja em cinza, então está tudo certo também.
- Clique na tab "Identificação" e coloque lá o nome que identificará o computador (até 15 caracteres) e o nome do grupo de trabalho que ele fará parte (por exemplo "workgroup", "suporte", etc) . No campo "Descrição do Computador", coloque algo que identifique a máquina na rede (por exemplo, "Computador da área de suporte").
- Clique na tab "Controle de Acesso" e marque o "Controle de acesso a nível de compartilhamento" (a não ser que tenha configurado um servidor que mantenha um controle de nível de usuário na rede para as máquinas fora do domínio).
- Clique em OK até reiniciar o computador.

A máquina cliente agora faz parte do grupo de trabalho! Tente acessar um outro computador da rede e navegar através do ambiente de rede. Caso a lista de máquinas demore em aparecer, tente acessar diretamente pelo nome do computador, usando o seguinte formato: "\\computador"

## Windows XP Home Edition

Siga as instruções de “Windows XP Professional Edition”.

## Windows XP Professional Edition

- Logue como administrador do sistemas local.
- Entre no item *Sistema* dentro do painel de controle. A tela propriedades de sistema será aberta.
- No campo *Descrição do Computador*, coloque algo que descreva a máquina (opcional).
- Clique na TAB *Nome do Computador* e no botão *Alterar* na parte de baixo da janela.

- No campo *nome do computador*, coloque um nome de no máximo 15 caracteres para identificar a máquina na rede.
- Clique em *grupo de trabalho* e digite o nome do *grupo de trabalho* na caixa de diálogo.
- Clique em OK e aguarde a mensagem confirmando sua entrada no grupo de trabalho. Será necessário reiniciar a máquina.

## Windows XP Server Edition

Siga as instruções de “Windows XP Professional Edition”.

## Windows NT WorkStation

Veja “Windows NT Server”.

## Windows NT Server

- Clique no item Rede do painel de controle.
- Na tab *Serviços*, confira se os serviços *Estação de trabalho*, *Interface de NetBIOS* e *Serviços TCP/IP simples* estão instalados. Caso não estejam, faça sua instalação usando o botão *Adicionar* nesta mesma janela.
- Na tab *Protocolos*, verifique se os protocolos *NetBEUI* e *TCP/IP* estão instalados. Caso não estejam, faça sua instalação clicando no botão *Adicionar* nesta mesma janela.
- Na tab *identificação*, clique no botão *Alterar*
- Na janela que se abrirá, coloque o nome do computador no campo *Nome do Computador*
- Clique em *Grupo de trabalho* e escreva o nome do grupo de trabalho em frente.
- Clique em OK até voltar.
- Pronto, seu computador agora faz parte do grupo de trabalho.

## Windows 2000 Professional

- Logue como administrador do sistemas local.
- Entre no item *Sistema* dentro do painel de controle. A tela propriedades de sistema será aberta. Clique em "Computador" e então no botão "Propriedades".
- No campo *nome do computador*, coloque um nome de no máximo 15 caracteres para identificar a máquina na rede.
- Clique em *grupo de trabalho* e digite o nome do *grupo de trabalho* na caixa de diálogo.
- Clique em OK e aguarde a mensagem confirmando sua entrada no grupo de trabalho. Será necessário reiniciar a máquina.

## Windows 2000 Server

- Logue como administrador do sistemas local.

- Entre no item *Sistema* dentro do painel de controle. A tela propriedades de sistema será aberta. Clique em "Descrição de rede" e então no botão "Propriedades".
- No campo *nome do computador*, coloque um nome de no máximo 15 caracteres para identificar a máquina na rede.
- Clique em *grupo de trabalho* e digite o nome do *grupo de trabalho* na caixa de diálogo.
- Clique em OK e aguarde a mensagem confirmando sua entrada no grupo de trabalho. Será necessário reiniciar a máquina.

## Linux

Os aplicativos **smbclient** e **smbmount** são usados para navegação e montagem dos discos e impressoras compartilhadas em máquinas **Linux**. Se você procura programas de navegação gráficos, como o *Ambiente de Rede* do **Windows** ou mais poderosos, veja "Programas de navegação gráficos". Como complemento, também é explicado o programa **nmblookup** para resolução de endereços NetBIOS em IP e vice-versa e a forma que as funções de máquinas são definidas em uma rede NetBEUI.

### smbmount

O **smbmount** é uma ferramenta que permite a montagem de um disco compartilhado por uma máquina NetBEUI remota como uma partição. Veja alguns exemplos:

|                                                                                          |                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>smbmount //servidor/discoc /mnt/discoc</code>                                      | Monta o compartilhamento de <code>//servidor/discoc</code> em <code>/mnt/discoc</code> usando o nome de usuário atual. Será pedido uma senha para acessar o conteúdo do compartilhamento, caso ele seja público, você pode digitar qualquer senha ou simplesmente pressionar enter. |
| <code>smbmount //servidor/discoc /mnt/discoc -N</code>                                   | Semelhante ao comando cima, com a diferença que o parâmetro <code>-N</code> não pergunta por uma senha. Isto é ideal para acessar compartilhamentos anônimos.                                                                                                                       |
| <code>smbmount //servidor/discoc /mnt/discoc -o username=gleydson,workgroup=teste</code> | Semelhante aos anteriores, mas acessa o compartilhamento usando <code>gleydson</code> como nome de usuário e <code>teste</code> como grupo de trabalho. Este método é ideal para redes que tem o nível de acesso por usuário ou para acessar recursos compartilhados em um domínio. |

### smbclient

O **smbclient** é uma ferramenta de navegação em servidores SAMBA. Ao invés dela montar o compartilhamento como um disco local, você poderá navegar na estrutura do servidor de forma semelhante a um cliente FTP e executar comandos como `ls`, `get`, `put` para fazer a transferência de arquivos entre a máquina remota e a máquina local. Também é através dele que é feita a interface com impressoras compartilhadas remotamente. Veja exemplos do uso do **smbclient**:

|                                                       |                                                                                                              |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>smbclient -L samba1</code>                      | Lista todos os compartilhamentos existentes ( <code>-L</code> ) no servidor <code>samba1</code> .            |
| <code>smbclient //samba1/discoc</code>                | Acessa o conteúdo do compartilhamento <code>discoc</code> no servidor <code>samba1</code> .                  |
| <code>smbclient //samba1/discoc -N</code>             | Idêntico ao acima, mas não utiliza senha (ideal para compartilhamentos com acesso anônimo).                  |
| <code>smbclient //samba1/discoc -I 192.168.1.2</code> | Se conecta ao compartilhamento usando o endereço IP <code>192.168.1.2</code> ao invés da resolução de nomes. |

|                                                                    |                                                                                                                   |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>smbclient //samba1/discoc -U gleydson -W teste</code>        | Se conecta ao compartilhamento como usuário <code>gleydson</code> usando o grupo de trabalho <code>teste</code> . |
| <code>smbclient //samba1/discoc -U gleydson%teste1 -W teste</code> | Idêntico ao acima, mas também envia a senha <code>teste1</code> para fazer a conexão diretamente.                 |

Caso receba a mensagem `NT Status Access Denied`, isto quer dizer que não possui direitos de acesso adequados para listas ou acessar os compartilhamentos da máquina. Nesse caso, utilize as opções `-U usuário` e `-W grupo/domínio` para fazer acesso com uma conta válida de usuário existente na máquina.

**OBS:**Note que a ordem das opções faz diferença no **smbmount**.

## nmblookup

Esta é uma ferramenta usada para procurar nomes de cliente usando o endereço IP, procurar um IP usando o nome e listar as características de cada cliente. Veja alguns exemplos:

|                                     |                                                                      |
|-------------------------------------|----------------------------------------------------------------------|
| <code>nmblookup -A 127.0.0.1</code> | Lista o nome e as opções usadas pelo servidor <code>127.0.0.1</code> |
| <code>nmblookup servidor</code>     | Resolve o endereço IP da máquina <code>servidor</code> .             |

A listagem exibida pela procura de IP do **nmblookup** possui códigos hexadecimais e cada um deles possui um significado especial no protocolo NetBEUI. Segue a explicação de cada um:

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identificação da máquina        | <ul style="list-style-type: none"> <li>• <code>COMPUTADOR&lt;00&gt;=</code> O serviço NetBEUI está sendo executado na máquina.</li> <li>• <code>COMPUTADOR&lt;03&gt;</code> = Nome genérico da máquina (nome NetBIOS).</li> <li>• <code>COMPUTADOR&lt;20&gt;</code> = Serviço LanManager está sendo executado na máquina.</li> </ul>                                                                                                                                                                                                                                                                                                                             |
| Identificação de grupos/domínio | <ul style="list-style-type: none"> <li>• <code>GRUPO_TRABALHO&lt;1d&gt;</code> - <code>&lt;GRUPO&gt;</code> = Navegador Local de Domínio/Grupo.</li> <li>• <code>GRUPO_TRABALHO&lt;1b&gt;</code> = Navegador Principal de Domínio.</li> <li>• <code>GRUPO_TRABALHO&lt;03&gt;</code> - <code>&lt;GRUPO&gt;</code> = Nome Genérico registrado por todos os membros do grupo de trabalho.</li> <li>• <code>GRUPO_TRABALHO&lt;1c&gt;</code> - <code>&lt;GRUPO&gt;</code> = Controladores de Domínio / Servidores de logon na rede.</li> <li>• <code>GRUPO_TRABALHO&lt;1e&gt;</code> - <code>&lt;GRUPO&gt;</code> = Resolvedores de Nomes Internet (WINS).</li> </ul> |

Estes códigos podem lhe ser úteis para localizar problemas mais complicados que possam ocorrer durante a configuração de um servidor.

## Configurando clientes em Domínio

Para configurar qualquer um dos cliente abaixo para fazer parte de um domínio de rede, é necessário apenas que tenha em mãos os seguintes dados:

- Nome do controlador de domínio PDC
- Nome do domínio
- Nome de usuário e senha que foram cadastrados no servidor.
- Acesso administrador no SERVIDOR PDC (SAMBA, NT, etc).
- Cria uma conta de máquina no domínio (no caso da máquina ser um Windows NT, Windows XP, Windows 2k ou Linux). Veja “Contas de máquinas de domínio” para maiores detalhes.

Como o Windows 3.11, Windows 95, Windows 98, Windows ME não possuem uma conta de máquina, eles nunca serão um membro real de um domínio, podendo sofrer um name spoofing e terem a identidade roubada. Mesmo assim, eles terão pleno acesso aos recursos do domínio e uma configuração mais fácil que os demais clientes. Com estes dados em mãos, selecione na lista abaixo o nome do cliente que deseja integrar no grupo de trabalho:

- “Windows 9X”
- “Windows XP Home Edition”
- “Windows XP Professional Edition”
- “Windows XP Server Edition”
- “Windows NT WorkStation”
- “Windows NT Server”
- “Windows 2000 Professional”
- “Windows 2000 Server”
- “Linux”

**OBS:** O Windows 2000 apresenta algumas dificuldades em entrar na rede do SAMBA 2.2, sendo necessário o uso do SAMBA TNG 2.2.x para aceitar o logon de estações Windows 2000.

## Windows 9X

Estas configurações são válidas para clientes Windows 95, Windows 95OSR/2, Windows 98. Caso utilize o **Windows 95** (qualquer uma das séries) é aconselhável atualizar a stack TCP/IP e NetBEUI para corrigir alguns problemas que podem deixar sua máquina vulnerável na versão que acompanha o WinSock do Windows 95.

Para tornar uma máquina parte do domínio, siga os seguintes passos:

- Entre nas propriedades de rede no Painel de Controle
- Instale o Cliente para redes Microsoft (caso não esteja instalado).
- Instale o Protocolo TCP/IP. Você também pode instalar o protocolo NetBIOS, mas utilizaremos o suporte NetBIOS sobre TCP/IP que é o usado pelo **SAMBA** além de ter um melhor desempenho, permitir integração com servidores WINS, etc.
- Clique em "Cliente para redes Microsoft", marque a opção "Efetuar logon no domínio do Windows NT". Coloque o nome do domínio que irá configurar o cliente para fazer parte na caixa "Domínio do

Windows NT" (por exemplo, "suporte"). Na parte de baixo da caixa de diálogo, você poderá escolher como será o método para restaurar as conexões de rede. Inicialmente, recomendo que utilize a "Efetuar logon e restaurar as conexões de rede" que é mais útil para depurar problemas (possíveis erros serão mostrados logo que fizer o logon no domínio).

Adeque esta configuração as suas necessidades quando estiver funcionando :)

- Clique em "Protocolo TCP/IP" e em Propriedades. Clique na tab "NetBIOS" e marque a opção "Desejo ativar o NetBIOS através do TCP/IP". Caso esta caixa esteja em cinza, então está tudo certo também.
- Clique na tab "Identificação" e coloque lá o nome que identificará o computador (até 15 caracteres).
- Digite o nome de um grupo de trabalho que a máquina fará parte no campo "Grupo de Trabalho" (por exemplo "workgroup", "suporte", etc). Este campo somente será usado caso o logon no domínio NT não seja feito com sucesso. No campo "Descrição do Computador", coloque algo que identifique a máquina na rede (por exemplo, "Computador da área de suporte").
- Clique na tab "Controle de Acesso" e marque o "Controle de acesso a nível de usuário e especifique o nome da máquina que serve a lista de usuários, que normalmente é a mesma do PDC.
- Clique em OK até reiniciar o computador.

Quando for mostrada a tela pedindo o nome/senha, preencha com os dados da conta de usuário que criou no servidor. No campo domínio, coloque o domínio que esta conta de usuário pertence e tecle <Enter>. Você verá o script de logon em ação (caso esteja configurado) e a máquina cliente agora faz parte do domínio! Tente acessar um outro computador da rede e navegar através do ambiente de rede. Caso a lista de máquinas demore em aparecer, tente acessar diretamente pelo nome do computador, usando o seguinte formato: "\\computador"

## Windows XP Home Edition

Não é possível fazer o **Windows XP Home Edition** ser parte de um domínio, por causa de limitações desta versão.

## Windows XP Professional Edition

- Primeiro, siga todos os passos para ingressar a máquina em um grupo de trabalho como documentado em "Windows XP Professional Edition".
- Atualize o registro para permitir a entrada no domínio:

1. Copie o seguinte conteúdo para o arquivo WinXP-Dom.reg:

```
REGEDIT4
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\netlogon\parameters
"RequireSignOrSeal"=dword:00000000
"SignSecureChannel"=dword:00000000
```

2. Execute o comando `regedit WinXP-Dom.reg` no cliente XP.

- Entre nos itens (em seqüência) Painel de controle/Ferramentas Administrativas/ Política de segurança local/políticas locais e depois em "opções de segurança". Na janela de opções de segurança, desative as opções "Encriptar digitalmente ou assinar um canal seguro (sempre)", "Desativar modificações de senha na conta de máquina" e "Requer chave de seção forte (Windows 2000 ou superior)."

- Reinicie a máquina.
- Após reiniciar a máquina, volte na tela de alteração de identificação de máquina na rede.
- Clique com o mouse em "Domínio" e digite o nome do domínio na caixa de diálogo.
- Na tela seguinte, será lhe pedido o nome de usuário e senha com poderes administrativos que podem inserir/remover máquinas do domínio.
- Clique em OK e aguarde a mensagem confirmando sua entrada no domínio. Será necessário reiniciar a máquina após concluir este passo.

## Windows XP Server Edition

Siga os procedimentos documentados em “Windows XP Professional Edition”

## Windows NT WorkStation

Veja os passos em “Windows NT Server”.

## Windows NT Server

- Clique no item Rede do painel de controle.
- Na tab *Serviços*, confira se os serviços *Estação de trabalho*, *Interface de NetBIOS* e *Serviços TCP/IP simples* estão instalados. Caso não estejam, faça sua instalação usando o botão *Adicionar* nesta mesma janela.
- Na tab *Protocolos*, verifique se os protocolos *NetBEUI* e *TCP/IP* estão instalados. Caso não estejam, faça sua instalação clicando no botão *Adicionar* nesta mesma janela.
- Na tab *identificação*, clique no botão *Alterar*
- Na janela que se abrirá, coloque o nome do computador no campo *Nome do Computador*
- Clique em *Domínio* e escreva o nome do domínio que deseja entrar.
- Para criar uma *conta de máquina* no domínio, clique em *criar uma conta de computador no domínio* e coloque na parte de baixo o nome do usuário sua senha. O usuário deverá ter poderes para adicionar máquinas no domínio. Caso a conta de máquina não seja criada, o Windows NT será como um Windows 95/98 na rede, sem a segurança que seu nome NetBIOS não seja usado por outros (veja “Contas de máquinas de domínio”).
- Clique em OK até voltar.
- Pronto, seu computador agora faz parte do domínio.

## Windows 2000 Professional

Siga os passos descritos em “Windows 2000 Server”.

## Windows 2000 Server

- Primeiro, siga todos os passos para ingressar a máquina em um grupo de trabalho como documentado em “Windows 2000 Server”.
- Após reiniciar a máquina, volte na tela de alteração de identificação de máquina na rede.

- Clique com o mouse em "Domínio" e digite o nome do domínio na caixa de diálogo.
- Na tela seguinte, será lhe pedido o nome de usuário e senha com poderes administrativos que podem inserir/remover máquinas do domínio.
- Clique em OK e aguarde a mensagem confirmando sua entrada no domínio. Será necessário reiniciar a máquina após concluir este passo.

Caso não consiga trocar a senha do Windows 2000 no servidor PDC, desative a opção `unix password sync`.

## Linux

- Entre no sistema como usuário `root`.
- Instale o **SAMBA** caso não esteja ainda instalado.
- Edite o arquivo de configuração do samba `/etc/samba/smb.conf`, será necessário modificar as seguintes linhas na seção `[global]`:

```
[global]
workgroup = nome_domínio
security = domain
password server = nome_pdc nome_bdc
encrypt passwords = true
```

Onde:

- `workgroup` - Nome do domínio que deseja fazer parte.
- `security` - Nível de segurança. Nesta configuração, utilize "domain".
- `password server` - Nome da máquina PDC, BDC. Também poderá ser usado \*, assim o **SAMBA** tentará descobrir o servidor PDC e BDC automaticamente, da mesma forma usada pelo **Windows**.
- `encrypt passwords` - Diz se as senhas serão encriptadas ou não. Sempre utilize senhas criptografadas para colocar uma máquina em um domínio.

Reinicie o servidor **SAMBA** após estas modificações.

- Execute o comando: `smbpasswd -j domínio -r PDC/BDC -U usuario_admin`. Onde:
  - `domínio` - Domínio que deseja fazer o logon
  - `PDC/BDC` - Nome da máquina PDC/BDC do domínio. Em alguns casos, pode ser omitido.
  - `usuario_admin` - Usuário com poderes administrativos para ingressara a máquina no domínio.
- Se tudo der certo, após executar este comando, você verá a mensagem:

```
Joined domain "domínio".
```

Se sua configuração não funcionou, revise com atenção todos os itens acima. Verifique se a conta de máquina foi criada no servidor e se o **SAMBA** na máquina cliente foi reiniciado. De também uma olhada em "Erros conhecidos durante o logon do cliente".

**OBS:**O SAMBA envia primeiramente um usuário/senha falso para verificar se o servidor rejeita o acesso antes de enviar o par de nome/senha corretos. Por este motivo, seu usuário pode ser bloqueado após um determinado número de tentativas em alguns servidores mais restritivos. Para acessar os recursos compartilhados, veja “Linux”. Note que não é obrigatório realizar as configurações acima para acessar os recursos de uma máquina em domínio, basta apenas que autentique com seu nome de usuário/senha no domínio e que ela seja autorizada pelo PDC.

## Erros conhecidos durante o logon do cliente

Esta seção contém os erros mais comuns e a forma de correção da maioria dos problemas que ocorrem quando um cliente SAMBA tenta entrar em domínio.

- `error creating domain user: NT_STATUS_ACCESS_DENIED` - A conta de máquina no domínio não foi criada. Veja “Contas de máquinas de domínio” para mais detalhes.
- `NT_STATUS_NO_TRUST_SAM_ACCOUNT` - Não existe conta de máquina no **Windows NT** para autenticar uma máquina no domínio. Esta mensagem é mostrada quando a máquina SAMBA é cliente de um domínio NT.
- `error setting trust account password: NT_STATUS_ACCESS_DENIED` - A senha para criação de conta na máquina está incorreta ou a conta utilizada não tem permissões para ingressar uma máquina no domínio (veja “Criando uma conta de administrador de domínio”). Caso esteja usando um cliente SAMBA, verifique se o parâmetro `encrypt passwords` está ativado.
- A senha informada não está correta ou o acesso ao seu servidor de logon foi negado - Verifique primeiro os logs de acessos do sistema. Caso o SAMBA esteja sendo executado via **inetd**, verifique se a configuração padrão é restritiva e se o acesso está sendo negado pelos arquivos do `tcp wrappers` `hosts.allow` e `hosts.deny`.
- não existem servidores de logon no domínio - Verifique se o parâmetro `domain logons = yes` foi usado para permitir o logon em domínio.

## Programas de navegação gráficos

O **smbclient**, **nmblookup** e **smbmount** são ferramentas extremamente poderosas auxiliando bastante o administrador na tarefa de configuração de sua rede e resolver problemas. Para o uso no dia a dia ou quando não é necessária a operação via console, você pode utilizar uma das alternativas abaixo que são front-ends a estas ferramentas e facilitam o trabalho de navegação na rede.

### linneighborhood

Cliente SAMBA baseado em GTK, muito leve e possibilita a navegação entre os grupos máquinas em forma de árvore. Ele também permite a montagem de compartilhamentos remotos. Caso precise de recursos mais complexos e autenticação, recomendo o “TkSmb”.

### TkSmb

Cliente SAMBA baseado em TCL/TK. Seu ponto forte é a navegação nos recursos da máquina ao invés da rede completa, possibilitando autenticação em domínio/grupo de trabalho, montagem de recursos, etc.

## Cliente de configuração gráficos

São ferramentas que permitem a configuração do samba usando a interface gráfica. Isto facilita bastante o processo, principalmente se estiver em dúvidas em algumas configurações, mas como todo bom

administrador UNIX sabe, isto não substitui o conhecimento sobre o funcionamento de cada opção e ajustes e organização feita diretamente no arquivo de configuração.

## gnosamba

Ferramenta de configuração gráfica usando o GNOME. Com ele é possível definir configurações localmente. Ele ocupa pouco espaço em disco, e se você gosta de GTK, este é o recomendado.

As opções do SAMBA são divididas em categorias facilitando sua localização e uso.

## swat

Ferramenta de administração via web do samba. Este é um daemon que opera na porta 901 da máquina onde o servidor samba foi instalado. A configuração é feita através de qualquer navegador acessando `http://ip_do_servidor:901` e logando-se como usuário root (o único com poderes para escrever no arquivo de configuração).

Esta ferramenta vem evoluindo bastante ao decorrer dos meses e é a recomendada para a configuração do servidor SAMBA remotamente. Seu modo de operação divide-se em *básico* e *avançado*. No modo básico, você terá disponível as opções mais comuns e necessárias para compartilhar recursos na rede. O modo avançado apresenta praticamente todos os parâmetros aceitos pelo servidor samba (restrições, controle de acesso, otimizações, etc.).

# Exemplos de configuração do servidor SAMBA

Os exemplos existentes nesta seção cobrem diferentes tipos de configuração do servidor, tanto em modo de compartilhamento com acesso público ou um domínio restrito de rede. Todos os exemplos estão bem comentados e explicativos, apenas pegue o que se enquadre mais em sua situação para uso próprio e adaptações.

## Grupo de Trabalho com acesso público

Este exemplo pode ser usado de modelo para construir uma configuração baseada no controle de acesso usando o nível de segurança *share* e quando possui compartilhamentos de acesso público. Esta configuração é indicada quando necessita de compatibilidade com softwares NetBIOS antigos.

```
Arquivo de configuração do SAMBA criado por
Gleydson Mazioli da Silva <gleydson@debian.org>
para o guia Foca GNU/Linux Avançado - Capítulo SAMBA
Este script pode ser copiado e distribuído livremente de
acordo com os termos da GPL. Ele não tem a intenção de
atender uma determinada finalidade, sendo usado apenas
para fins didáticos, portanto fica a inteira responsabilidade
do usuário sua utilização.

[global]
nome da máquina na rede
netbios name = teste
nome do grupo de trabalho que a máquina pertencerá
workgroup = focalinux
nível de segurança share permite que clientes antigos mantenham a compatibilidade
enviando somente a senha para acesso ao recurso, determinando o nome de usuário
de outras formas
```

```

security = share
O recurso de senhas criptografadas não funciona quando usamos o nível share
de segurança. O motivo disto é porque automaticamente é assumido que você
está selecionando este nível por manter compatibilidade com sistemas antigos
ou para disponibilizar compartilhamentos públicos, onde
encrypt passwords = false
Conta que será mapeada para o usuário guest
guest account = nobody
Como todos os compartilhamentos desta configuração são de acesso público
coloquei este parâmetro na seção [global], assim esta opção afetará todos
os compartilhamentos.
guest ok = 1
Conjunto de caracteres utilizados para acessar os compartilhamentos. O padrão
para o Brasil e países de língua latina é o ISO 8859-1
character set = ISO8859-1

Compartilha o diretório /tmp (path = /tmp) com o nome "temporario" ([temporario]
é adicionada a descrição "Diretório temporário" com acesso leitura/gravação
(read only = no) e exibido na janela de navegação da rede (browseable = yes).
[temporario]
path = /tmp
comment = Diretório temporário
read only = no
browseable = yes

Compartilha o diretório /pub (path = /pub) com o nome "publico" ([publico]).
A descrição "Diretório de acesso público" é associada ao compartilhamento
com acesso somente leitura (read only = yes) e exibido na janela de navegação
da rede (browseable = yes).
[publico]
path = /pub
comment = Diretório de acesso público
read only = yes
browseable = yes

Compartilha todas as impressoras encontradas no /etc/printcap do sistema
Uma descrição melhor do tipo especial de compartilhamento "[printers]"
é explicado no início do guia Foca Linux
[printers]
comment = All Printers
path = /tmp
create mask = 0700
printable = Yes
browseable = No

```

## Grupo de Trabalho com acesso por usuário

O exemplo abaixo descreve uma configuração a nível de segurança por usuário onde existem compartilhamentos que requerem login e usuários específicos, e restrições de IPs e interface onde o servidor opera. Esta configuração utiliza senhas em texto claro para acesso dos usuários, mas pode ser facilmente modificada para suportar senhas criptografadas.

```

Arquivo de configuração do SAMBA criado por
Gleydson Mazioli da Silva >gleydson@debian.org>
para o guia Foca GNU/Linux Avançado - Capítulo SAMBA
Este script pode ser copiado e distribuído livremente de
acordo com os termos da GPL. Ele não tem a intenção de
atender uma determinada finalidade, sendo usado apenas
para fins didáticos, portanto fica a inteira responsabilidade
do usuário sua utilização.

[global]
nome da máquina na rede
netbios name = teste
nome do grupo de trabalho que a máquina pertencerá
workgroup = focalinux
nível de segurança user somente aceita usuários autenticados após o envio
de login/senha
security = user
É utilizada senhas em texto claro nesta configuração
encrypt passwords = false
Conta que será mapeada para o usuário guest
guest account = nobody
Permite restringir quais interfaces o SAMBA responderá
bind interfaces only = yes
Faz o samba só responder requisições vindo de eth0
interfaces = eth0
Supondo que nossa interface eth0 receba conexões roteadas de diversas
outras redes, permite somente as conexões vindas da rede 192.168.1.0/24
hosts allow = 192.168.1.0/24
A máquina 192.168.1.57 possui gateway para acesso interno, como medida
de segurança, bloqueamos o acesso desta máquina.
hosts deny = 192.168.1.57/32

Conjunto de caracteres utilizados para acessar os compartilhamentos. O padrão
para o Brasil e países de língua latina é o ISO 8859-1
character set = ISO8859-1

As restrições do PAM terão efeito sobre os usuários e recursos usados do SAMBA
obey pam restriction = yes

Mapeia o diretório home do usuário autenticado. Este compartilhamento especial
é descrito em mais detalhes no início do capítulo sobre o SAMBA no Foca Linux.
[homes]
 comment = Diretório do Usuário
 create mask = 0700
 directory mask = 0700
 browseable = No

Compartilha o diretório win (path = /win) com o nome "win" ([win]).
A descrição associada ao compartilhamento será "Disco do Windows",
o nome de volume precisa ser especificado pois usamos programas
que a proteção anti cópia é o serial. Ainda fazemos uma proteção
onde qualquer usuário existente no grupo @adm é automaticamente
rejeitado e o usuário "baduser" somente possui permissão de leitura
(read list = baduser).

```

```
#
[win]
path = /win
comment = Disco do Windows
volume = 3CF434C
invalid users = @adm
browseable = yes
read list = baduser

Compartilha o diretório /pub (path = /pub) com o nome "publico" ([publico]).
A descrição "Diretório de acesso público" é associada ao compartilhamento
com acesso somente leitura (read only = yes) e exibido na janela de navegação
da rede (browseable = yes). O parâmetro public = yes permite que este
compartilhamento seja acessado usando o usuário "nobody" sem o fornecimento
de senha.
[publico]
path = /pub
comment = Diretório de acesso público
read only = yes
browseable = yes
public = yes
```

## Domínio

```
Arquivo de configuração do SAMBA criado por
Gleydson Mazioli da Silva <gleydson@debian.org>
para o guia Foca GNU/Linux Avançado - Capítulo SAMBA
Este script pode ser copiado e distribuído livremente de
acordo com os termos da GPL. Ele não tem a intenção de
atender uma determinada finalidade, sendo usado apenas
para fins didáticos, portanto fica a inteira responsabilidade
do usuário sua utilização.

[global]
nome da máquina na rede
netbios name = teste
nome do grupo de trabalho que a máquina pertencerá
workgroup = focalinux
String que será mostrada junto com a descrição do servidor
server string = servidor PDC principal de testes
nível de segurança user somente aceita usuários autenticados após o envio
de login/senha
security = user
Utilizamos senhas criptografadas nesta configuração
encrypt passwords = true
smb passwd file = /etc/samba/smbpasswd
Conta que será mapeada para o usuário guest
guest account = nobody
Permite restringir quais interfaces o SAMBA responderá
bind interfaces only = yes
Faz o samba só responder requisições vindo de eth0
interfaces = eth0
```

```

como estamos planejando ter um grande número de usuários na rede, dividimos
os arquivos de log do servidor por máquina.
log file = /var/log/samba/samba-%m-%I.log
O tamanho de CADA arquivo de log criado deverá ser 1MB (1024Kb).
max log size = 1000
Escolhemos um nível de OS com uma boa folga para vencer as eleições de
controlador de domínio local
os level = 80
Dizemos que queremos ser o Domain Master Browse (o padrão é auto)
domain master = yes
Damos algumas vantagens para o servidor ganhar a eleição caso
aconteça desempate por critérios
preferred master = yes
Também queremos ser o local master browser para nosso segmento de rede
local master = yes
Este servidor suportará logon de usuários
domain logons = yes
Usuários que possuem poderes para adicionar/remover máquinas no domínio
(terão seu nível de acesso igual a root)
admin users = gleydson
Unidade que será mapeada para o usuário local durante o logon (apenas
sistemas baseados no NT).
logon drive = m:
Nome do script que será executado pelas máquinas clientes
logon script = logon.bat

Ação que será tomada durante o recebimento de mensagens do
Winpopup.
message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &

Conjunto de caracteres utilizados para acessar os compartilhamentos. O padrão
para o Brasil e países de língua latina é o ISO 8859-1
character set = ISO8859-1

As restrições do PAM terão efeito sobre os usuários e recursos usados do SAMBA
obey pam restriction = yes

Mapeia o diretório home do usuário autenticado. Este compartilhamento especial
é descrito em mais detalhes no início do capítulo sobre o SAMBA no Foca Linux.
[homes]
 comment = Diretório do Usuário
 create mask = 0700
 directory mask = 0700
 browseable = No

Compartilha o diretório win (path = /win) com o nome "win" ([win]).
A descrição associada ao compartilhamento será "Disco do Windows",
o nome de volume precisa ser especificado pois usamos programas
que a proteção anti cópia é o serial. Ainda fazemos uma proteção
onde qualquer usuário existente no grupo @adm é automaticamente
rejeitado e o usuário "baduser" somente possui permissão de leitura
(read list = baduser).
#

```

```
[win]
path = /win
comment = Disco do Windows
volume = 3CF434C
invalid users = @adm
browseable = yes
read list = baduser

Compartilha o diretório /pub (path = /pub) com o nome "publico" ([publico]).
A descrição "Diretório de acesso público" é associada ao compartilhamento
com acesso somente leitura (read only = yes) e exibido na janela de navegação
da rede (browseable = yes). O parâmetro public = yes permite que este
compartilhamento seja acessado usando o usuário "nobody" sem o fornecimento
de senha.
[publico]
path = /pub
comment = Diretório de acesso público
read only = yes
browseable = yes
public = yes

Compartilhamento especial utilizado para o logon de máquinas na rede
[netlogon]
path = /pub/samba/netlogon/logon.bat
read only = yes
```

---

# Capítulo 19. Restrições de acesso, recursos e serviços

Este capítulo documenta diversos métodos de fazer restrições de contas, limitação de acesso interno/externo, de recursos por usuários/grupos, login, tempo máximo ocioso, e outros modos para limitar o uso de recursos do sistema. Também são descritos métodos para aumentar a segurança do acesso físico a seu servidor e maneiras de restringir o uso de serviços disponíveis no sistema.

Se você deseja restringir o acesso de máquinas na rede ou portas específicas em sua máquina, veja também Capítulo 10, *Firewall iptables*.

## Limitando recursos no bash

### Uso do comando `readonly` para exportar variáveis

Variáveis exportadas na forma comum podem ser modificadas a qualquer momento pelo usuário, e isso pode trazer problemas de acordo com o tipo de sistema que administramos. A definição da variável como somente leitura (`readonly`) evita a maioria destes problemas:

```
readonly TESTE="123"
```

A variável `TESTE` não poderá ser modificada ou excluída. Com isto o administrador pode "bloquear" a modificação de variáveis que controlam o funcionamento de determinados recursos do interpretador de comandos (alguns deles serão vistos ainda nesta seção).

**OBS1:** Algumas variáveis de controle de ambiente do interpretador de comandos já são iniciadas com valores somente leitura (como as variáveis `EUID` e `PPID`)

**OBS2:** Variáveis exportadas como somente leitura em shell scripts são mantidas até a finalização do script e depois liberadas.

## Restrições nos diretórios de usuários e root

O controle de acesso a diretórios de usuários é importante quando desejamos que outras pessoas não tenham acesso ao diretório de outros usuários, violando a privacidade do mesmo e obtendo acesso a partes indesejáveis, principalmente do usuário `root`. É recomendado restringir o acesso somente ao dono e grupo do usuário, bloqueando o acesso a outros tipos de usuários:

```
chmod 2750 /root
chmod 2750 /home/usuario
```

O exemplo acima permitirá o acesso do diretório `/root` e `/home/usuario` somente ao usuário e grupo que pertencem. Este processo pode ser facilitado na criação dos diretórios de usuários em `/home` especificando a variável: `DIR_MODE=0750` no arquivo `/etc/adduser.conf`.

**OBS:** Algumas distribuições de **Linux** garantem o acesso livre a diretórios de usuários por padrão pois alguns daemons que requerem acesso a diretório de usuários rodam sob outros usuários ao invés do `root`. Um bom exemplo é a utilização do recurso "UserDir" do **Apache** para servir requisições como `http://servidor.org/~usuario`.

A restrição de diretório home neste caso bloqueará o acesso do servidor web **Apache** ao diretório `/home/usuario/public_html`. Mesmo assim, uma alternativa para garantir a utilização da restrição é incluir o usuário do servidor web **Apache** (`www-data`) no grupo "usuario" (que possui acesso ao diretório `/home/usuario`):

```
adduser www-data usuario
```

Isto garantirá que o servidor **Apache** continue servindo as requisições dentro do diretório `/home/usuario`, com acesso garantido via grupo. O mesmo princípio pode ser aplicado em outros programas, apenas leve em consideração que se um cracker tomar conta do processo que tem acesso ao seu diretório home restrito, ele certamente também terá acesso.

## Restrições básicas do shell bash com `bash -r/--restricted`, `rbash`

Quando o **bash** é iniciado com o parâmetro `-r`, `--restricted` ou como **rbash**, o shell restringe o uso dos seguintes recursos em sua seção:

- Usar o comando **cd** para mudar de diretório.
- Definindo, modificar ou apagar a variáveis `SHELL`, `PATH`, `ENV`, `BASH_ENV`.
- Nomes de comandos que contém `/`
- Especificar um nome de arquivo contendo uma `/` como argumento para o comando **builtin** (embutido no interpretador de comandos).
- Especificar uma `/` como argumento a opção `-p` no comando **hash** (embutido no interpretador de comandos).
- Importar a definição de funções do ambiente do shell atual.
- Analisar o valor da variável `SHELLOPTS` do ambiente do shell atual.
- Redirecionando a saída padrão usando os operadores de redirecionamento `>`, `> |`, `<>`, `>&`, `>&&`; e `>>`.
- Usando o comando embutido **exec** para substituir o shell por outro comando.
- Usar as opções `-f` ou `-d` com o comando **enable** (embutido no interpretador de comandos).
- Especificar a opção `-p` ao comando interno **command**.
- Desativar o modo restrito com `set +r` ou `set +o restricted *`

Estas restrições são ativadas após a leitura dos arquivos de inicialização do interpretador de comandos. O shell restrito desliga as restrições quando um shell script é executado.

## Finalizando consoles inativos

A variável `TMOUT` determina o tempo de inatividade de um shell para que ele seja terminado.

```
export TMOUT=600
```

Terminará o **bash** caso nenhum comando seja executado no período de 600 segundos (5 minutos). Veja “Uso do comando `readonly` para exportar variáveis” como complemento.

## Desabilitando o registro de comandos digitados

Todos os comandos que digitamos em uma seção do shell são registrados no arquivo `~/.bash_history`, as seguintes variáveis fazem seu controle:

- *HISTFILE* - Nome do arquivo que armazenará o histórico de comandos. O padrão é `~/.bash_history`. Caso não seja especificado, os comandos não serão gravados após finalizar o shell.
- *HISTSIZE* - Define o número de comandos que o arquivo de histórico poderá armazenar, o padrão é 500.
- *HISTFILESIZE* - Define o número máximo de linhas no arquivo de histórico.

Se você possui muitos usuários em seu sistema, é recomendado ajustar estas variáveis como somente leitura para que o usuário não desative o logging por qualquer motivo (veja “Uso do comando `readonly` para exportar variáveis”).

## Desabilitando serviços de shell para usuários

Existem casos onde o usuário precisa estar cadastrado no sistema mas não precisa ter acesso a uma conta de login válida (como um sistema servidor de e-mail ou outros serviços). Neste caso a desabilitação dos serviços de shell aumentará um pouco a segurança do sistema, mesmo conseguindo acesso a conta/senha estará impedido de entrar no sistema (pelo menos terá um pouco mais dificuldade para conseguir isso).

Um programa que é muito usado para desabilitar o shell exibindo uma mensagem ao usuário que fez a tentativa é o **falselogin**. Ele deve ser colocado como o "shell padrão" no arquivo `/etc/passwd` e exibirá a mensagem contida no arquivo `/etc/falselogin.conf` quando o login para aquele usuário for tentado. Esta operação pode ser facilitada usando a variável `DSHELL=/usr/bin/falselogin` no arquivo `/etc/adduser.conf`.

Uma forma alternativa de desativar o serviço de login de **TODOS** os usuários (exceto o `root` e os já logados no sistema) é criar um arquivo chamado `/etc/nologin` e colocando uma mensagem dentro dele, que será exibida quando tentarem efetuar o login no sistema.

**OBS:** Tome cuidado ao usar esta alternativa, este método deve ser usado somente em caso de **EMERGÊNCIA**, as distribuições **Linux** usam este método para bloquear o login de outros usuários durante o processo de inicialização, removendo assim que o processo é terminado. Esteja consciente disso.

Em alguns casos, o uso do PAM pra desabilitar os serviços de login pode ser mais adequado (veja “Restringindo/Bloqueando o login”).

## Limitação de recursos usando PAM

Pluggable Authentication Modules (Módulos de autenticação plugáveis) são um conjunto de bibliotecas usadas para fazer autenticação, gerenciamento de contas, controle de recursos dos usuários no sistema, em adição ao tradicional sistema de acesso baseado em usuários/grupos. Este recurso permite modificar a forma que um aplicativo autentica e define recursos para o usuário sem necessidade de recompilar o aplicativo principal. Os recursos que desejamos controlar restrições via PAM são especificados individualmente por serviços nos arquivos correspondentes em `/etc/pam.d` e então os arquivos correspondentes em `/etc/security` são usados para controlar tais restrições.

Nesta seção assumirei explicações dirigidas aos recursos controlados pelos arquivos em `/etc/security`. A maioria das explicações são baseadas em testes e nos próprios exemplos dos arquivos de configuração do PAM.

## Descobrir se um determinado programa tem suporte a PAM

Um método simples de se determinar se um programa binário possui suporte a PAM é executando o comando:

```
ldd [programa]
```

Por exemplo:

```
ldd /bin/login

libc.so.1 => /lib/libcrypt.so.1 (0x4001c000)
libpam.so.0 => /lib/libpam.so.0 (0x40049000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x40051000)
libdl.so.2 => /lib/libdl.so.2 (0x40054000)
libc.so.6 => /lib/libc.so.6 (0x40058000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Caso a biblioteca `libpam` for listada, o programa tem suporte a PAM compilado. Programas que não possuem suporte a PAM deverão ter o código fonte modificado inserindo as funções para tratamento dos módulos de autenticação.

## Definindo uma política padrão restritiva

A política padrão do PAM é especificado em `/etc/pam.d/other` e define o que acontecerá caso nenhum dos arquivos de controle de serviço em `/etc/pam.d` confirmem com o serviço em questão. Normalmente o módulo `pam_unix.so` é usado para fazer a política padrão, para deixar o sistema mais seguro, utilize a seguinte configuração no arquivo `/etc/pam.d/other`:

```
auth required /usr/lib/security/pam_warn.so
auth required /usr/lib/security/pam_deny.so
account required /usr/lib/security/pam_deny.so
password required /usr/lib/security/pam_warn.so
password required /usr/lib/security/pam_deny.so
session required /usr/lib/security/pam_deny.so
```

O módulo `pam_deny.so` é responsável por fazer o bloqueio, e o `pam_warn` envia avisos ao **syslog** (facilidade *auth* nível *notice*) caso serviços módulos PAM que necessitem do serviço de autenticação sejam bloqueados (isto não é feito automaticamente pelo `pam_deny.so`).

**OBS:** Esta configuração poderá causar bloqueio em muitas coisas caso possua módulos de autenticação mau configurados. Esteja certo de utilizar o módulo `pam_warn.so` (antes do `pam_deny.so`) nas diretivas restritivas para entender qual é o problema através da análise dos arquivos de logs.

Mais detalhes sobre a configuração de módulos de autenticação poderão ser encontrados no endereço <ftp://ftp.us.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html> e <http://www.kernel.org/pub/linux/libs/pam/pre/doc/rfc86.0.txt.gz>.

## Restringindo/Bloqueando o login

Isto é controlado pelo arquivo `/etc/security/access.conf`. O formato deste arquivo consistem em três campos separados por ":":

- Primeiro campo - Garante ("+") ou bloqueia ("-") o acesso caso as condições nos outros campos confirmem.
- Segundo campo - Contém o login, grupo. O formato `usuário@computador` pode ser usado para conferir com usuários que acessam de determinadas máquinas. Caso existam mais de um parâmetro, estes devem ser separados usando espaços. As palavras chave ALL (todos) e EXCEPT (exceção) e console também podem ser usadas.
- Terceiro campo - Lista de terminais (tty - na forma listada pelo **ttynome**), nomes de máquinas, nomes de domínios (começando com "."), endereços IP ou FQDN, porção de rede (finalizando com um "."). As palavras chave ALL (todos) e LOCAL (máquinas na mesma rede) também podem ser usadas.

**OBS1:** - A configuração padrão do `access.conf` é garantir o acesso a todos os usuários, através de qualquer lugar (permissiva).

**OBS2:** Mesmo se existir uma regra autorizando o acesso ao usuário, as restantes serão verificadas em busca de uma que bloqueie o acesso do usuário. Se nenhuma regra conferir, o usuário terá acesso garantido.

**OBS3:** - O nome de grupo somente é checado quando nenhum nome de usuário confere com nenhum usuário logado no sistema.

**OBS4:** - Grupos/usuários NIS podem ser especificados precedendo o nome do usuário ou grupo por uma "@".

Abaixo uma configuração restrita de `/etc/security/access.conf`:

```
#
Desabilita o login de todos os usuários EXCETO o root no terminal tty1
-:ALL EXCEPT root:tty1

Permite o login no console de todos os usuários especificados.
+:gleydson root:console

Conexões vindas da rede *.debian.org e *.debian.org.br de usuários pertencendo
ao grupo operadores são consideradas seguras (exceto para o usuário root).
+:operadores EXCEPT root: .debian.org .debian.org.br

Qualquer outra tentativa de acesso não definida acima é bloqueada imediatamente.
-: ALL: ALL
```

## Restringindo o acesso a root no su

A restrição de acesso a usuário `root` pelo PAM funciona permitindo que somente alguns usuários que pertençam a um grupo criado pelo administrador possam se tornar o superusuário usando o comando **su**. Esta restrição funciona até mesmo para os usuários que possuem a senha correta de `root`, retornando uma mensagem de login ou senha incorretos. Isto é extremamente útil para restrições de acesso.

Um outro ponto positivo é caso ocorra um possível acesso não autorizado em seu sistema ou um daemon seja corrompido e o atacante cair em um shell, ele não poderá obter `root` na máquina pois o UID do

daemon provavelmente não terá autorização. A distribuição **Debian**, em especial, possui grupos e nomes de usuários organizados de forma a permitir segurança e separação total caso utilize este mecanismo.

Este recurso se mostra bem eficiente para proteger a integridade da máquina até mesmo no comprometimento de máquinas que possui a senha semelhante, somente se usado em conjunto com as restrições de acesso de outros serviços remotos (como o **ssh**, **ftp**, etc). O guia Foca documenta as formas de restrição e seu impacto na segurança da máquina nos capítulos do nível Avançado (veja o índice para buscar o capítulo correspondente ao que deseja proteger).

Para configurar esta restrição, siga os seguintes passos:

- Crie um grupo onde os usuários cadastrados terão acesso `root`. Por exemplo, `usuarios-su` (ou algo mais discreto).
- Edite o arquivo `/etc/pam.d/su`. Insira a seguinte linha (caso não existir) no arquivo de configuração:

```
auth required pam_wheel.so group=usuarios-su
```

O que ela faz é usar o módulo `pam_wheel.so` requerendo que os usuários pertençam ao grupo `usuarios-su`. Salve e saia do editor.

- Ainda como usuário `root`, adicione os usuários que terão acesso a `root` no grupo `usuarios-su`. Recomendo que adicione seu usuário primeiro, principalmente se estiver fazendo acesso remoto, pois se acontecer uma queda no link não ficará sem acesso `root` por cair na restrição :-)
- Tente pegar o `root` com outros usuários que não pertençam ao grupo `usuarios-su` estes simplesmente terão o acesso negado.

## Restrições de serviços PAM baseados em dia/hora

Estas restrições são controladas pelo arquivo `/etc/security/time.conf`, a sintaxe deste arquivo é quatro campos separados por ";" :

- **Primeiro campo** - Nome do serviço PAM que será controlado (um dos serviços contidos em `/etc/pam.d`).
- **Segundo campo** - Lista de nomes de terminais que a regra que aplicará. O sinal "&" tem a função *and*, "|" tem a função *or* e "!" especifica uma exceção.
- **Terceiro campo** - Nome de usuários afetados pela regra. O sinal "&" tem a função *and*, "|" tem a função *or* e "!" especifica uma exceção.

**OBS:** O "\*" poderá ser usado somente no primeiro, segundo ou terceiro campo em uma mesma regra.

- **Quarto campo** - DiaSemana/faixa-de-horas que a restrição se aplicará. O dia da semana é especificado em duas letras:
  - `Mo` - Segunda-feira
  - `Tu` - Terça-feira
  - `We` - Quarta-feira
  - `Th` - Quinta-feira
  - `Fr` - Sexta-feira

- Sa - Sábado
- Su - Domingo
- Wk - Todos os dias da semana
- Wd - Somente sábado e domingo (fim de semana)
- Al - Todos os dias

O sinal "!" especifica uma exceção. A faixa de horas é especificada após o dia no formato HHMM-HHMM. Por exemplo:

```
MoTuWe0000-2400 - Segundas, terças e quartas
MoFrSu0800-1900- - Segundas, sextas e domingo das 08:00 da manhã as 19:00 da noite
FrFr0500-0600 - Não será realizada na sexta (especificações repetidas são anuladas)
de 05:00 as 06:00.
WkWe0731-1456 - Todos os dias da semana a partir de Quarta de 07:31 da manhã as
14:56 da tarde.
AlMo0000-2400 - Todos os dias da semana, exceto segunda-feira.
```

Por padrão o acesso é garantido a todos os usuários. Abaixo um exemplo de restrições usando o `/etc/security/time.conf`:

```
Bloqueia o login do usuário user1 ou user2 em qualquer tty, a restrição
durante todos os dias de 00:00 as 06:30
login;tty*;user1|user2;!Al0000-0630

Bloqueia o acesso do usuário root ao serviço login nos terminais tty*
(e não nos terminais ttyp*) nos finais de semana.
login;tty* & !ttyp*;root;!Wd0000-2400

O usuário 1 não poderá efetuar o login as terças feiras de 00:00 as 06:00
login;!tty*;user1;Tu0000-0600
```

**OBS1:** Mesmo se existir uma regra autorizando o acesso ao usuário, as restantes serão verificadas em busca de uma que bloqueie o acesso do usuário. Se nenhuma regra conferir, o usuário terá acesso garantido.

**OBS2:** Quando as restrições de tempo são ativadas no `/etc/security/time.conf`, o daemon **logoutd** poderá ser ativado manualmente (através de `/etc/init.d/logoutd`) para monitorar as restrições neste arquivo, forçando o logout de usuário de acordo com as configurações do `/etc/security/time.conf`. Isto ocorrerá automaticamente na próxima vez que iniciar o sistema (a distribuição detecta a presença de restrições de tempo no arquivo `/etc/security/time.conf` para decidir se deve ou não carregar este daemon).

Quando não está em execução, os limites de tempo são verificados somente no login do usuário, ele poderá ultrapassar este tempo sem ser desconectado do sistema.

## Permitindo acesso a grupos extras

Este recurso é controlado pelo arquivo `/etc/security/group.conf`. Este arquivo é composto por 5 campos separados por ";" (os 4 primeiros são os mesmos explicados em "Restrições de serviços PAM

baseados em dia/hora". O 5o campo contém um ou mais grupos (separados por espaços ou vírgulas) que serão adicionados aos grupos do usuário quando as condições dos campos anteriores conferirem.

**OBS:** Se o usuário escrever um programa que chama um interpretador de comandos e der a permissão SGID (chmod g+s programa), ele terá acesso àquele grupo na hora que quiser. Restrinja o uso de grupos somente a usuários de confiança ou crie grupos específicos para evitar problemas.

Exemplo de configuração do arquivo `/etc/security/group.conf`:

```
Permite que o usuário gleydson tenha acesso ao grupo floppy efetuando o login
entre 08:00 da manhã e 19:00 da noite
login;tty*;gleydson;A10800-1900;floppy

Todos os usuários podem ter acesso ao grupo games e sound aos sábados e domingos
login;tty*;*;SaSu0000-2400;sound games

Todos os usuários podem ter acesso ao grupo games e sound todos os dias
de 18:00 as 05:00 da manhã (fora do horário de expediente ;-)
login;tty*;*;A11800-0500;sound,games

Backups são permitidos fora do horário de expediente (para não sobrecarregar
a CPU e evitar o uso excessivo de disco).
login;tty*;gleydson;A11830-2400;backup
```

**OBS1:** Mesmo que uma regra confira com o usuário, as outras também serão verificadas para garantir acesso grupos extras.

**OBS2:** O padrão na maioria das distribuições é limitar o número máximo de grupos do usuário para 32. Caso precise aumentar este limite, será necessário recompilar o kernel (e também a glibc, se necessário) para aceitar um número maior modificando a variável `ngroup`.

## Limitação de recursos do shell

Estas restrições são especificadas no arquivo `/etc/security/limits.conf`. Seu formato consiste em 4 campos separados por ou ou mais espaços:

- Primeiro campo - Especifica o nome de usuário, um nome de grupo (@grupo) ou um "\*" especificando que as restrições nos outros campos se aplicam a todos os grupos e todos os usuários.
- Segundo campo - Tipo de restrição:
  - `soft` - Limite suave de bloqueio.
  - `hard` - Limite rígido de bloqueio.
  - - - Quando o tipo de restrição não se aplica ao Ítem que deseja restringir o acesso.

Quando somente o limite "hard" (rígido) é especificado, o limite suave assume o mesmo valor.

- Terceiro campo - Ítem que deseja restringir o acesso:
  - `core` - Limita o tamanho do arquivo core (KB)
  - `data` - Tamanho máximo de arquivo de dados (KB)
  - `fsize` - Tamanho máximo de arquivo (KB)

- `memlock` - Tamanho máximo do espaço de endereços bloqueado na memória (KB)
- `nofile` - Número máximo de arquivos abertos
- `rss` - Tamanho máximo residente (KB)
- `stack` - Tamanho máximo da pilha (KB)
- `cpu` - Tempo máximo de uso da CPU (MIN)
- `nproc` - Número máximo de processos
- `as` - Limite de espaço de endereços
- `maxlogins` - Número máximo de logins
- `priority` - Prioridade de execução de processos de usuários
- Quarto campo - Especifica o valor do campo anterior

Os limites aplicados ao usuário podem ser visualizados através do comando `ulimit -S -a` (para listar limites suaves - soft) e `ulimit -H -a` (para listar limites rígidos - hard). Caso o parâmetro `-S` ou `-H` sejam omitidos, os limites listados serão os suaves (soft). Um exemplo de `/etc/security/limits.conf` (retirado da distribuição **Debian GNU/Linux**):

```
* soft core 0
* hard rss 10000
@student hard nproc 20
@faculty soft nproc 20
@faculty hard nproc 50
ftp hard nproc 0
@student - maxlogins 4
gleydson - maxlogins 2
```

**OBS:** Estas permissões passam a ter efeito no momento que o usuário se conecta ao sistema, e não quando elas são modificadas no arquivo `/etc/security/limits.conf`.

## Restrições de acesso a programas/diretórios/ arquivos usando grupos

Usuários podem ter o acesso liberado a diretórios/arquivos execução de programas de acordo com o grupo que pertencem. Este é um recurso valioso na administração de sistemas **Unix** que se bem usado, aumenta as restrições de acesso e segurança no acesso/utilização de programas em um ambiente de trabalho. Usuários de sistema tendem a usar o usuário `root` para fazer tarefas como conexão com internet, utilização da placa de som, modem, etc. e as vezes nem sabem que isso pode ser feito através do mesmo usuário adicionando este a um grupo específico.

Esta tarefa pode ser feita com o comando `adduser usuário grupo` ou editando manualmente os arquivos `/etc/group` e `/etc/gshadow`. Podemos ter as seguintes situações facilitadas com o uso de grupos:

- Usar a placa de som. Os dispositivos usados pela placa de som como `/dev/audio`, `/dev/dsp`, `/dev/sndstat`, etc. normalmente tem permissão leitura/gravação para o usuário `root` e grupo `audio`

(cheque com o comando `ls -la /dev/audio`). Para autorizar determinados usuários usar a placa de som basta adicioná-los neste grupo: `adduser usuario audio`.

- Conectar a Internet. Normalmente o utilitário `ppp` tem as permissões `SUID root` e grupo `dip`. Adicionamos o usuário a este grupo: `adduser usuario dip`. Agora ele poderá conectar/desconectar a internet sem a intervenção do usuário `root`.

**OBS** Certamente o usuário terá acesso aos arquivos de configuração da discagem do **ppp** e conseqüentemente a senha de conexão internet, e esta senha é a mesma usada no e-mail primário do provedor (com o mesmo nome da conta). Esta mesma situação pode acontecer com outros programas que autorize o acesso a grupos, é importante que conheça bem as permissões do programa e entender se existem riscos.

- Utilizar o modem. Um bom grupo para permitir a utilização do modem é `dialout`. O dispositivo utilizado pelo modem (não seu link) deve ter permissões leitura/gravação para o usuário `root` e grupo `dialout`. Cadastrando o usuário neste grupo autorizará a utilização do modem: `adduser usuario dialout`.
- Permitir que diversos usuários compartilhem um mesmo diretório. Isto é útil quando muitas pessoas estão desenvolvendo um mesmo projeto. Siga estes passos:
  - Crie um novo grupo no sistema: **groupadd gp1**, a opção `-g` permite selecionar manualmente a GID. Opcionalmente você poderá usar um grupo já existente no sistema (veja o arquivo `/etc/group`).
  - Crie o diretório que será usado para armazenar os arquivos deste grupo de usuários: `mkdir projeto1`.
  - Mude o dono/grupo do diretório: `chown root.gp1 projeto1/`
  - De permissões de leitura/gravação para o dono/grupo do diretório, vamos também incluir a permissão `SGID` para que todos os arquivos criados dentro deste diretório pertençam ao mesmo grupo e não ao grupo primário do usuário, assim todos os usuários terão acesso: `chmod 2770 projeto1`
  - Agora cadastre os usuários que deverão ter acesso ao diretório `projeto1/` no grupo `gp1`, somente estes usuários e o `root` terão acesso ao diretório (permissões `2770`).
  - É interessante também mudar a "umask" do usuário de `022` para `002` (ou equivalente) para que os novos arquivos criados tenham permissão de leitura/gravação para o grupo `gp1`. Caso contrário, lembre-se de modificar as permissões de seus arquivos manualmente. Um ótimo comando para fazer isso (sem afetar diretórios) é: `find . -type f -user usuario1 -exec chmod 0660 \{\} \;`. Este comando parece estranho mas é excelente! um `chmod -R 0660` afetaria até os diretórios, imagine o caos.

A maioria das distribuições **Linux** vem com uma boa política de grupos para permitir um controle eficaz de recurso. Se você quer saber quais arquivos em seu sistema pertencem a determinado grupo (útil para saber o que o usuário terá acesso se adicioná-lo àquele grupo) execute o comando:

```
find / -group nome_do_grupo
```

## Dando poderes de root para executar determinados programas

A ferramenta ideal para isto é o **sudo**. Através dela é possível permitir um usuário comum executar um comando como **root** e registrar quando isto foi feito. É possível selecionar os usuários/grupos que

terão acesso e quais aplicativos que poderão ser usados, estas configurações são feitas no arquivo `/etc/sudoers`.

Por exemplo, para o usuário "john" usar o comando `shutdown` para desligar o computador: `sudo shutdown -h now`.

O **sudo** é um programa muito completo, tomaria muitos Kilobytes neste guia. Recomendo dar uma lida na página de manual para entender como as variáveis do arquivo de configuração funcionam. Mesmo assim aqui vai um exemplo simples deste arquivo para iniciar rapidamente o uso do **sudo**:

```
arquivo sudoers.
#
Edite este arquivo com o comando 'visudo' como root
#
#
Especificação de máquinas. O primeiro campo (Host_Alias) diz que a variável
LOCALSERVER será um nome/endereço de máquina
Host_Alias LOCALSERVER=192.168.0.1
Especificação de usuários. O primeiro campo (User_Alias) diz que a variável
NETMASTERS armazenará nomes de usuários
User_Alias NETMASTERS=gleydson, goodboy
Comandos. O primeiro campo (Cmnd_Alias) diz que a variável
C_REDE contém comandos do sistema. Mais de um parâmetro
deve ser separado por vírgulas
Cmnd_Alias C_REDE=/sbin/ipchains, /sbin/iptables
Padrões que se aplicam aos usuários da variável NETMASTERS. O parâmetro
mail_always sempre envia um e-mail ao root avisando sobre o uso do
sudo
Defaults:NETMASTERS mail_always
As linha que começam com o nome de usuário ou variável "User_Alias"
definem o acesso aos recursos. O primeiro campo é o usuário, o segundo
o endereço de acesso (opcionalmente seguido de um sinal "=" para
especificar opções adicionais) o terceiro o comando ou lista de comandos
#
O usuário root não tem restrições
root ALL=(ALL) ALL
Permite que os usuários especificados na variável NETMASTERS
acessando dos locais em LOCALSERVER utilizem os comandos
em C_REDE (sem fornecer senha).
NETMASTERS LOCALSERVER=NOPASSWD: C_REDE
```

Edite este arquivo com o comando **visudo**, ele faz algumas checagens para detectar problemas de configuração. Para listar os comandos disponíveis para o usuário no **sudo**, utilize a opção `-l`, ex: `sudo -l`.

## Restringindo o comando su

Restrições de acesso através de grupos, bloqueio de acesso, acesso direto sem senha, etc. podem ser aplicados ao sudo via seu arquivo de configuração PAM `/etc/pam.d/su`. Abaixo um exemplo explicativo deste arquivo:

```
A configuração abaixo requer que o usuário seja membro do
grupo adm para usar o 'su'.
auth required pam_wheel.so group=adm

Membros do grupo acima não precisam fornecer senha, temos confiança neles.
auth sufficient pam_wheel.so trust

Usuário que pertencem ao grupo "nosu" nunca deverão ter acesso ao 'su'
auth required pam_wheel.so deny group=nosu

O root não precisa fornecer senha ao 'su'
auth sufficient pam_rootok.so

Ativa as restrições PAM de /etc/security/limits.conf
session required pam_limits.so

Isto ativa as restrições PAM de /etc/security/time.conf no
comando 'su'
account requisite pam_time.so

Módulos padrões de autenticação Unix
auth required pam_unix.so
account required pam_unix.so
session required pam_unix.so
```

## Restrições baseadas em usuário/IP

O serviço **identd** permite identificar os usuários que estão realizando conexões TCP, adicionalmente esta característica é usada por programas para fazer restrições para usuários em adição ao endereço de origem/destino. A sintaxe usada nas diretivas de acesso é especificada na forma *usuário@endereço*. O ??? explica a configuração/utilização/vulnerabilidades e recomendações sobre este serviço.

Diversos programas que possuem controle de acesso baseado em IP's/hosts aceitam esta especificação, como o **exim**, **ircd**, e o conhecido **tcpd**.

Segue um exemplo da utilização do **identd** com o arquivo `hosts.allow`:

```
Permite o acesso ao serviço de telnet somente ao usuário gleydson conectando
a partir da máquina com IP 192.168.1.1
in.telnetd: gleydson@192.168.1.1

Permite o acesso ao serviço ftp somente ao usuário gleydson conectando de
qualquer máquina da rede 192.168.1.*
in.ftpd: gleydson@192.168.1.*
```

Note que a utilização do **identd** torna a utilização do serviço um pouco mais restrita, somente conhecendo os "logins" de quem tem acesso ao serviço, um cracker conseguirá ter acesso ao mesmo serviço naquele sistema (este é um dos motivos que é recomendado sempre divulgar o mínimo detalhes possíveis sobre o sistema para minimizar riscos de ataques).

Veja mais detalhes sobre o uso do **identd** em ???.

## Restrições por MAC Address/IP

Esta proteção oferece uma barreira maior se segurança contra IPs spoofing evitando que pessoas mal intencionadas façam um IP spoofing da máquina para obter acessos privilegiados que somente o detentor original do MAC/IP teria. Recomendo não levar em consideração que isto seja a solução definitiva contra IP spoofing, pois é possível falsificar o MAC address de uma interface para tomar outra identidade.

Este método poderá ser aplicado para fornecer um maior laço de confiança por hardware entre as máquinas que compõem uma rede de servidores. Ele também evita mesmo que uma máquina configurada de forma errônea tenha acesso indevido ao servidor ou em uma situação extrema, se torne o gateway da rede.

Para restringir as conexões para uma máquina **Linux** por MAC address, utilize o firewall **iptables**. Com ele será permitido fazer a restrição por serviços, criando uma barreira bastante chata para crackers tentarem se conectar a um serviço. Como referência, leia a seção "Especificando o endereço MAC da interface".

Outra situação é a restrição por par MAC/IP usando o próprio cache arp da máquina, usando entradas estáticas de endereços. Um exemplo deste uso é quando você é extremamente paranóico ou quando uma rede que utiliza algum método de autenticação baseado no `rhosts` (como é o caso do sistema de backup do **Amanda**), então é importante dizer para as máquinas servidoras, qual o MAC address/IP privilegiado que terá o acesso ao usuário para conexão sem senha.

O local padronizado para definir um MAC estático (e bastante desconhecido da maioria dos administradores de sistemas) é o `/etc/ethers`. O formato deste arquivo é o MAC Address e IP separados por espaço, cada linha com uma nova entrada de MAC Address. Veja o exemplo:

```
00:03:47:AA:AA:AB www.focalinux.org.br
00:03:47:BB:AA:BA www2.focalinux.org.br
00:03:47:BB:AA:BB 192.168.0.1
```

Caso não conheça o formato do endereço de MAC Address, os três primeiros 3 campos definem o fabricante da placa de rede, e os 3 últimos é uma identificação única do fabricante para a Placa, ou seja, NENHUMA placa de rede fabricada tem o mesmo MAC Address físico.

Para que o comando **arp** crie as entradas estáticas no seu cache ARP, será necessário executar o comando `arp -f /etc/ethers`. Este comando poderá ser colocado em algum script ou diretório de inicialização de sua distribuição para que seja executado automaticamente (como por exemplo, no `/etc/rc.boot` da **Debian**). Digitando `arp` você verá as linhas definidas no arquivo `/etc/ethers` marcadas com as opção (flag) **M** (manual/permanente). Outra forma de verificar, é usando o `arp -a máquina` ou somente `arp -a`. As máquinas especificadas estaticamente (manualmente) terão o nome **PERM** listados (cache arp permanente).

**OBS:** Como deve ter notado, a restrição por MAC Address implica em um aumento no trabalho de gerenciamento das configurações. Assim, planeje-se para que esta tarefa não seja desgastante, crie programas para realizar atualizações dinâmicas estudando a estrutura de sua rede e como suas máquinas se comunicam para não ter problemas obscuros quando tiver que fazer uma simples modificação em uma interface de rede :)

Uma boa configuração restritiva requer análise sobre os impactos na rede.

## Desabilitando serviços não usados no Inetd

Desative todos os serviços que não serão utilizados no arquivo `/etc/inetd.conf`, isto diminui bastante as possibilidades de ataques em seu sistema. Os nomes de serviços são os parâmetros especificados na primeira coluna do arquivo `/etc/inetd.conf` (por exemplo, `talk`, `ircd`, `pop3`, `auth`, `smtp`).

Para desativar serviços neste arquivo, ponha o símbolo `"#"` no início das linhas que deseja comentar e execute um `killall -HUP inetd`. Alternativamente, o comando `update-inetd` pode ser usado para facilitar esta tarefa:

```
update-inetd --disable finger,talk,time,daytime
```

```
update-inetd --disable
```

Este comando envia automaticamente o sinal de reinício (HUP) ao **inetd**. O serviço poderá ser novamente ativado substituindo a opção `--disable` por `--enable` ou retirando o trecho `"#<off>#"` no começo da linha do serviço do `/etc/inetd.conf`.

## Evitando o uso de `hosts.equiv` e `.rhosts`

O arquivo `hosts.equiv` contém uma lista de usuários autorizados/desautorizados que podem fazer uso dos serviços `"r*"` sem fornecer uma senha (como **rsh**, **rcp**, **rexec**, etc) , veja `"/etc/hosts.equiv e /etc/shosts.equiv"`. É muito fácil falsificar um nome de usuário para obter acesso aos privilégios de outro usuário usando este recurso.

Os arquivos `~/ .rhosts`, `~/ .shosts` tem o funcionamento parecido com o `hosts.equiv` mas contém somente dois campos, o primeiro especificando o nome do computador (FQDN) e o segundo o nome do usuário que tem permissão de acesso sem fornecer senha. Ele garantirá este acesso ao usuário e máquina remota especificada neste arquivo. Se for definido somente o nome do computador, o nome de usuário deverá ser o mesmo do local para que o acesso sem senha seja garantido. É recomendável restringir o acesso a estes arquivos somente ao usuário/grupo quando for realmente necessário. Um exemplo de `~/ .rhosts`:

```
maquina1.dominio.com.br usuario1
maquina2.dominio.com.br usuario2
```

O uso destes dois mecanismos e dos serviços `"r*"` são desencorajados! (o último por usar transferência de dados não criptografadas). Veja ??? para uma alternativa melhor. Utilize estes dois mecanismos somente se deseja facilidade no gerenciamento e se sua rede seja absolutamente confiável e a segurança de dados não seja prioridade pra você...

## Restringindo o uso do shutdown

Por padrão todos que tem acesso ao console do sistema podem efetuar o reinício do computador pressionando `CTRL+ALT+DEL`. Estas teclas de combinação são definidas pela linha

```
ca:12345:ctrlaltdel:/sbin/shutdown -r now
```

do arquivo `/etc/inittab`. A opção `-a` (access) do **shutdown** restringe isto, permitindo somente o reinício do sistema caso um dos usuários cadastrados no arquivo `/etc/shutdown.allow` estejam logados no console. Caso nenhum usuário autorizado esteja logado, a mensagem `shutdown: no authorized users logged in` é exibida no console local.

O arquivo `/etc/shutdown.allow` deve conter um usuário por linha e 32 no máximo.

A mesma linha do `/etc/inittab` pode ser modificada para a seguinte:

```
ca:12345:ctrlaltdel:/sbin/shutdown -a -t5 -r now
```

**OBS:** Se a opção `-a` seja especificada e o arquivo `/etc/shutdown.allow` não existe, a opção `-a` é ignorada.

## Restringindo o acesso ao sistema de arquivos `/proc`

O patch *restricted proc fs* é um dos melhores para realizar esta tarefa. Restringindo o acesso ao sistema de arquivos `/proc` evita que o usuário normal tenha acesso aos detalhes sobre processos de outros (com `ps aux`) ou acesso a detalhes de processos de outros usuários existentes nos subdiretórios numéricos (equivalentes a PID) em `/proc`. Abaixo algumas características do patch *restricted proc fs*:

- É pequeno, rápido e faz poucas modificações no fonte do kernel.
- Seu método de funcionamento é baseado nas restrições de dono/grupo (nativas de ambiente **Unix**).
- Restringe a visualização de processos só dos usuários. Adicionalmente será especificada uma GID para o diretório `/proc`, qualquer usuário que pertença ao grupo especificado poderá visualizar todos os processos e entrar em qualquer diretório do kernel (sem restrições, como se não tivesse o patch).
- Muito estável e confiável.

Este patch deve ser baixado de <http://noc.res.cmu.edu/proc>, existem versões para os kernels da série 2.2 e 2.4, baixe e aplique o patch, na configuração do kernel ative a opção `Restricted Proc fs support`. Compile e instale seu kernel.

No arquivo `/etc/fstab` inclua um grupo para a montagem do sistema de arquivos `/proc` (vamos usar o grupo `adm` com a GID 4):

```
/etc/fstab: informações estáticas do sistemas de arquivos.
#
<Sist. Arq.> <Ponto Mont.> <tipo> <opções> <dump> <passo>
proc /proc proc defaults,gid=4 0 0
```

Após reiniciar o sistema, execute o comando `ls -lad /proc` note que o grupo do diretório `/proc` será modificado para `adm`. Agora entre como um usuário e execute um `ps aux`, somente seus processos serão listados. Para autorizar um usuário específico ver todos os processos (ter acesso novamente ao diretório `/proc`), inclua este no grupo que usou no arquivo `/etc/fstab`:

```
adduser usuario adm
```

Após efetuar o usuário já estará pertencendo ao grupo `adm` (confira digitando `groups`), e poderá ver novamente os processos de todos os usuários com o comando `ps aux`.

**OBS1:** Incluir um usuário no grupo adm É PERIGOSO, porque este usuário poderá ter acesso a arquivos/diretórios que pertençam a este grupo, como os arquivos/diretórios em `/var/log`. Se esta não é sua intenção, crie um grupo independente como `restrproc` para controlar quem terá acesso ao diretório `/proc`: `addgroup restrproc`.

**OBS2:** Se a opção `gid` não for especificada para a montagem de `/proc` no `/etc/fstab`, o grupo `root` será usado como padrão. NUNCA adicione usuários ao grupo `root`, use o método da observação acima para permitir outros usuários ver todos os processos em execução.

**OBS3** Caso o servidor `identd` esteja sendo usado na máquina servidora, será necessário roda-lo com a mesma GID do diretório `/proc` para que continue funcionando. Se ele é executado como daemon, adicione a opção `-g GRUPO` no script que inicia o serviço em `/etc/init.d` e reinicie o daemon. Caso ele seja iniciado via `inetd`, faça a seguinte modificação no arquivo `/etc/inetd.conf` (assumindo o uso do `oidentd`):

```
#:INFO: Info services
auth stream tcp nowait.40 nobody.adm /usr/sbin/oidentd oidentd -q -i -t 40
```

Veja ??? para detalhes sobre este serviço.

## Limitando o uso de espaço em disco (quotas)

O sistema de **quotas** é usado para limitar o espaço em disco disponível a usuários/grupo. O uso de partições independentes para o diretório `/home` e outros montados separadamente não é muito eficaz porque muitos usuários serão prejudicados se a partição for totalmente ocupada e alguns possuem requerimentos de uso maior do que outros.

O suporte a *Quotas* deve estar compilado no kernel (seção *FileSystems*) e o sistema de arquivos deverá ser do tipo *ext2* ou *XFS* para funcionar.

## Instalando o sistema de quotas

Abaixo o passo a passo para a instalação de quotas em seu sistema:

1. Recompile seu kernel com suporte a quota. Habilite a opção "Quota support" na seção "FileSystems" na configuração de recursos do seu kernel.
2. Instale o pacote no sistema (`apt-get install quota`).
3. Habilite a quota para os sistemas de arquivos que deseja restringir no arquivo `/etc/fstab`:

|                        |                    |                   |                                         |                |                |
|------------------------|--------------------|-------------------|-----------------------------------------|----------------|----------------|
| <code>/dev/hda1</code> | <code>/boot</code> | <code>ext2</code> | <code>defaults</code>                   | <code>1</code> | <code>1</code> |
| <code>/dev/hda3</code> | <code>/</code>     | <code>ext2</code> | <code>defaults,usrquota</code>          | <code>1</code> | <code>2</code> |
| <code>/dev/hda4</code> | <code>/usr</code>  | <code>ext2</code> | <code>defaults,grpquota</code>          | <code>1</code> | <code>3</code> |
| <code>/dev/hda5</code> | <code>/pub</code>  | <code>ext2</code> | <code>defaults,usrquota,grpquota</code> | <code>1</code> | <code>4</code> |

O sistema de arquivos `/dev/hda1` não terá suporte a quota, `/dev/hda3` terá suporte a quotas de usuários (`usrquota`), `/dev/hda4` terá suporte a quotas de grupos (`grpquota`) e `/dev/hda5` terá suporte a ambos. Por padrão é assumido que os arquivos de controle de quota estão localizados no ponto de montagem da partição com os nomes `quota.user` e `quota.group`.

4. Agora será necessário criar os arquivos `quota.user` e `quota.group` no ponto de montagem de cada partição `ext2` acima que utilizará o recurso de quotas. O arquivo `quota.user` controla as quotas de usuários e `quota.group` controla as quotas de grupos.

- Crie um arquivo vazio `quota.user` em `/` (terá suporte somente a quota de usuários, veja a opção de montagem no `/etc/fstab`): `touch /quota.user` ou `echo -n >/quota.user`.
- Crie um arquivo vazio `quota.group` em `/usr` (terá suporte somente a quota de grupos): `touch /usr/quota.group` ou `echo -n >/usr/quota.group`.
- Crie um arquivo vazio `quota.user` e `quota.group` em `/pub` (este sistema de arquivos tem suporte a ambos os tipos de quota): `touch /pub/quota.user /pub/quota.group`.

Por motivos de segurança, as permissões dos arquivos de controle de quota `quota.user` e `quota.group` devem ser leitura/gravação ao usuário `root` e sem permissões para grupo/outras usuários: `chmod 0600 /quota.user /quota.group`.

**OBS:** Se deseja utilizar o quota versão 1, certifique-se que não existem os arquivos chamados `aquota.user` e `aquota.group` no diretório raiz de sua partição. Se eles estiverem disponíveis, os utilitários de quota utilizarão esta versão como padrão, atualmente o kernel 2.4 possui somente suporte a quota versão 1.

A versão 2 do quota checa corrompimento dos arquivos de dados de quota e trabalha mais rápido em partições grandes. São necessários patches da série "ac" (Alan Cox) para usar a versão 2 do quota.

5. Entre em modo monousuário `init 1`, desmonte os sistemas de arquivos que utilizarão a quota e monte-os novamente (isto serve para ativar as opções de quota). Alternativamente, execute `umount -a` (para desmontar todos os sistemas de arquivos) e `mount -a` para remontar todos.

Se você ativou as quotas para o sistema de arquivos `/` (como em nosso exemplo) será necessário reiniciar o sistema.

6. O próximo passo é scanear o disco para criar os dados para as partições com suporte a quota (ativadas no `/etc/fstab`):

```
quotacheck -augv
```

O parâmetro `-a` diz para checar todas as partições com suporte a quota no arquivo `/etc/mstab`, `-u` para checar quotas de usuários, `-g` para checar grupos e `-v` para mostrar o progresso da checagem da partição.

Na primeira execução é mostrado uma mensagem de erro de arquivo `quota.user/quota.group` corrompido, mas isto é normal porque o arquivo anterior tem tamanho zero. Estes nomes também servem para o **quotacheck** "auto-detectar" a versão do sistema de quota usada no sistema de arquivos.

**OBS:** Certamente será necessário "forçar" a remontagem como somente leitura do sistema de arquivos `/` com a opção `-m` para o **quotacheck** criar as configurações de quota nesta partição.

7. Agora resta ativar o suporte as quotas de disco em todas as partições (`-a`) com recurso de quota especificado (no `/etc/mstab`):

```
quotaon -augv
```

As opções possuem o mesmo significado do comando **quotacheck**. O utilitário **quotaoff** serve para desativar quotas de usuários e usa as mesmas opções do **quotaon**. Estes três utilitários somente podem ser usados pelo usuário `root`. As opções de quota podem ser especificadas independente para cada sistema de arquivos:

```
Ativa o suporte a quota em /pub (somente grupos de usuários no momento).
quotaon -gv /pub

Ativa as quotas de usuários em /pub
quotaon -uv /pub

Desativa as quotas de grupos em /pub (deixando somente a de usuários ativa)
quotaoff -gv /pub
```

A atualização de quotas durante a gravação/exclusão de arquivos é feita automaticamente. O utilitário **quotacheck** deverá ser executado sempre que o sistema de quotas for desativado (por não haver atualização automática dos dados de uso de disco) ou quando ocorrerem falhas de disco.

Na distribuição **Debian** o **quotacheck** é disparado sempre que necessário após as situações de checagem de disco. As quotas de todas as partições também são ativadas automaticamente pelo script `/etc/init.d/quota` e `/etc/init.d/quotarpc`.

Em sistemas que utilizam NFS e possuem sistemas de arquivos exportados em `/etc/exports`, o daemon **rpc.rquotad** deverá ser carregado. Sua função é fornecer os detalhes de **quota** dos sistemas de arquivos locais exportados para as máquinas clientes.

## Editando quotas de usuários/grupos

O programa **edquota** é usado pelo `root` para editar as quotas de usuários/grupos. Por padrão, todos os usuários/grupos do sistema não possuem quotas. Sua sintaxe é a seguinte

```
edquota [opções] [usuário/grupo]
```

As opções podem ser:

- u Edita a quota do usuário especificado (esta é a padrão).
- g Edita a quota de grupo especificado.
- r Permite editar a quota de sistemas de arquivos remotos através do daemon **rpc.rquotad**.
- p [usuário/grupo] Usa os valores especificados para o *usuário/grupo* para definir a nova quota, sem necessidade de entrar no modo de edição.
- t Permite modificar o valor de tolerância dos limites que ultrapassam *soft* até que sejam bloqueados. Durante o tempo de tolerância, serão enviados somente avisos sobre a quota ultrapassada sem bloquear totalmente a gravação de arquivos (até que o limite *hard* seja atingido ou o tempo de tolerância seja ultrapassado).

Quando a quota *soft* do usuário/grupo é estourada, a mensagem "warning: user disk quota exceeded" será exibida. Quando a quota *hard* é ultrapassada, a gravação atual é interrompida e a mensagem "write failed, user disk limit reached" é mostrada ao usuário. Nenhuma nova gravação que ultrapasse a quota *hard* é permitida. Por exemplo, para modificar a quota do usuário gleydson: `edquota gleydson`

```
Disk quotas for user gleydson (uid 1000):
Filesystem blocks soft hard inodes soft
```

Restrições de acesso,  
recursos e serviços

---

|                    |        |        |        |       |      |
|--------------------|--------|--------|--------|-------|------|
| /dev/hda5<br>20000 | 504944 | 500100 | 600000 | 10868 | 1500 |
|--------------------|--------|--------|--------|-------|------|

O editor de textos usado poderá ser modificado através da variável `$EDITOR`. Abaixo a explicação destes campos:

- `Filesystem` - Sistema de arquivos que terá a quota do usuário/grupo editada. As restrições se aplicam individualmente de acordo com o sistema de arquivos.
- `blocks` - Número máximo de blocos (especificado em Kbytes) que o usuário possui atualmente. O usuário `gleydson` está usando atualmente 504944 Kbytes.
  - `soft` - Restrição mínima de espaço em disco usado. Atualmente 500100 Kb.
  - `hard` - Limite máximo aceitável de uso em disco para o usuário/grupo sendo editado. 600000 Kb atualmente. O sistema de quotas nunca deixará este limite ser ultrapassado.
- `inodes` - Número máximo de arquivos que o usuário possui atualmente na partição especificada. O usuário `gleydson` possui atualmente 10868 arquivos na partição `/pub`.
  - `soft` - Restrição mínima de número de arquivos que o usuário/grupo possui no disco. Atualmente em 15.000.
  - `hard` - Restrição máxima de número de arquivos que o usuário/grupo possui no disco. Atualmente em 20.000.

Para desativar as restrições coloque "0" no campo `soft` ou `hard`. Quando o limite `soft` é atingido, o usuário é alertado por ter ultrapassado sua quota com a mensagem "warning: user quota exceeded" (quota do usuário excedida). O programa `setquota` é uma programa não-interativo para edição de quotas para ser usado diretamente na linha de comando ou em shell scripts.

Após ultrapassar o limite `soft`, começa a contagem do tempo para que este passe a valer como limite `hard` (o máximo aceitável e que nunca poderá ser ultrapassado). O comando `edquota -t` serve para modificar estes valores na partição especificada:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem Block grace period Inode grace period
/dev/hda5 2days 7days
```

Abaixo a explicação destes campos:

- `Filesystem` - Sistema de arquivos que terá o período de tolerância modificado.
- `Block grade period` - Tempo máximo de tolerância para usuários/grupos que ultrapassaram sua quota `soft` de espaço em disco antes de passar a valer como `hard`. No exemplo, o usuário tem 2 dias para excluir possíveis arquivos ou contactar o administrador para redimensionar o tamanho de quota. O valor padrão é 7 dias.
- `Inode grade period` - Tempo máximo de tolerância para usuários/grupos que ultrapassaram sua quota `soft` de número de arquivos gravados antes de passar a valer como `hard`. No exemplo, o usuário tem 7 dias para excluir possíveis arquivos ou contactar o administrador para analisar seu tamanho de quota. O valor padrão é 7 dias.

**OBS1:** - O comando `quotacheck` deverá ser executado na partição sempre que novas restrições/limites forem editados com o `edquota`. Isto atualiza os arquivos `quota.user` e `quota.group`. Lembre-se de

desativar o sistema de quotas (`quotaoff -ugv /partição`) antes de executar este comando (para liberar totalmente a partição, **quotacheck** remonta a partição somente para leitura quando é executado). Por este motivo é recomendável fazer isso em modo monousuário.

**OBS2:** Quando o limite *soft* (suave) é excedido, o sistema começará a lhe mostrar mensagens alertando a passagem do limite (para lhe dar tempo de eliminar arquivos ou não ser pego desprevenido com o bloqueio de gravação) porque o limite *hard* (rígido) nunca poderá ser ultrapassado.

**OBS3:** - O tempo de tolerância restante ao usuário/grupo quando a quota é ultrapassada poder ser visualizada com o comando **quota** (veja “Verificando a quota disponível ao usuário”).

**OBS4:** - Quando o usuário exclui seus arquivos e volta a ficar abaixo dos limites *soft* da quota, o tempo de tolerância é resetado aos valores padrões (especificados por `edquota -t`).

**OBS5:** - As quotas de espaço em disco podem ser definidas automaticamente para os novos usuários adicionados ao sistema colocando o espaço em disco na variável `QUOTAUSER=numero` do arquivo `/etc/adduser.conf`. Isto será equivalente a digitar o comando `edquota -q QUOTA novo_usuario`.

## Modificando a quota de todos os usuários de uma vez

Editar manualmente a quota de cada usuário é uma tarefa trabalhosa quando se está instalando quotas e possui muitos usuários, existe uma maneira mais fácil de fazer isso usando o próprio **edquota** e um usuário com a quota já definida. Por exemplo, instalamos quota em nosso sistema e queremos que todos os 300 usuários tenham a quota de usuário de 10MB e de grupo de 15MB:

1. Criamos um usuário com esta quota usando o **edquota** (como descrito em “Editando quotas de usuários/grupos”). Como exemplo usaremos o usuário `teste_user`. Use o comando `quota teste_user` para verificar se as quotas para este usuário está correta.
2. Criamos um script que modifique a quota padrão de todos os usuários do sistema de uma só vez:

```
#!/bin/sh
cd /home
for USUARIO in *
do
edquota -u ${USUARIO} -p teste_user

done
```

Pronto, verifique a quota de todos os usuários com o comando `repquota -a`.

## Verificando a quota disponível ao usuário

Execute o comando **quota** mostra os limites de usuários/grupos e a tolerância restante antes do limite *soft* se tornar rígido. Abaixo alguns exemplos descritivos deste comando:

```
quota
```

```
Disk quotas for user gleydson (uid 1234):
 Filesystem blocks quota limit grace files quota limit grace
 /dev/hda5 504944* 500100 600000 00:05 10868 0 0
```

Os campos tem o seguinte significado:

- `filesystem` - Sistema de arquivos.
- `blocks` - Número de blocos usados atualmente na partição (em Kb). O "\*" indica que o limite foi ultrapassado. Atualmente em 504944.
- `quota` - Limite suave (*soft*) de espaço na partição que o usuário/grupo possui. Atualmente 500100. O valor 0 indica que o usuário/grupo não possui restrições.
- `limit` - Limite máximo (*hard*) de espaço na partição que o usuário/grupo possui. Atualmente em 600000. O valor 0 indica que o usuário/grupo não possui restrições.
- `grace` - Tolerância antes que o limite *soft* passe a valer como *hard* quando o espaço em disco é ultrapassado. Este usuário tem 5 minutos restantes para que isto ocorra. Quando o valor *soft* volta a ficar abaixo da quota, a tolerância é resetada.

O parâmetro "none" indica que o tempo de tolerância expirou (caso existam limitações de quota que foram ultrapassadas) ou que o usuário/grupo não possui restrições. Veja se existe um "\*" no campo `blocks`.

- `files` - Número máximo de arquivos que usuário/grupo possui atualmente na partição. Um "\*" indica que o limite foi ultrapassado. Atualmente em 10868.
- `quota` - Limite suave (*soft*) de número de arquivos na partição que o usuário/grupo possui. Atualmente ilimitado.
- `limit` - Limite máximo (*hard*) de número de arquivos na partição que o usuário/grupo possui. Atualmente ilimitado.
- `grace` - Tolerância antes que o limite *soft* passe a valer como *hard* para o número de arquivos ultrapassados. Como não existe quota para número de arquivos, não existe tolerância. A tolerância é resetada aos valores padrões quando o valor *soft* volta a ficar abaixo da quota.

A quota de outros usuários/grupos podem ser visualizadas especificando as opções `-u` (padrão) e `-g` na linha de comando respectivamente. A opção `-v` permite visualizar quotas em sistemas de arquivos não alocados e `-q` mostra somente uma mensagem dizendo se o usuário está ou não dentro de sua quota:

```
quota -u usuario
```

```
quota -uq usuario
```

```
quota -g users
```

Por motivos de segurança, você não poderá visualizar as quotas de outros usuários e grupos que não pertence (exceto para o usuário `root`).

## Verificando a quota de todos os usuários/grupos do sistema

Quando precisamos verificar o uso de quotas de todos os usuários/grupos do sistema o **quota** se torna incômodo e pouco prático. O comando **repquota** lista está disponível ao administrador para facilitar esta tarefa. Sua listagem é organizada por partições listando dados adicionais como `grace time` e aceita as mesmas opções dos utilitários **quotaon** e **quotaoff**. Primeiro são listados as restrições de usuários e depois

de grupos para a partição. (tolerância) As opções aceitas por este utilitário tem o mesmo significado das opções do **quotaon** e **quotaoff**:

```
repquota -aug
```

```
*** Report for user quotas on device /dev/hda3
```

```
Block grace time: 7days; Inode grace time: 7days
```

| User        |    | Block limits |      |      |       | File limits |      |      |       |
|-------------|----|--------------|------|------|-------|-------------|------|------|-------|
|             |    | used         | soft | hard | grace | used        | soft | hard | grace |
| root        | -- | 29160        | 0    | 0    | none  | 9970        | 0    | 0    | none  |
| daemon      | -- | 64           | 0    | 0    |       | 22          | 0    | 0    |       |
| man         | -- | 944          | 0    | 0    |       | 65          | 0    | 0    |       |
| mail        | -- | 4960         | 0    | 0    |       | 823         | 0    | 0    |       |
| news        | -- | 4            | 0    | 0    |       | 1           | 0    | 0    |       |
| gleydson    | -- | 31032        | 0    | 0    |       | 6956        | 0    | 0    |       |
| testuser    | -- | 16           | 0    | 0    |       | 4           | 0    | 0    |       |
| anotheruser | -- | 16           | 0    | 0    |       | 4           | 0    | 0    |       |
| nobody      | -- | 2344         | 0    | 0    |       | 2           | 0    | 0    |       |

```
*** Report for user quotas on device /dev/hda5
```

```
Block grace time: 2days; Inode grace time: 7days
```

| User     |    | Block limits |        |        |       | File limits |      |      |       |
|----------|----|--------------|--------|--------|-------|-------------|------|------|-------|
|          |    | used         | soft   | hard   | grace | used        | soft | hard | grace |
| root     | -- | 16052        | 0      | 0      | none  | 6443        | 0    | 0    | none  |
| gleydson | +- | 4944         | 500100 | 600000 | none  | 10868       | 0    | 0    |       |

```
*** Report for group quotas on device /dev/hda5
```

```
Block grace time: 7days; Inode grace time: 7days
```

| Group    |    | Block limits |      |      |       | File limits |      |      |       |
|----------|----|--------------|------|------|-------|-------------|------|------|-------|
|          |    | used         | soft | hard | grace | used        | soft | hard | grace |
| root     | -- | 20308        | 0    | 0    | none  | 636         | 0    | 0    | none  |
| src      | -- | 11404        | 0    | 0    |       | 660         | 0    | 0    |       |
| users    | -- | 1756         | 0    | 0    |       | 6561        | 0    | 0    |       |
| gleydson | -- | 3452         | 0    | 0    |       | 9307        | 0    | 0    |       |

Um sinal de "+-" no segundo campo indica quota ultrapassada ou no espaço em disco, "-+" em número de arquivos e "++" em ambos. Como vimos acima, o este comando também lista o número de arquivos e bytes pertencentes a cada usuário na partição (mesmo não sendo monitorado pelas restrições de quota), isto ajuda a monitorar ações suspeitas com a excedência de espaço em disco de determinados usuários/grupos do sistema. Um exemplo é alguém que esteja fora da quota e abusando de seu usuário/grupo para uso excessivo de espaço em disco sem seu conhecimento.

**OBS:** Este utilitário pode ser executado por qualquer usuário no sistema e mostrar o uso de quotas de usuários/grupos que não deveria ter acesso. É recomendado deve ter permissões de leitura/gravação somente para o usuário root e sem permissões para grupo/outros usuários.

## Avisando usuários sobre o estouro de quota

Avisos sobre quota ultrapassada podem ser enviadas automaticamente a todos os usuários pelo utilitário **warnquota**. Ele poderá ser executado periodicamente através do **cron** (por padrão isto é feito diariamente

na distribuição **Debian** pelo script `/etc/cron.daily/quota`). Dados adicionais sobre o envio das mensagens devem ser especificados no arquivo `/etc/warnquota.conf` seu formato é o seguinte:

```
Programa usado para enviar as mensagens
MAIL_CMD = "/usr/sbin/sendmail -t"
Campo de origem da mensagem
FROM = "root@localhost"
but they don't have to be:
SUBJECT = Quota excedida
CC_TO = "root@localhost"
SUPPORT = "root@localhost"
PHONE = "5555-2525"
#
```

O e-mail é enviado aos usuários (e usuários que pertencem a grupos com a quota excedida) com o seguinte formato:

```
From: root@localhost
To: gleydson@debian.gms.com.br
Cc: root@localhost
Reply-To: root@localhost
Subject: Quota Excedida
Date: Sat, 22 Sep 2001 14:27:38 -0400
```

Hi,

We noticed that you are in violation with the `quotasystem` used on this system. We have found the following violations:

| Filesystem | Block limits |        |        |       | File limits |      |      |       |
|------------|--------------|--------|--------|-------|-------------|------|------|-------|
|            | used         | soft   | hard   | grace | used        | soft | hard | grace |
| /dev/hda5  | +- 504944    | 500100 | 600000 | none  | 10868       | 0    | 0    |       |

We hope that you will cleanup before your grace period expires.

Basically, this means that the system thinks you are using more disk space on the above partition(s) than you are allowed. If you do not delete files and get below your quota before the grace period expires, the system will prevent you from creating new files.

For additional assistance, please contact us at `root@localhost` or via phone at 5555-2525.

## Suporte a senhas ocultas

Veja "Shadow Passwords".

## Suporte a senhas md5

Veja "Senhas MD5".

## Restrições no hardware do sistema

As restrições descritas aqui são úteis para diminuir as chances de um ataque por acesso físico ser realizado com sucesso no sistema que desejamos proteger.

Ter um sistema totalmente seguro é praticamente impossível, mas existem diversas maneiras de se dificultar as coisas.

### BIOS do sistema

Algumas restrições podem ser configuradas na para diminuir as chances de se obter acesso `root` (usando métodos conhecidos de recuperação via disquete/CD inicializável) ou simplesmente aumentar nossa confiança no sistema:

- Coloque uma senha para entrada no Setup da máquina, compartilhe esta senha **somente** com as pessoas que tem poder de `root` (ou seja, pessoal de confiança que administra a máquina).
- Mude a seqüência de partida para somente sua unidade de disco rígido que contém o sistema operacional. As BIOS trazem convenções de DOS para especificar o método de partida, então *Only C* quer dizer somente o primeiro disco rígido, *SCSI* tentar dispositivos SCSI primeiro, etc. Isso pode variar de acordo com o modelo de sua BIOS.

Com os dois ítems acima qualquer um ficará impedido de inicializar o sistema a partir de um disco de recuperação ou entrar no Setup para modificar a ordem de procura do sistema operacional para dar a partida via disquetes.

### Retirada da unidade de disquetes

Como não é seguro confiar nas restrições de senha da BIOS (qualquer um com conhecimentos de hardware e acesso físico a máquina pode abrir o gabinete e dar um curto na bateria que mantém os dados na CMOS ou aterrar o pino de sinal da CMOS), a retirada da unidade de disquetes é recomendada, isso dificultará bastante as coisas.

### Placas de rede com eprom de boot

Evite a utilização de placas de rede com recursos de boot via EPROM no servidor, um servidor dhcp/bootp/tftp poderá ser configurado sem problemas por um cracker na rede (caso a BIOS esteja com a ordem inadequada de procura de discos) e o ataque se dar com mais "sofisticação" e rapidez.

### Protegendo o LILO

A opção *passwd=senha* e *restricted* poderão ser usadas na seção da imagem que desejamos proteger. Respectivamente pedem uma senha para a inicialização do sistema e caso argumentos como *root=single* sejam usados para conseguir acesso `root` sem fornecer senha.

E deixe somente as permissões de acesso ao usuário `root` (caso contrário sua senha poderá ser vista por qualquer usuário) e modifique os atributos deste arquivo para imutável para que nem mesmo o `root` possa modificá-lo: `chattr +i /etc/lilo.conf`.

### Disco rígido

O disco rígido do servidor poderá se retirado como alternativa para se ter acesso aos dados armazenados. Isto poderá ser dificultado com o uso de lacres de disco ou outras maneiras de dificultar mais esta tarefa

(mais parafusos, armazenamento em partes de difícil manipulação do HD, etc) qualquer coisa que possa lhe fazer ganhar tempo e despertar suspeitas para evitar o sucesso desta alternativa (ousada).

Dados importantes ou confidenciais poderão ser armazenados em um sistema de arquivos criptografados e serem montados somente pelos administradores que possuem acesso físico ao sistema. O algoritmo *Serpent* é muito forte na proteção de dados além de possuir um ótimo desempenho. Patches de criptografia poderão ser aplicados no kernel para ativação deste recurso (veja “Sistemas de arquivos criptográfico”) para detalhes.

Sensores podem ser ligados na carcaça do HD como forma de disparar um pequeno alarme embutido no gabinete do servidor, se você gosta de eletrônica poderá montar um destes facilmente para chamar a atenção alimentado por fonte/baterias em um circuito de emergência, e poderá acomodar sua caixa em uma segunda "carcaça de fonte" apenas para desviar suspeitas. Um circuito interno de câmeras também é uma boa alternativa para monitorar a movimentação.

Esquemas de segurança dependendo do porte da organização e dos dados que se desejam proteger deverão ser elaborados e postos em prática. Todos os métodos imagináveis deverão ser considerados de acordo com as possibilidades do ambiente.

---

# Capítulo 20. Introdução ao uso de criptografia para transmissão/armazenamento de dados

Este capítulo explica como dados transmitidos em uma rede pode ser capturados, isto ajudará a entender a vulnerabilidade de serviços comuns que não utilizam criptografia para a transmissão de dados e alternativas/programas equivalentes que fazem transmissão de dados usando métodos criptográficos para deixar a mensagem somente legível para origem e destino.

## Introdução

Quando enviamos um tráfego de nossa máquina para outra (e-mails, mensagens de ICQ, navegação, ftp, etc) os dados passam por várias máquinas até atingir o seu destino (isto se chama roteamento). Se algum cracker instalou algum capturador de pacotes (sniffer) em alguma das máquinas de nossa rota os dados poderão facilmente visualizados.

Crackers normalmente configuram estes programas a procura de campos como "passwd" e outras expressões que sejam úteis para acesso ao seu sistema ou espionagem. Quem gosta de ter sua privacidade violada? A internet definitivamente é uma rede insegura e nem todos os administradores de servidores são responsáveis o suficiente para fazer uma configuração restrita para evitar acesso de pessoas mal intencionadas.

Este capítulo mostra (na prática) como um sniffer funciona para captura de pacotes, isto ajudará a entender como serviços que enviam seus dados em forma texto plano são vulneráveis a isto e alternativas para transmissão segura de dados. Este capítulo tem a intenção de mostrar alternativas seguras de proteção dos dados que trafegam em sua rede e a segurança de suas instalações.

## Sniffer

O sniffer (farejador) é um programa que monitoram/registram a passagem de dados entre as interfaces de rede instaladas no computador. Os dados coletados por sniffers são usados para obtenção de detalhes úteis para solução de problemas em rede (quando usado com boas intenções pelo administrador do sistema) ou para ataques ao sistema (quando usado pelo cracker para obter nomes/senhas e outros detalhes úteis para espionagem).

Os sniffers mais conhecidos para sistemas **Linux** são **tcpdump**, **ethereal**. Este último apresenta uma interface gráfica GTK para fácil operação em máquinas que executam o servidor X. Para explicar o funcionamento de um sniffer, vou assumir o **ethereal** instalado (ele não requer modificações no sistema além de ser fácil de executar e fazer pesquisa de expressões específicas). Instale o **ethereal** com o comando `apt-get install ethereal`.

Agora vamos a prática para entender como o sniffer funciona e a importância da criptografia de dados (só assim mesmo, não dá para entender falando muita teoria :-):

1. Conecte-se a Internet
2. Execute o **ethereal** como usuário `root`.
3. Pressione `CTRL+K` para abrir a tela de captura de pacotes. Em Interface selecione sua interface de internet. Nesta tela clique no botão "FILE" e coloque um nome de arquivo que a captura será gravada.

Opcionalmente marque a opção "Update list of packets in real time" para monitorar a passagem de pacotes em tempo real.

4. Clique em "OK". A captura de pacotes será iniciada
5. Conecte-se a um site ftp qualquer (digamos ftp.debian.org.br). Entre com o usuário "anonymous" e senha "minhasenha@segura.com.br"
6. Finalize a captura de pacotes clicando no botão "STOP"

Agora vá em "File"/"Open" e abra o arquivo capturado. Ele está no formato usado pelo sniffer **tcpdump** como padrão. Procure no campo "INFO" a linha "Request: USER anonymous", logo abaixo você verá a senha digitada pelo usuário. Entendeu agora a importância da criptografia na transferência segura de dados? não só o nome/senha pode ser capturado mas toda a seções feitas pelo usuário. Scanners como o **tcpdump** e **ethereal** são flexivelmente configuráveis para procurar por dados específicos nas conexões e salva-los para posterior recuperação.

## Detectando a presença de sniffers

Uma característica comum de sniffers é mudar o modo de operação das interfaces monitoradas para o "Modo Promíscuo" com o objetivo de analisar todo o tráfego que passa por aquele segmento de rede (mesmo não sendo destinados para aquela máquina).

A entrada/saída de interfaces no modo promíscuo é monitorada nos logs do sistema:

```
Sep 25 16:53:37 myserver kernel: device eth0 left promiscuous mode
Sep 25 16:53:56 myserver kernel: device eth0 entered promiscuous mode
Sep 25 16:54:18 myserver kernel: device eth0 left promiscuous mode
Sep 25 16:54:31 myserver kernel: device eth0 entered promiscuous mode
```

O **logcheck** monitora estas atividades e classificam esta mensagem como prioridade "Violação" (dependendo da configuração dos seus filtros em `/etc/logcheck`. Veja "logcheck" para detalhes sobre este programa.

**OBS:** A utilização de switches dificulta a captura de pacotes em redes distribuídas porque somente os dados destinados a máquina onde o sniffer está instalado poderão ser capturados.

## Alternativas seguras a serviços sem criptografia

### http

O uso de alternativas seguras é indispensável em servidores que servem páginas de comércio eletrônico, banco de dados, sistemas bancários, administração via web ou que tenham dados que oferecem risco, se capturados.

Existem duas alternativas: instalar o servidor Apache-ssl (pacote ou adicionar o módulo `mod-ssl` na instalação padrão do **Apache**. Esta segunda é a preferida por ser mais rápida e simples de se administrar, por usar o servidor Web **Apache** padrão e sua configuração. Veja "Uso de criptografia SSL" para detalhes de como configurar um servidor Web para transmissão de dados criptografados.

## Transmissão segura de e-mails

A codificação padrão usada para o envio de mensagens em muitos clientes de e-mail é o MIME/base64. Isto não oferece muita segurança porque os dados podem ser facilmente descriptografados se pegos por sniffers (veja “Sniffer”) ou abertos por administradores não confiáveis no diretório de spool do servidor.

Existem uma diversidade de servidores SMTP, POP, IMAP do **Linux** que já implementam o protocolo de autenticação *SSL/TLS*, exigindo login/senha para o envio/recepção de mensagens, cabeçalhos de autenticação (aumentando um pouco mais a confiança sobre quem enviou a mensagem). Em especial, a autenticação é útil quando desejamos abrir nossas contas de e-mail para a Internet, por algum motivo, e não queremos que outros façam relay sem nossa autorização.

Outra forma de garantir a segurança da mensagem/arquivos através do correio eletrônico é usando o PGP (veja “Usando **gpg** para criptografia de arquivos”) em conjunto com um MUA (Mail User Agent - cliente de e-mails) que suporte o envio de mensagens criptografadas/assinadas usando PGP. A vantagem do GPG em cima da autenticação SSL é que você tem garantidas da autenticidade da mensagem e você pode verificar sua integridade. Os dois programas mais usados em sistemas **Unix** são o **mutt** e o **sylpheed**. O **mutt** é um MUA para modo texto e o **sylpheed** para modo gráfico. Ambos são muito flexíveis, permitem uma grande variedade de configurações, personalizações, possuem agenda de endereços e gerenciam diversas contas de e-mails em um só programa.

Para encriptar/assinar uma mensagem no **mutt** escreva/responda seu e-mail normalmente, quando aparecer a tela onde você tecla "y" para enviar a mensagem, tecla "p" e selecione uma das opções para criptografar/assinar uma mensagem.

Para fazer a mesma operação no **sylpheed**, escreva/responda seu e-mail normalmente e clique no menu "Mensagem" e marque "assinar", "criptografar" ou ambos. A chave pública deverá estar disponível para tal operação (veja “Adicionando chaves públicas ao seu chaveiro pessoal” e “Extraindo sua chave pública do chaveiro”).

## Servidor pop3

A alternativa mais segura é a utilização do protocolo **IMAP** com suporte a ssl. Nem todos os clientes de e-mail suportam este protocolo.

## Transferência de arquivos

Ao invés do **ftp**, use o **scp** ou o **sftp** para transferência segura de arquivos. Veja “scp” e “sftp”. Uma outra alternativa é a configuração de uma VPN entre redes para garantir não só a transferência de arquivos, mas uma seção em cima de um tunel seguro entre duas pontas.

## login remoto

Ao invés do uso do **rlogin**, **telnet** e **rsh** utilize o **ssh** (veja “ssh”) ou o telnet com suporte a ssl (veja “Instalação”).

## Bate papo via IRC

O programa **SILC** (Secure Internet Live Conference) realiza a criptografia de dados durante o bate papo entre diversos usuários conectados via rede.

## Transmissão de mensagens via ICQ

O protocolo ICQ trabalha de forma plana para transmissão de suas mensagens, inclusive as senhas. Clientes anteriores ainda usavam o UDP (até a versão 7) para envio de mensagens, piorando um pouco mais a situação e deixando o cliente mais vulnerável a falsificações de pacotes. Outro ponto fraco é que se alguma coisa acontecer com os pacotes UDP, eles serão simplesmente descartados perdendo a mensagem.

Ao invés do ICQ, você poderá usar algum cliente do protocolo **Jabber** (como o **gaim**, **gaber** ou **gossip**) ou o LICQ mais atual com suporte a ssl compilado. O problema do LICQ com ssh, é que as duas pontas deverão ter este suporte compilado e funcionando.

## Sistemas de arquivos criptográfico

Esta é uma forma excelente para armazenamento seguro de seus dados, pois estarão criptografados e serão somente acessados após fornecer uma senha que só você conhece. O sistema usado é a montagem de um arquivo comum como um sistema de arquivos via loopback (veja ???) você pode escolher um nome de arquivo discreto para dificultar sua localização (use a imaginação) e poderá ser armazenado até mesmo em partições não-ext2. Siga estes passos para criar seu sistema de arquivos criptografado (baseado no Loopback-Encrypted-Filesystem):

Suporte no kernel

Baixe o patch criptográfico de <ftp://ftp.kernel.org/pub/linux/kernel/crypto> de acordo com a sua versão do kernel e aplique os patches. Este suporte não pode ser incluído nativamente no kernel devido a restrições de uso e importação de criptografia impostas pelos EUA e outros países, com este suporte embutido o kernel não poderia ser distribuído livremente.

Se o patch para seu kernel não existir, pegue a versão anterior mais próxima (se não existir o patch para seu kernel 2.2.19, pegue a versão 2.2.18 do patch internacional). Isto certamente funcionará.

Opções de compilação do kernel

Na seção `Crypto Support` ative `Crypto Ciphers` e ative o suporte aos ciphers `Twofish`, `blowfish`, `cast128`, e `serpent` (estes são distribuídos livremente e sem restrições). Todos possuem cifragem de 128 bits, exceto o `blowfish` que é 64 bits. Também é recomendado ativar os módulos em `Digest algorithms`.

Na seção `Block Devices`: ative o suporte a `loopback` (necessário para montar arquivos como dispositivos de bloco) e `Use relative block numbers as basis for transfer functions` (isto permite que um backup do sistema de arquivos criptografado seja restaurado corretamente em outros blocos ao invés dos originais). Ative também o suporte para `General encryption support` e o suporte aos cyphers `cast128` e `twofish`.

Não ative as opções de criptografia para a seção "Networking" (a não ser que saiba o que está fazendo). Recompile e instale seu kernel (veja ???).

Crie um arquivo usando os números aleatórios de `/dev/urandom`:

```
dd if=/dev/urandom of=/pub/swap-fs bs=1M count=15
```

|                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                              | <p>Será criado um arquivo chamado <code>swap-fs</code> (um arquivo de troca tem características que ajudam a esconder um sistema de arquivos criptografado que é o tamanho e não poderá ser montado pelo usuário comum, evitando desconfianças).</p> <p>O processo de criação deste arquivo é lento, em média de 1MB a cada 10 segundos em um Pentium MMX.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Monte o arquivo como um sistema de arquivos loop             | <pre>losetup -e twofish /dev/loop0 /pub/swap-fs</pre> <p>O algoritmo de criptografia é selecionado pela opção <code>-e</code>. Algoritmos recomendados são o <code>serpent</code> e <code>twofish</code> (ambos possuem cifragem de 128 bits), sendo o <code>serpent</code> o preferido. O gerenciamento do sistema loop encriptado é feito através do módulo <code>loop_gen</code>.</p> <p>Quando é executado pela primeira vez, será lhe pedida uma senha que será usada para montagens futuras de seu sistema de arquivos. Digite-a com atenção pois ela será lhe pedida apenas uma vez. Para desativar o sistema de arquivos loop, execute o comando:</p> <pre>losetup -d /dev/loop0</pre> <p><b>OBS:</b> Se errou a senha será necessário desmontar, apagar o arquivo criado e repetir o procedimento.</p> |
| Crie um sistema de arquivos ext2 para armazenamento de dados | <pre>mkfs -t ext2 /dev/loop0 ou mkfs.ext2 /dev/loop0</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Monte o sistema de arquivos                                  | <p>Crie um diretório que será usado para montagem do seu sistema de arquivos, se preferir monta-lo dentro de seu diretório pessoal para armazenar seus arquivos, crie um diretório com as permissões "0700".</p> <pre>mount /pub/swap-fs /pub/criptofs -t ext2 -o loop</pre> <p>Agora poderá gravar seus arquivos dentro deste diretório normalmente como qualquer outro. O comando <code>df -hT</code> listará a partição loop como uma partição do tipo <code>ext2</code> comum.</p>                                                                                                                                                                                                                                                                                                                          |
| Desmontando/Protegendo os dados                              | <p>Após usar o sistema de arquivos criptográfico, desmonte-o e desative o dispositivo loopback:</p> <pre>umount /pub/criptofs<br/>losetup -d /dev/loop0</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Remontando o sistema de arquivos criptografado               | <p>Execute novamente os comandos:</p> <pre>losetup -e twofish /dev/loop0 /pub/swap-fs<br/>mount /pub/swap-fs /pub/criptofs -t ext2 -o loop</pre> <p>Será pedida a senha que escolheu e seu sistema de arquivos será montado em <code>/pub/swap-fs</code>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Com este sistema, seus dados estarão protegidos mesmo do usuário `root`.

## Usando pgp (gpg)para criptografia de arquivos

O **gpg** (GNU pgp, versão livre da ferramenta pgp) permite encriptar dados, assim somente o destinatário terá acesso aos dados, adicionalmente poderá verificar se a origem dos dados é confiável (através da assinatura de arquivos). O sistema PGP se baseia no conceito de chave *pública* e *privada*: Sua chave *pública* é distribuída para as pessoas que deseja trocar dados/mensagens e a chave *privada* fica em sua máquina (ela não pode ser distribuída). As chaves públicas e privadas são armazenadas nos arquivos `pubring.gpg` e `secring.gpg` respectivamente, dentro do subdiretório `~/ .gnupg`. Veja “Criando um par de chaves pública/privada” para criar este par de chaves.

Os dados que recebe de outra pessoa são criptografados usando sua chave pública e somente você (de posse da chave privada) poderá descriptar os dados. Quando assina um arquivo usando o pgp, ele faz isto usando sua chave privada, o destinatário de posse da chave pública poderá então confirmar que a origem dos dados é confiável.

O **gpg** vem largamente sendo usado para transmissão segura de dados via internet. Muitos programas de e-mails como o **mutt** e **sylpheed** incluem o suporte a pgp embutido para envio de mensagens assinadas/ encriptadas (MIME não tem uma codificação segura e não garante que a mensagem vem de quem realmente diz ser). Um servidor de e-mail no **Linux** configurado como as mesmas configurações/endereços do provedor da vítima pode enganar com sucesso um usuário passando-se por outro.

## Instalando o PGP

```
apt-get install gnupg
```

Após instalar o , execute o comando **gpg** para criar o diretório `~/ .gnupg` que armazenará as chaves pública e privada.

## Criando um par de chaves pública/privada

Para gerar um par de chaves pessoais use o comando `gpg --gen-key`. Ele executará os seguintes passos:

1. Chave criptográfica - Selecione *DSA e ELGamal* a não ser que tenha necessidades específicas.
2. Tamanho da chave - 1024 bits traz uma boa combinação de proteção/velocidade.
3. Validade da chave - 0 a chave não expira. Um número positivo tem o valor de dias, que pode ser seguido das letras w (semanas), m (meses) ou y (anos). Por exemplo, "7m", "2y", "60".

Após a validade, a chave será considerada inválida.

4. Nome de usuário - Nome para identificar a chave
5. E-mail - E-mail do dono da chave
6. comentário - Uma descrição sobre a chave do usuário.
7. Confirmação - Tecla "O" para confirmar os dados ou uma das outras letras para modificar os dados de sua chave.
8. Digite a FraseSenha - Senha que irá identificá-lo(a) como proprietário da chave privada. É chamada de FraseSenha pois pode conter espaços e não há limite de caracteres. Para alterá-la posteriormente, siga as instruções em “Mudando sua FraseSenha”.
9. Confirme e aguarde a geração da chave pública/privada.

## Encriptando dados

Use o comando `gpg -e arquivo` faz a encriptação de dados:

```
gpg -e arquivo.txt
```

Será pedida a identificação de usuário, digite o nome que usou para criar a chave. O arquivo criado será encriptado usando a chave pública do usuário (`~/ .gnupg/pubring.gpg`) e terá a extensão `.gpg` adicionada (`arquivo.txt.gpg`). Além de criptografado, este arquivo é compactado (recomendável para grande quantidade de textos). A opção `-a` é usada para criar um arquivo criptografado com saída ASCII 7 bits:

```
gpg -e -a arquivo.txt
```

O arquivo gerado terá a extensão `.asc` acrescentada (`arquivo.txt.asc`) e não será compactado. A opção `-a` é muito usada para o envio de e-mails.

Para criptografar o arquivo para ser enviado a outro usuário, você deverá ter a chave pública do usuário cadastrado no seu chaveiro (veja “Adicionando chaves públicas ao seu chaveiro pessoal”) e especificar a opção `-r` seguida do nome/e-mail/ID da chave pública:

```
gpg -r kov -e arquivo.txt
```

O exemplo acima utiliza a chave pública de `kov` para encriptar o arquivo `arquivo.txt` (somente ele poderá decriptar a mensagem usando sua chave privada).

**OBS:** É recomendável especificar o nome de arquivo sempre como último argumento.

## Decriptando dados com o gpg

Agora vamos fazer a operação reversa da acima, a opção `-d` é usada para decriptar os dados usando a chave privada:

```
gpg -d arquivo.txt.asc >arquivo.txt
gpg -d arquivo.txt.gpg >arquivo.txt
```

Descriptografa os arquivos `arquivo.txt.asc` e `arquivo.txt.gpg` recuperando seu conteúdo original. A sua "FraseSenha" será pedida para descriptografar os dados usando a chave privada (`~/ .gnupg/secring.gpg`).

## Assinando arquivos

Assinar um arquivo é garantir que você é a pessoa que realmente enviou aquele arquivo. Use a opção `-s` para assinar arquivos usando sua chave privada:

```
gpg -s arquivo.txt
```

A "FraseSenha" será pedida para assinar os dados usando sua chave privada. Será gerado um arquivo `arquivo.txt.gpg` (assinado e compactado). Adicionalmente a opção `--clearsign` poderá ser usada

para fazer uma assinatura em um texto plano, este é um recurso muito usado por programas de e-mails com suporte ao gpg:

```
gpg -s --clearsign arquivo.txt
```

Será criado um arquivo chamado `arquivo.txt.asc` contendo o arquivo assinado e sem compactação.

## Checando assinaturas

A checagem de assinatura consiste em verificar que quem nos enviou o arquivo é realmente quem diz ser e se os dados foram de alguma forma alterados. Você deverá ter a chave pública do usuário no seu chaveiro para fazer esta checagem (veja “Adicionando chaves públicas ao seu chaveiro pessoal”). Para verificar os dados assinados acima usamos a opção `--verify`:

```
gpg --verify arquivo.txt.asc
```

Se a saída for "Assinatura Correta", significa que a origem do arquivo é segura e que ele não foi de qualquer forma modificado.

```
gpg --verify arquivo.txt.gpg
```

Se a saída for "Assinatura INCORRETA" significa que ou o usuário que enviou o arquivo não confere ou o arquivo enviado foi de alguma forma modificado.

## Extraindo sua chave pública do chaveiro

Sua chave pública deve ser distribuída a outros usuários para que possam enviar dados criptografados ou checar a autenticidade de seus arquivos. Para exportar sua chave pública em um arquivo que será distribuído a outras pessoas ou servidores de chaves na Internet, use a opção `--export`:

```
gpg --export -a usuario >chave-pub.txt
```

Ao invés do nome do usuário, poderá ser usado seu e-mail, ID da chave, etc. A opção `-a` permite que os dados sejam gerados usando bits ASCII 7.

## Adicionando chaves públicas ao seu chaveiro pessoal

Isto é necessário para o envio de dados criptografados e checagem de assinatura do usuário, use a opção `--import`:

```
gpg --import chave-pub-usuario.txt
```

Assumindo que o arquivo `chave-pub-usuario.txt` contém a chave pública do usuário criada em “Extraindo sua chave pública do chaveiro”. O **gpg** detecta chaves públicas dentro de textos e faz a extração corretamente. Minha chave pública pode ser encontrada em “Chave Pública PGP” ou <http://pgp.ai.mit.edu>.

## Listando chaves de seu chaveiro

Use o comando `gpg --list-keys` para listar as chaves pública do seu chaveiro. O comando `gpg --list-secret-keys` lista suas chaves privadas.

## Apagando chaves de seu chaveiro

Quando uma chave pública é modificada ou por qualquer outro motivo deseja retirá-la do seu chaveiro público, utilize a opção `--delete-key`:

```
gpg --delete-key usuario
```

Pode ser especificado o nome de usuário, e-mail IDchave ou qualquer outro detalhe que confira com a chave pública do usuário. Será pedida a confirmação para excluir a chave pública.

**OBS:** A chave privada pode ser excluída com a opção `--delete-secret-key`. Utilize-a com o máximo de atenção para excluir chaves secretas que não utiliza (caso use mais de uma), a exclusão acidental de sua chave secreta significa é como perder a chave de um cofre de banco: você não poderá descriptografar os arquivos enviados a você e não poderá enviar arquivos assinados.

Mesmo assim se isto acontecer acidentalmente, você poderá recuperar o último backup da chave privada em `~/ .gnupg/secring.gpg`.

## Mudando sua FraseSenha

Execute o comando `gpg --edit-key usuário`, quando o programa entrar em modo de comandos, digite `passwd`. Será lhe pedida a "Frase Senha" atual e a nova "Frase Senha". Digite "save" para sair e salvar as alterações ou "quit" para sair e abandonar o que foi feito.

O `gpg --edit-key` permite gerenciar diversos aspectos de suas chaves é interessante explorá-lo digitando "?" para exibir todas as opções disponíveis.

## Assinando uma chave digital

A assinatura de chaves é um meio de criar laços de confiança entre usuários PGP. Assinar uma chave de alguém é algo sério, você deve ter noção do que isto significa e das conseqüências que isto pode trazer antes de sair assinando chaves de qualquer um.

O próprio teste para desenvolvedor da distribuição **Debian** requer como primeiro passo a identificação do candidato, caso sua chave `gpg` seja assinada por algum desenvolvedor desta distribuição, imediatamente o teste de identificação é completado. A partir disso você deve ter uma noção básica do que isto significa. Para assinar uma chave siga os seguintes passos:

1. Importe a chave pública do usuário (veja "Adicionando chaves públicas ao seu chaveiro pessoal").
2. Execute o comando `gpg --edit-key usuario` (onde *usuario* é o nome do usuário/e-mail/IDchave da chave pública importada).
3. Digite `list`, e selecione a chave pública (`pub`) do usuário com o comando `uid [numero_chave]`. Para assinar todas as chaves públicas do usuário, não selecione qualquer chave com o comando `uid`.
4. Para assinar a chave pública do usuário digite `sign`, será perguntado se deseja realmente assinar a chave do usuário e então pedida a "FraseSenha" de sua chave privada.
5. Digite "list", repare que existe um campo chamado `trust: n/q` no lado direito. O primeiro parâmetro do "trust" indica o valor de confiança do dono e o segundo (após a /) o valor de confiança calculado automaticamente na chave. As seguintes possuem o seguinte significado:
  - - - Nenhum dono encontrado/confiança não calculada.

- e - Chave expirada/falha na checagem de confiança.
- q - Quando não conhece o usuário.
- n - Quando não confia no usuário (é o padrão).
- m - Pouca confiança no usuário.
- f - Totalmente confiável.
- u - Indiscutivelmente confiável. Somente usado para especificar a chave pública do próprio usuário.

O valor de confiança da chave pode ser modificado com o comando `trust` e selecionando uma das opções de confiança. Os valores de confiança para a chave pública pessoal é `-/u` (não é necessário calcular a confiança/indiscutivelmente confiável).

## Listando assinaturas digitais

Execute o comando `gpg --list-sigs` para listas todas as assinaturas existentes no seu chaveiro. Opcionalmente pode ser especificado um parâmetro para fazer referência a assinatura de um usuário: `gpg --list-sigs usuario`.

O comando `gpg --check-sigs` adicionalmente faz a checagem de assinaturas.

## Recomendações para a assinatura de chaves gpg

Este texto foi divulgado por uma pessoa que pediu para permanecer anônima na lista `<debian-user-portuguese@lists.debian.org>` explicando os procedimentos de segurança para a troca de chaves públicas individuais e em grupo de usuários. Ele é um pouco longo mas a pessoa é especializada no assunto, e seu foco é a segurança na troca de chaves e o que isto significa. Após consulta ao autor do texto, o texto foi reproduzido na íntegra, mantendo os padrões de formatação da mensagem.

Trocando assinaturas de chaves digitais

Direitos de republicação cedidos ao domínio público, contanto que o texto seja reproduzido em sua íntegra, sem modificações de quaisquer espécie, e incluindo o título e nome do autor.

1. Assinaturas digitais
2. Chaves digitais e a teia de confiança
3. Trocando assinaturas de chaves digitais com um grupo de pessoas

1. Assinaturas digitais

Uma assinatura digital é um número de tamanho razoável (costuma ter de 128 a 160 bits) que representa um bloco bem maior de informação, como um e-mail.

Pense numa assinatura como se ela fosse uma versão super-comprimida de um texto. Se você muda alguma coisa (por menor que seja) no texto que uma assinatura "assina", essa assinatura se torna inválida: ela não mais representa aquele texto.

Existe uma relação direta entre uma assinatura e informação que ela assina. Se uma das duas for modificada, elas passam a não mais "combinar" uma com a outra. Um programa de computador pode detectar isso, e avisar que a assinatura é "inválida".

Os algoritmos mais usados para criar e verificar assinaturas digitais são o SHA-1, RIPEM160 e MD5. O MD5 não é considerado tão bom quanto os outros dois.

Assinaturas digitais também funcionam com arquivos "binários", ou seja: imagens, som, planilhas de cálculo... e chaves digitais.

## 2. Chaves digitais e a teia de confiança

Chaves digitais são fáceis de falsificar, você só precisa criar uma chave nova no nome de sicrano, por um endereço de e-mail novinho em folha daqueles que você consegue nesses webmail da vida, e pronto. Agora é só espalhar essa chave por aí que os bestas vão usá-la pensando que é de sicrano.

A menos que os "bestas" não sejam tão bestas assim, tenham lido o manual do seu software de criptografia, e saibam usar assinaturas e a teia de confiança para verificar se a tal chave é de sicrano mesmo.

Programas de criptografia (os bons, tipo PGP e GNUpg) usam um sistema de assinaturas nas chaves digitais para detectar e impedir esse tipo de problema: Ao usuário é dado o poder de "assinar" uma chave digital, dizendo "sim, eu tenho certeza que essa chave é de fulano, e que o e-mail de fulano é esse que está na chave".

Note bem as palavras "certeza", e "e-mail". Ao assinar uma chave digital, você está empenhando sua palavra de honra que o nome do dono de verdade daquela chave é o nome que está na chave, e que o endereço de e-mail daquela chave é da pessoa (o "nome") que também está na chave.

Se todo mundo fizer isso direitinho (ou seja, não sair assinando a chave de qualquer um, só porque a outra pessoa pediu por e-mail, ou numa sala de chat), cria-se a chamada teia de confiança.

Numa teia de confiança, você confia na palavra de honra dos outros para tentar verificar se uma chave digital é legítima, ou se é uma "pega-bobo".

Suponha que Marcelo tenha assinado a chave de Cláudia, e que Roberto, que conhece Marcelo pessoalmente e assinou a chave de Marcelo, queira falar com Cláudia.

Roberto sabe que Marcelo leu o manual do programa de criptografia, e que ele não é irresponsável. Assim, ele pode confiar na palavra de honra de Marcelo que aquela chave digital da Cláudia é da Cláudia mesmo, e usar a chave pra combinar um encontro com Cláudia.

Por outro lado, Roberto não conhece Cláudia (ainda), e não sabe que tipo de pessoa ela é. Assim, rapaz prevenido, ele não confia que Cláudia seja uma

pessoa responsável que verifica direitinho antes de assinar chaves.

Note que Roberto só confiou na assinatura de Marcelo porque, como ele já tinha assinado a chave de Marcelo, ele sabe que foi Marcelo mesmo quem assinou a chave de Cláudia.

Enrolado? Sim, é um pouco complicado, mas desenhe num papel as flechinhas de quem confia em quem, que você entende rapidinho como funciona.

O uso da assinatura feita por alguém cuja chave você assinou, para validar a chave digital de um terceiro, é um exemplo de uma pequena teia de confiança.

### 3. Trocando assinaturas de chaves digitais com um grupo de pessoas

Lembre-se: ao assinar uma chave digital, você está empenhando sua palavra de honra que toda a informação que você assinou naquela chave é verdadeira até onde você pode verificar, e que você tentou verificar direitinho.

Pense nisso como um juramento: "Eu juro, em nome da minha reputação profissional e pessoal, que o nome e endereços de e-mail nessa chave são realmente verdadeiros até onde posso verificar, e que fiz uma tentativa real e razoável de verificar essa informação."

Sim, é sério desse jeito mesmo. Você pode ficar muito "queimado" em certos círculos se você assinar uma chave falsa, pensando que é verdadeira: a sua assinatura mal-verificada pode vir a prejudicar outros que confiaram em você.

Bom, já que o assunto é sério, como juntar um grupo de pessoas numa sala, e trocar assinaturas de chaves entre si? Particularmente se são pessoas que você nunca viu antes? Siga o protocolo abaixo, passo a passo, e sem pular ou violar nenhum dos passos.

- 1 - Reúna todos em uma sala, ou outro local não tumultuado, pressa e bagunça são inimigas da segurança.
- 2 - Cada um dos presentes deve, então, ir de um em um e:
  - 2.1 - Apresentar-se, mostrando calmamente documentação original (nada de fotocópia) comprovando sua identidade. RG, CPF, passaporte, certidão de nascimento ou casamento, carteira de motorista, cartão de crédito são todos bons exemplos. Só o RG sozinho não é -- tem muito RG falsificado por aí -- mas o RG junto com o cartão de banco já seria suficiente. Se nenhum documento tiver foto, também não é o bastante.

\* Se alguém pedir o documento na mão, para verificar direitinho, não leve pro lado pessoal. Deixe a pessoa verificar até estar satisfeita (mas não descuide do documento). Isso só significa que ela leva muito a sério a responsabilidade de assinar chaves.

- 2.2 - Entregar um papel com as informações da chave: Nome (QUE OBRIGATORIAMENTE PRECISA SER O MESMO NOME CONSTANTE NOS DOCUMENTOS APRESENTADOS EM 2.1), e-mail, número da chave (keyID), e fingerprint da chave (assinatura digital da chave)

RECIPIENTE DO PAPEL: Se você achar que os documentos que te apresentaram não são prova suficiente, talvez porque o nome não bate com o da chave, ou porque uma foto nos documentos não está parecida com quem mostrou os documentos, marque discretamente no papel, porque você NÃO deve assinar essa chave. Se achar que o outro vai engrossar, não diga para ele que não vai assinar a chave dele.

- 3 - Pronto. Podem ir embora, porque o resto dos passos deve ser feito com calma, em casa. Lembre-se que você não vai estar efetuando nenhum julgamento moral a respeito de quem você assinar a chave. Você só irá afirmar que a chave de sicrano é realmente aquela, e mais nada.

- 4 - Para cada uma das chaves que você marcou no papel que "posso assinar":

- 4.1 - Peça para o seu programa de criptografia mostrar a chave e sua assinatura (fingerprint).

SE: O nome no papel for exatamente igual ao nome na chave (user ID/UID da chave). E: A assinatura no papel for exatamente igual à assinatura na chave (fingerprint). ENTÃO: Vá para o passo 4.3.

- 4.2 - As informações não bateram, por isso você não deve assinar a chave. Se quiser, envie um e-mail avisando que não poderá assinar a chave. Não aceite tentativas de retificação por e-mail ou telefone. Um outro encontro face-à-face, refazendo todos os passos 2.1 e 2.2 é o único jeito de retificar o problema.

- 4.3 - As informações bateram, o que garante que o \*nome\* está correto. Agora é preciso ter certeza do endereço de e-mail. Para isso, envie uma e-mail \*CIFRADA\* pela chave que você está testando, para o endereço de e-mail constante na chave. Nessa e-mail, coloque uma palavra secreta qualquer e peça para o destinatário te responder dizendo qual a palavra secreta que você escreveu. Use uma palavra diferente para cada chave que estiver testando, e anote no papel daquela chave qual palavra você usou.

- 4.4 - Se você receber a resposta contendo a palavra secreta correta, você pode assinar a chave. Caso contrário, não assine a chave -- o endereço de e-mail pode ser falso.

Comandos do gpg (GNUpg) correspondentes a cada passo:

- 2.2 - gpg --fingerprint <seu nome ou 0xSuaKEYID>  
(retorna as informações que devem estar no papel a ser

entregue no passo 2.2)

4.1 - `gpg --receive-key <0xKEYID>`  
(procura a chave especificada nos keyserver)  
`gpg --sign-key <0xKEYID>`  
(assina uma chave)

Assume-se que você sabe cifrar e decifrar mensagens. Caso não saiba, ainda não é hora de querer sair assinando chaves.

---

# Capítulo 21. Como obter ajuda

Dúvidas são comuns durante o aprendizado e uso do **Linux** e existem várias maneiras de se obter ajuda e encontrar a resposta para algum problema. O **GNU/Linux** é um sistema bem documentado, e ter um programa bem documentado é o princípio para seu sucesso junto a utilizadores e desenvolvedores, pois demonstra a dedicação de seu desenvolvedor em garantir boa usabilidade. Abaixo segue algumas formas úteis para encontrar a solução de sua dúvida, vale a pena conhece-las.

## Páginas de Manual

As *páginas de manual* acompanham quase todos os programas **GNU/Linux**. Elas trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento de opção.

É mais comum fazer a visualização de uma página de manual em modo texto, com rolagem vertical, em geral com a aparência de arquivos visualizados com os comandos **less** e **more**. Também documenta parâmetros usados em alguns arquivos de configuração.

A utilização da página de manual é simples, digite:

```
man [seção] [comando/arquivo]
```

*onde:*

*seção*                      É a seção de manual que será aberta, se omitido, mostra a *primeira* seção sobre o comando encontrada (em ordem crescente).

*comando/arquivo*        Comando/arquivo que deseja pesquisar.

A navegação dentro das páginas de manual é feita usando-se as teclas abaixo:

- q - Sai da página de manual
- PageDown ou f - Rola uma página abaixo (25 linhas em consoles tradicionais 80x25)
- PageUP ou w - Rola uma página acima (25 linhas em consoles tradicionais 80x25)
- SetaAcima ou k - Rola 1 linha acima
- SetaAbaixo ou e - Rola 1 linha abaixo
- r - Redesenha a tela (refresh)
- p ou g - Início da página
- h - Ajuda sobre as teclas e atalhes da página de manual
- s - Salva a página de manual em formato texto no arquivo especificado (por exemplo: /tmp/ls).

Cada seção da página de manual contém explicações sobre uma determinada parte do sistema. As seções são organizadas em diretórios separados e localizadas no diretório /usr/man. Os programas/arquivos são classificados nas seguintes seções:

1. Programas executáveis ou comandos internos
2. Chamadas do sistema (funções oferecidas pelo kernel)

3. Chamadas de Bibliotecas (funções dentro de bibliotecas do sistema)
4. Arquivos especiais (normalmente encontrados no diretório `/dev`)
5. Formatos de arquivos e convenções (`/etc/inittab` por exemplo).
6. Jogos
7. Pacotes de macros e convenções (por exemplo **man**)
8. Comandos de Administração do sistema (normalmente usados pelo root)
9. Rotinas do kernel (não padrões)

A documentação de um programa também pode ser encontrada em 2 ou mais categorias, como é o caso do arquivo `host_access` que é documentado na seção 3 (bibliotecas) e 5 (formatos de arquivo e convenções). Por este motivo é necessário digitar `man 5 hosts_access` para ler a página sobre o formato do arquivo, porque o comando **man** procura a página de manual nas seções em ordem crescente e a digitação do comando **man hosts\_access** abriria a seção 3.

As páginas de manual contém algumas regras para facilitar a compreensão do comando:

- Texto Negrito - Deve ser digitado exatamente como é mostrado
- [bla bla bla] - Qualquer coisa dentro de [ ] são opcionais

Exemplo, `man ls`, `man 5 hosts_access`.

## Info Pages

Idêntico as páginas de manual, mas utiliza a ligação (links) em textos indicados por um '\*', levando diretamente a itens da infopage, ou para outras info pages relacionadas com o tópico estudado. Se pressionarmos <Enter> em cima de uma palavra destacada, a **infopages** nos levará a seção correspondente.

A *info pages* é útil quando sabemos o nome do comando e queremos saber para o que ele serve. Assim como o **man**, também traz explicações detalhadas sobre uso, opções e comandos.

Para usar a info pages, digite:

**info** [*comando/programa*]

Se o nome do *comando/programa* não for digitado, a info pages mostra a lista de todos os manuais de *comandos/programas* disponíveis. A *info pages* possui algumas teclas de navegação úteis:

- q - Sai da info pages
- ? - Mostra a tela de ajuda (que contém a lista completa de teclas de navegação e muitas outras opções).
- n - Avança para a próxima página
- p - Volta uma página
- u - Sobre um nível do conteúdo (até chegar ao índice de documentos)
- m - Permite usar a localização para encontrar uma página do **info**. Pressione m, digite o comando e tecle <Enter> que será levado automaticamente a página correspondente. Caso não digite nada e aperte a tecla TAB, o **info** lhe mostrará todas as opções disponíveis de busca.

- d - Volta ao índice de documentos.

Existem muitas outras teclas de navegação úteis na info pages, mas estas são as mais usadas. Para mais detalhes, entre no programa **info** e pressione ?.

Exemplo, `info cvs`.

## Ajuda na própria linha de comandos

Ajuda rápida, sendo útil para sabermos quais opções podem ser usadas com o comando/programa.

Quase todos os comandos/programas **GNU/Linux** oferecem este recurso útil para consultas rápidas (e quando não precisamos dos detalhes das páginas de manual). Deve quando se sabe o nome do programa. Para acionar a *ajuda on line*, digite:

```
[comando] --help
```

*comando* - é o comando/programa que desejamos ter uma explicação rápida.

O Help on Line não funciona com comandos internos (embutidos no Bash), devendo ser usado o comando `man builtin` ou o `help comando` para obter ajuda.

Por exemplo, `ls --help`.

## help

Ajuda rápida, útil para saber que opções podem ser usadas com os *comandos internos* do interpretador de comandos. O comando **help** somente mostra a ajuda para comandos internos, para ter uma ajuda similar para comandos externos, veja “Ajuda na própria linha de comandos”. Para usar o **help** digite:

```
help [comando]
```

Por exemplo, `help echo`, `help exit`

## apropos

Apropos procura por *programas/comandos* através da descrição. É útil quando precisamos fazer alguma coisa mas não sabemos qual comando usar. Ele faz sua pesquisa nas páginas de manual existentes no sistema e lista os comandos/programas que atendem a consulta. Para usar o comando **apropos** digite:

```
apropos [descrição]
```

Digitando `apropos copy`, será mostrado todos os comandos que tem a palavra `copy` em sua descrição (provavelmente os programas que copiam arquivos, mas podem ser mostrados outros também).

## whatis

O **whatis** exhibe a função do comando especificado como argumento. Útil quando estamos na dúvida sobre o que determinado programa faz. Para usar o comando **whatis** digite:

```
whatis [programa/comando]
```

Digitando `whatis ls`, lhe será mostrada a descrição do comando **ls**.

## locate

Localiza uma palavra na estrutura de arquivos/diretórios do sistema. É útil quando queremos localizar onde um comando ou programa se encontra (para copia-lo, curiosidade, etc). A pesquisa é feita em um banco de dados construído com o comando **updatedb** sendo feita a partir do diretório raiz / e sub-diretórios. Para fazer uma consulta com o **locate** usamos:

**locate** [*expressão*]

A *expressão* deve ser o nome de um arquivo diretório ou ambos que serão procurados na estrutura de diretórios do sistema. Como a consulta por um programa costuma localizar também sua página de manual, é recomendável usar "*pipes*" para filtrar a saída do comando (para detalhes veja ???).

Por exemplo, para listar os diretórios que contém o nome "*cp*": `locate cp`. Agora mostrar somente arquivos binários, usamos: `locate cp|grep bin/`

## which

Localiza um programa na estrutura de diretórios do path. É muito semelhante ao **locate**, mas a busca é feita no `path` do sistema e somente são mostrados arquivos executáveis.

**which** [*programa/comando*].

## Documentos HOWTO's

São documentos em formato *texto*, *html*, etc, que explicam como fazer determinada tarefa ou como um programa funciona. Normalmente são feitos na linguagem **SGML** e convertidos para outros formatos (como o texto, HTML, Pos Script) depois de prontos.

Estes trazem explicações detalhadas desde como usar o **bash** até sobre como funciona o modem ou como montar um *servidor internet completo*. Os HOWTO's podem ser encontrados no diretório do projeto de documentação do **GNU/Linux** (LDP) em <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/> ou traduzidos para o Português pelo LDP-BR em <http://www.tldp.org/projetos/howto/traduzidos.php>. Caso tenha optado por instalar o pacote de HOWTO's de sua distribuição **GNU/Linux**, eles podem ser encontrados em: `/usr/doc/how-to`

## Documentação de Programas

São documentos instalados junto com os programas. Alguns programas também trazem o *aviso de copyright*, *changelogs*, *modelos*, *scripts*, *exemplos* e *FAQs* (*perguntas freqüentes*) junto com a documentação normal.

Seu princípio é o mesmo do How-to; documentar o programa. Estes arquivos estão localizados em:

`/usr/share/doc/[programa]`.

*Programa* é o nome do programa ou comando procurado.

## FAQ

*FAQ* é um arquivo de perguntas e respostas mais freqüentes sobre o programa. Normalmente os arquivos de FAQ estão localizados junto com a documentação principal do programa em `/usr/share/doc/[programa]`.

## RFC's

São textos que contém normas para a padronização dos serviços e protocolos da Internet (como a porta padrão de operação, comandos que devem ser utilizados, respostas) e outros detalhes usados para padronizar o uso de serviços Internet entre as mais diversas plataformas de computadores, com o objetivo de garantir a perfeita comunicação entre ambos. As RFC's podem ser obtidas de <http://rfc.net>.

O arquivo de uma RFC segue o formato RFC+Número, onde RFC descreve que o documento é uma RFC e Número é o seu número de identificação, como o documento RFC1939 que documenta o funcionamento e comandos do protocolo POP3. Os arquivos de RFC's podem ser encontrados no pacote da distribuição **Debian** e baseadas.

Segue abaixo o índice principal do diretório de RFC's que poderá ser usado para localizar RFC's específicas de um determinado serviço/assunto:

- 0001    PADRÕES OFICIAIS DO PROTOCOLO INTERNET. J. Reynolds, R. Braden. Março 2000. (Formato: TXT=86139 bytes) (Deixa obsoleto RFC2500, RFC2400, RFC2300, RFC2200, RFC2000, RFC1920, RFC1880, RFC1800, RFC1780, RFC1720, RFC1610, RFC1600, RFC1540, RFC1500, RFC1410, RFC1360, RFC1280, RFC1250, RFC1200, RFC1140, RFC1130, RFC1100, RFC1083) (Também RFC2600)
- 0002    Números designados. J. Reynolds, J. Postel. Outubro 1994. (Formato: TXT=458860 bytes) (Também RFC1700)
- 0003    Requerimentos do sistema. R. Braden. Outubro 1989. (Formato: TXT=528939 bytes) (Também RFC1122, RFC1123)
- 0004    Requerimentos do Gateway. R. Braden, J. Postel. Junho 1987. (Formato: TXT=125039 bytes) (Também RFC1009)
- 0005    Protocolo Internet. J. Postel. Setembro 1981. (Formato: TXT=241903 bytes) (Também RFC0791, RFC0950, RFC0919, RFC0922, RFC792, RFC1112)
- 0006    User Datagram Protocol. J. Postel. Agosto 1980. (Formato: TXT=5896 bytes) (Também RFC0768)
- 0007    Transmission Control Protocol. J. Postel. September 1981. (Formato: TXT=172710 bytes) (Também RFC0793)
- 0008    Protocolo Telnet. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=44639 bytes) (Também RFC0854, RFC0855)
- 0009    File Transfer Protocol. J. Postel, J. Reynolds. Outubro 1985. (Formato: TXT=148316 bytes) (Também RFC0959)
- 0010    SMTP Service Extensions. J. Klensin, N. Freed, M. Rose, E. Stefferud & D. Crocker. Novembro 1995. (Formato: TXT=23299 bytes) (Deixa obsoleto RFC1651) (Também RFC821, RFC1869)
- 0011    Standard for the format of ARPA Internet text messages. D. Crocker. 13-Ago-1982. (Formato: TXT=109200 bytes) (Deixa obsoleto RFC1653) (Também RFC0822)
- 0012    Network Time Protocol. D. Mills. Setembro 1989. (Formato: TXT=193 bytes) (Também RFC1119)
- 0013    Domain Name System. P. Mockapetris. Novembro 1987. (Formato: TXT=248726 bytes) (Também RFC1034, RFC1035)

- 0014 Mail Routing and the Domain System. C. Partridge. Janeiro 1986. (Formato: TXT=18182 bytes) (Também RFC0974)
- 0015 Simple Network Management Protocol. J. Case, M. Fedor, M. Schoffstall, J. Davin. Maio 1990. (Formato: TXT=72876 bytes) (Também RFC1157)
- 0016 Structure of Management Information. M. Rose, K. McCloghrie. Maio 1990. (Formato: TXT=82279 bytes) (Deixa obsoleto RFC1065) (Também RFC1155)
- 0017 Management Information Base. K. McCloghrie, M. Rose. March 1991. (Formato: TXT=142158 bytes) (Deixa obsoleto RFC1158) (Também RFC1213)
- 0018 Exterior Gateway Protocol. D. Mills. Abril 1984. (Formato: TXT=63836 bytes) (Também RFC0904)
- 0019 NetBIOS Service Protocols. NetBIOS Working Group. Março 1987. (Formato: TXT=319750 bytes) (Também RFC1001, RFC1002)
- 0020 Echo Protocol. J. Postel. Maio 1983. (Formato: TXT=1237 bytes) (Também RFC0862)
- 0021 Discard Protocol. J. Postel. Maio 1983. (Formato: TXT=1239 bytes) (Também RFC0863)
- 0022 Character Generator Protocol. J. Postel. Maio 1983. (Formato: TXT=6842 bytes) (Também RFC0864)
- 0023 Quote of the Day Protocol. J. Postel. Maio 1983. (Formato: TXT=1676 bytes) (Também RFC0865)
- 0024 Active Users Protocol. J. Postel. Maio 1983. (Formato: TXT=2029 bytes) (Também RFC0866)
- 0025 Daytime Protocol. J. Postel. Maio 1983. (Formato: TXT=2289 bytes) (Também RFC0867)
- 0026 Time Server Protocol. J. Postel. Maio 1983. (Formato: TXT=3024 bytes) (Também RFC0868)
- 0027 Binary Transmission Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=8965 bytes) (Também RFC0856)
- 0028 Echo Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=10859 bytes) (Também RFC0857)
- 0029 Suppress Go Ahead Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=3712 bytes) (Também RFC0858)
- 0030 Status Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=4273 bytes) (Também RFC0859)
- 0031 Timing Mark Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=7881 bytes) (Também RFC0860)
- 0032 Extended Options List Telnet Option. J. Postel, J. Reynolds. Maio 1983. (Formato: TXT=3068 bytes) (Também RFC0861)
- 0033 Trivial File Transfer Protocol. K. Sollins. Julho 1992. (Formato: TXT=24599 bytes) (Também RFC1350)
- 0034 Routing Information Protocol. C. Hedrick. Junho 1988. (Formato: TXT=91435 bytes) (Também RFC1058)

- 0035 ISO Transport Service on top of the TCP (Version: 3). M. Rose, D. Cass. Maio 1978. (Formato: TXT=30662 bytes) (Também RFC1006)
- 0036 Transmission of IP and ARP over FDDI Networks. D. Katz. Janeiro 1993. (Formato: TXT=22077 bytes) (Também RFC1390)
- 0037 An Ethernet Address Resolution Protocol. David C. Plummer. Novembro 1982. (Formato: TXT=21556 bytes) (Também RFC0826)
- 0038 A Reverse Address Resolution Protocol. Ross Finlayson, Timothy Mann, Jeffrey Mogul, Marvin Theimer. Junho 1984. (Formato: TXT=9345 bytes) (Também RFC0903)
- 0039 Interface Message Processor: Especificações para a Interconexão de um computador e um IMP (Revisado). BBN. Dezembro 1981. (fora de linha)
- 0040 Host Access Protocol specification. Bolt Beranek and Newman. Agosto 1993. (Formato: TXT=152740 bytes) (Deixa obsoleto RFC0907) (Também RFC1221)
- 0041 Standard for the transmission of IP datagrams over Ethernet networks. C. Hornig. Abril 1984. (Formato: TXT=5697 bytes) (Também RFC0894)
- 0042 Standard for the transmission of IP datagrams over experimental Ethernet networks. J. Postel. Abril 1984. (Formato: TXT=4985 bytes) (Também RFC0895)
- 0043 Standard for the transmission of IP datagrams over IEEE 802 networks. J. Postel, J.K. Reynolds. Agosto 1993. (Formato: TXT=34359 bytes) (Deixa obsoleto RFC0948) (Também RFC1042)
- 0044 DCN Local-Network Protocols. D.L. Mills. Agosto 1993. (Formato: TXT=65340 bytes) (Também RFC0891)
- 0045 Internet Protocol on Network System's HYPERchannel: Protocol Specification. K. Hardwick, J. Lekashman. Agosto 1993. (Formato: TXT=100836 bytes) (Também RFC1044)
- 0046 Transmitting IP traffic over ARCNET networks. D. Provan. Agosto 1993. (Formato: TXT=16565 bytes) (Deixa obsoleto RFC1051) (Também RFC1201)
- 0047 Nonstandard for transmission of IP datagrams over serial lines: SLIP. J.L. Romkey. Agosto 1993. (Formato: TXT=12578 bytes) (Também RFC1055)
- 0048 Standard for the transmission of IP datagrams over NetBIOS networks. L.J. McLaughlin. Agosto 1993. (Formato: TXT=5579 bytes) (Também RFC1088)
- 0049 Standard for the transmission of 802.2 packets over IPX networks. L.J. McLaughlin. Agosto 1993. (Formato: TXT=7902 bytes) (Também RFC1132)
- 0050 Definitions of Managed Objects for the Ethernet-like Interface Types. F. Kastholz. Julho 1994. (Formato: TXT=39008, bytes) (Deixa obsoleto RFC1623, RFC1398) (Também RFC1643)
- 0051 The Point-to-Point Protocol (PPP). W. Simpson, Editor. Julho 1994. (Formato: TXT=151158 bytes) (Deixa obsoleto: RFC1549) (Também RFC1661, RFC1662)
- 0052 The Transmission of IP Datagrams over the SMDS Service. D. Piscitello, J. Lawrence. Março 1991. (Formato: TXT=24662 bytes) (Também RFC1209)
- 0053 Post Office Protocol - Version 3. J. Myers & M. Rose. Maio 1996. (Formato: TXT=47018 bytes) (Deixa Obsoleto: RFC1725) (Também RFC1939)
- 0054 OSPF Version 2. J. Moy. Abril 1998. (Formato: TXT=447367 bytes) (Também RFC2328)

- 0055 Multiprotocol Interconnect over Frame Relay. C. Brown, A. Malis. Setembro 1998. (Formato: TXT=74671 bytes) (Deixa Obsoleto: RFC1490, RFC1294) (Também RFC2427)
- 0056 RIP Version 2. G. Malkin. Novembro 1998. (Formato: TXT=98462 bytes) (Atualiza RFC1723, RFC1388) (Também RFC2453)
- 0057 RIP Version 2 Protocol Applicability Statement. G. Malkin. Novembro 1994. (Formato: TXT=10236 bytes) (Também RFC1722)
- 0058 Structure of Management Information Version 2 (SMIv2. K. McCloghrie, D. Perkins, J. Schoenwaelder. Abril 1999. (Formato: TXT=89712 bytes) (Deixa Obsoleto RFC1902) (Também RFC2578, RFC2579)
- 0059 Remote Network Monitoring Management Information Base. S. Waldbusser. Maio 2000. (Formato: TXT=198676 bytes) (Deixa Obsoleto RFC1757) (Também RFC2819)

## Internet

Certamente o melhor suporte ao **GNU/Linux** é via Internet, veja abaixo alguns locais úteis de onde pode obter ajuda ou se atualizar.

## Páginas Internet de Referência

Existem boas páginas Internet Nacionais e Internacionais sobre o **GNU/Linux** e assuntos relacionados com este sistema. A maioria trazem documentos e explicações sobre configuração, instalação, manutenção, documentação, suporte, etc.

Estas páginas podem ser encontradas através de ferramentas de busca. Entre outras páginas, posso citar as seguintes:

- <http://www.debianbrasil.org/> Projeto Debian-Br. A **Debian** é uma distribuição de **Linux** conhecida por sua qualidade, grande número de pacotes, estabilidade, facilidade de atualização, desenvolvimento aberto, segurança, ferramentas de gerenciamento de servidores e comprometimento com o software livre.

A Debian é feita originalmente em inglês e traduzida por grupos em vários lugares do mundo. O projeto **Debian-br** destina-se a colaborar na tradução da **Debian** para o Português (nossa língua-mãe). Através desse projeto, todos poderão, da forma colaborativa como na **Debian**, trazer essa excelente distribuição em nosso idioma!

Participe:

- Você pode pegar um documento pra traduzir
- Reformular a página do projeto
- Programando para o projeto
- Sendo um desenvolvedor da Debian
- A pagina do projeto é a <http://www.debianbrasil.org/>
- Revisar documentação
- Ou participar de outras tarefas do seu interesse!

Entre em contato com o responsável pelo projeto pelo email <debian-br@listas.cipsga.org.br> para saber como entrar no projeto ou visite a página <http://www.debianbrasil.org/>. Todos os interessados estão convidados a participar do projeto!

- <http://www.tldp.org/> - Projeto de documentação do **GNU/Linux** no Brasil. Toda a documentação traduzida para o Português do Brasil pode ser encontrada lá.

Responsável pela página: <ricardo@conectiva.com.br> endereço: <http://www.tldp.org/>.

- <http://www.linux.org/> - Página oficial do **GNU/Linux** mantida pela *Transmeta* (a empresa que Linus Torvalds vem trabalhando atualmente). Muita referência sobre **GNU/Linux**, distribuições, hardwares, softwares, downloads, etc.

Responsável pela página: <webmaster@linux.org> endereço: <http://www.linux.org/>.

- <http://www.oreill.com/safari/> - Neste site você encontra os livros publicados sobre a licença OpenBook da O'Reil. Na maioria livros que não atende mais propósitos atualmente e livros em que os autores concordaram em licenciar sob os termos OpenBook.

Endereço: <http://www.oreill.com/safari/>.

Caso conhecer uma página de Internet que contenha materiais úteis a comunidade **GNU/Linux** ou desejar incluir a sua, entre em contato para sua inclusão na próxima versão do guia junto com uma descrição da página.

## Listas de discussão

São grupos de usuários que trocam mensagens entre si, resolvem dúvidas, ajudam na configuração de programas, instalação, etc. É considerado o melhor suporte ao **GNU/Linux** pois qualquer participante pode ser beneficiar das soluções discutidas. Existem milhares de listas de discussões sobre o **GNU/Linux** espalhadas pelo mundo, em Português existem algumas dezenas.

Algumas listas são específicas a um determinado assunto do sistema, algumas são feitas para usuários iniciantes ou avançados, outras falam praticamente de tudo. Existem desde usuários iniciantes, hackers, consultores, administradores de redes experientes e gurus participando de listas e oferecendo suporte de graça a quem se aventurar em instalar e usar o sistema **GNU/Linux**.

A lista de discussão funciona da seguinte forma: você se inscreve na lista enviando uma mensagem ao endereço de inscrição, será enviada um pedido de confirmação por e-mail, simplesmente dê um reply na mensagem para ser cadastrado. Pronto! agora você estará participando do grupo de usuários e receberá todas as mensagens dos participantes do grupo. Assim você poderá enviar sua mensagem e ela será vista por todos os participantes da lista.

Da mesma forma, você pode responder uma dúvida de outro usuário da lista ou discutir algum assunto, tirar alguma dúvida sobre a dúvida de outra pessoa, etc.

Não tenha vergonha de enviar sua pergunta, participar de listas de discussão é uma experiência quase obrigatório de um *Linuxer*. Abaixo segue uma relação de listas de discussão em Português com a descrição, endereço de inscrição, e o que você deve fazer para ser cadastrado:

<debian-user-portuguese@lists.debian.org> **Lista de discussão para usuários Portugueses da Debian.** Também são discutidos assuntos relacionados ao Linux em geral. A inscrição é aberta a todos os interessados.

Para se inscrever, envie uma mensagem para <debian-user-portuguese-request@lists.debian.org>

contendo a palavra `subscribe` no assunto da mensagem. Será enviada uma mensagem a você pedindo a confirmação da inscrição na lista de discussão, simplesmente dê um `reply` na mensagem (responder) e você estará cadastrado e poderá enviar e receber mensagens dos participantes.

<debian-news-portuguese@lists.debian.org> A **Debian** é extremamente bem estruturada quanto a divulgações e notícias, várias listas de email e várias páginas compõe essa base. A *Debian Weekly News* é especialmente importante pois dá uma visão geral do que se passou na **Debian** durante a semana. E não traz apenas traduções mas também adições dos acontecimentos atuais da **Debian** no Brasil, ou projetos concluídos ou lançados pela equipe *Debian-br* (<http://www.debianbrasil.org/>).

Essa lista NÃO é usada para resolução de dúvidas e problemas, apenas para o RECEBIMENTO de notícias relacionadas a Debian. Não poste mensagens nela!

Para se inscrever, envie uma mensagem para <debian-news-portuguese-request@lists.debian.org> contendo a palavra `subscribe` no assunto da mensagem. Será enviada uma mensagem a você pedindo a confirmação da inscrição na lista de discussão, simplesmente dê um `reply` na mensagem (responder) e você passará a receber as notícias sobre a Debian em Português.

<linux-br@unicamp.br>

Lista de discussão que cobre assuntos diversos. Esta lista é voltada para usuários com bons conhecimentos no **GNU/Linux**, são abordados assuntos como redes, configurações, etc. Esta é uma lista moderada, o que significa que a mensagem que envia passam por uma pessoa que verifica (modera) e a libera caso estejam dentro das normas adotada na lista. É uma lista de alto nível e recomendada para quem deseja fugir de mensagens como não consigo instalar o Linux, não sei compilar o kernel, o que eu faço quando vejo uma tela com o nome `login:?`, etc.

Para se inscrever nesta lista, envie uma mensagem para: <linux-br-request@unicamp.br> contendo a palavra `subscribe` no assunto da mensagem e aguarde o recebimento da confirmação da inscrição. Apenas responda a mensagem de confirmação para se inscrever. Para se descadastrar envie uma mensagem para o mesmo endereço mas use a palavra `unsubscribe`.

<dicas-l@unicamp.br>

Esta lista envia diariamente uma dica de **Unix**, sistemas da Microsoft ou novidades da Internet.

Para se inscreve nesta lista de discussão, envie uma mensagem para: <dicas-l-request@unicamp.br> contendo a palavra `subscribe` no corpo da mensagem e aguarde o recebimento da confirmação da inscrição. Apenas responda a mensagem de confirmação para confirmar sua inscrição na lista. Para se descadastrar envie uma mensagem para o mesmo endereço mas use a palavra `unsubscribe`.

Esta listagem deveria estar mais completa, mas eu não lembro de todas as listas!. Também recomendo dar uma olhada em "Listas de Discussão via Email" que descreve recomendações de comportamento em listas de discussão.

## Netiqueta

São recomendações que tem como objetivo facilitar a comunicação através dos recursos de uma rede. O nome *Netiqueta* vem de "Etiqueta de Rede" (*Net Etiquete*). O material desta seção foi escrito com base nos anos de observação que tive via internet e também com referência a rfc 1855.

## Recomendações Gerais sobre a Comunicação Eletrônica

- Como recomendação geral, lembre-se que a conversa via internet é feita sempre de uma para outra pessoa ou de uma para várias pessoas, e que a forma de comunicação é a mesma que utilizaria se estivesse de frente a frente com a pessoa. Nunca diga algo que não diria se estivesse diante da outra pessoa. Existem pessoas que por estar atrás de um monitor, se sentem "maiores" se esquecendo disso e causando prejuízos de comunicação (e sem imaginar que a pessoa do outro lado da linha existe).

Apesar do modo que as frases são escritas expressarem o jeito que a outra pessoa está do outro lado da linha e seu tom de comunicação no decorrer da conversa, existem algumas coisas que não podem ser totalmente expressadas através da Internet, como por exemplo a expressão da "face" das pessoas. Para isto foram criados símbolos chamados *smileys* que expressam a face da outra pessoa em determinado momento, e dependendo do sentido da conversa, um smiley pode expressar corretamente a intenção de sua frase. Os mais usados são os seguintes:

:-) --> Sorriso  
:-( --> Triste  
;-) --> Piscadinha  
:-O --> De boca aberta  
:-| --> Sem graça  
8-) --> De óculos  
|-) --> Com sono e feliz  
&:-) --> Bobo

Para entender o sentido do smiley, veja ele de lado (45 graus). Use os smileys em suas conversas, mas com cautela. Não espere que a inclusão de um smiley sorridente ":-)" deixe o destinatário da mensagem contente com um comentário rude ou insulto.

- **ESCREVER EM MAIÚSCULAS** significa gritar quando escrever mensagens eletrônicas.
- Use **\*asteriscos\*** para destacar uma palavra ou frase. **\_Isso\_** indica uma palavra/frase sublinhada.
- Se você troca mensagens com pessoas do mundo todo, não espere que um japonês responda logo seu e-mail que enviou as 15:00 da tarde. A essa hora no país dele, ele está roncando forte na cama e sonhando com a placa 3D que vai ganhar para melhorar o desempenho de seus jogos de **Linux**.

- Durante a comunicação com pessoas de diferentes regiões (ou países), evite a utilização de gírias, ou expressões regionais. Uma interpretação em uma determinada região não garante que ela tenha o mesmo significado para seu destinatário, as vezes pode ser até ofensiva.
- Assuma que sua mensagem está trafegando sobre uma via não segura, desta forma não envie informações pessoais que não enviaria em uma carta comum. O uso de criptografia pode garantir melhor segurança na transmissão de dados.

## Email

- Tenha o hábito de colocar sempre um assunto na mensagem que envia para identificar seu conteúdo.
- Respeite os direitos autorais das mensagens de e-mail. Se precisar encaminhar mensagens, preserve seu conteúdo original.
- Procure limitar o tamanho da linha a 70 caracteres. Muitos usuários utilizam cliente de e-mail em modo texto, e nem todo mundo usa a mesma resolução que você.
- Caso o e-mail que responda tenha mais que 100 linhas, é recomendável colocar a palavra "LONGA" no assunto da mensagem. Se possível corte as partes não necessárias da mensagens de respostas tendo o cuidado de não "cortar" de forma mal educada a mensagem de outra pessoa.
- Caso utiliza um editor programa de e-mails com suporte a HTML, envie o e-mail utilizando ambos os formatos TEXTO e HTML, muitos administradores Linux utilizam sistemas que não suportam HTML.
- Não espere que o espaçamento ou desenhos ASCII usados em uma mensagem sejam mostrados corretamente em todos os sistemas.
- Utilize sempre uma assinatura no final da mensagem para identificar você e principalmente seu endereço de e-mail. Em alguns clientes de e-mail, o campo Reply-to é bagunçado, e em e-mails redirecionados o endereço de resposta é excluído. A assinatura facilita encontrar o remetente da mensagem. Tente manter a assinatura em um tamanho de no máximo 4 linhas.
- Não repasse mensagens de corrente por e-mail. Elas tem somente o objetivo de espalhar boatos na Internet e se espalhar. Normalmente elas vem com uma história bonita e no final diz se não repassar acontecerá tudo ao contrário com você ou algo do tipo. *Não vai acontecer nada! ignore isso e não entre na corrente!*

Pelas políticas da Internet, você pode ter sua conta de e-mail perdida se fizer mal uso dele.

## Telegram/Whatsapp/Messenger/Gtalk/Skype

Ferramentas de mensagens instantâneas são eficientes, alertando a presença on-line do usuário, auxiliando na redução de custos, etc. Este documento inclui algumas recomendações etiqueta para os usuários aproveitarem melhor as ferramentas de comunicação que seguem o padrão IM:

- De atenção ao status da outra pessoa. Se ela estiver "on-line" ou "free for chat" significa que ela está desocupada e que pode conversar naquele instante. Se estiver como não perturbe, envie somente mensagens se for mesmo preciso.
- EVITE colocar nicks chamativos e caracteres exóticos. Nem todos os usuários vêem o nick da mesma forma que a pessoa que os colocou.
- Seja também sensato ao usar ferramentas de mensagem instantanea. Não entre nele caso não possa conversar, ou avise isso mudando seu status para o mais adequado para a situação, assim os outros poderão entender que está longe do computador, não disponível ou ocupado.

- É recomendável ser prudente quanto ao envio de mensagens, não envie mais do que 4 mensagens seguidas, pois a outra pessoa terá dificuldades para responder a todas elas mais outra que talvez possa estar recebendo de outras (ou nem tenha recebido, caso exista algum problema temporário no servidor).
- Guarde seu login e senha em lugar seguro. Caso ela seja perdida, você terá trabalho para avisar a todos de sua lista de contato.
- Sempre que enviar uma URL, procure do que se trata na mensagem.
- No modo de chat, use as recomendações descritas sobre o talk (em "Talk").
- Como em toda comunicação on-line, seja cauteloso quando a pessoa que conversa. Nem sempre quem conversamos do outro lado é a pessoa que esperamos encontrar. Lembre-se que um registro falso e uma identidade pode ser criada sem dificuldades por qualquer pessoa.

## Talk

- Use sempre quebra de linhas ao escrever suas mensagens, use pelo menos 70 caracteres para escrever suas mensagens de talk. Evite escrever continuamente até a borda para fazer quebra de linha automática, alguns clientes de talk não aceitam isso corretamente.
- Sempre que termina uma frase, deixe uma linha em branco (tecle enter 2 vezes) para indicar que a outra pessoa pode iniciar a digitação.
- Sempre se despeça da outra pessoa e espere ela responder antes de fechar uma seção de conversação. O respeito mútuo durante um diálogo é essencial :-)
- Lembre-se que o talk normalmente interrompe as pessoas que trabalham nativamente no console. Evite dar talk para estranhos, pois podem fazer uma má impressão de você. Tente antes estabelecer outros meios de comunicação.
- Se a outra pessoa não responder, não assuma de cara que ela está ignorando você ou não levando sua conversa muito bem. Ela pode simplesmente estar ocupada, trabalhando, ou com problemas no cliente de talk. Alguns cliente de talk dão problemas durante a comunicação remota, lembre-se também que sua comunicação é via UDP :-)
- Se a pessoa não responder seus talks durante certo tempo, não deixe ele infinitamente beepando a pessoa. Tente mais tarde :-)
- Seja atencioso caso utilize mais de uma seção de talk ao mesmo tempo.
- O talk também leva em consideração sua habilidade de digitação. Muitos erros e correções contínuas fazem a outra pessoa ter uma noção de você, suas experiências, etc :-)

## Listas de Discussão via Email

- Tente se manter dentro do assunto quando responder mensagens de listas. Seja claro e explicativo ao mesmo tempo :-)
- Sempre coloque um assunto (subject) na mensagem. O assunto serve como um resumo do problema ou dúvida que tem. Alguns usuários, principalmente os que participam de várias listas de discussão, verificam o assunto da mensagem e podem simplesmente descartar a mensagem sem lê-la porque as vezes ele não conhece sobre aquele assunto.
- Nunca use "Socorro!", "Help!" ou coisa do gênero como assunto, seja objetivo sobre o problema/dúvida que tem: "*Falha ao carregar módulo no do kernel*", "*SMAIL retorna a mensagem Access denied*", "*Novidades: Nova versão do guia Foca Linux*";-).

- Procure enviar mensagens em formato texto ao invés de HTML para as listas de discussão pois isto faz com que a mensagem seja vista por todos os participantes (muitos dos usuários **GNU/Linux** usam leitores de e-mail que não suportam formato html) e diminui drasticamente o tamanho da mensagem porque o formato texto não usa tags e outros elementos que a linguagem HTML contém (muitos dos usuários costumam participar de várias listas de discussão, e mensagens em HTML levam a um excesso de tráfego e tempo de conexão).
- Tenha cautela e bom senso em suas mensagens para listas e grupos de discussão, considere que cada mensagem que posta é são arquivadas para futura referência.
- Quando o conteúdo das mensagens tomar outro rumo, é ético modificar o assunto do e-mail para se adequar ao novo conteúdo da mensagem. Por exemplo, Correção nas regras de Netiqueta para Conversa de pessoa para pessoa (Era: Correção das regras de Netiqueta).
- Quando a conversa em grupo sair do assunto e envolver apenas duas pessoas, é conveniente retirar os endereços das pessoas/listas do CC.
- Não mande arquivos grandes para as listas, principalmente se eles tiverem mais que 40Kb de tamanho. Se precisar enviar arquivos maiores que isso, envie diretamente para os e-mails dos interessados depois de perguntar.
- Quando enviar mensagens para listas de discussão, seja educado e cordial quanto ao conteúdo de sua mensagem. Envie CC's para as pessoas que dizem respeito ao assunto, assim com a lista.
- Tente ignorar ou não responda mensagens de "Guerras" em listas (*Flame Wars*), caso queira reponde-la por algum tipo de agressão de quem mandou a mensagem, esperar para responde-la a noite (nunca é garantida uma boa resposta no momento que está de cabeça quente). Lembre-se de quando responde uma mensagem de "Flame War" a "altura" de quem mandou seus ataques, está sendo igualmente tão baixo quando o "nível" dessa pessoa.
- Caso se desentenda com alguma pessoa em uma lista de discussão, não envie mensagens agressivas para a listas, se precisar, faça isso diretamente para a pessoa! Você pode se arrepender disso mais tarde.
- Não culpe o administrador da lista pelos usuários que participam dela. Notifique somente usuários que não estejam colaborando com a lista e outras coisas que prejudiquem seu funcionamento. Administradores preservam o funcionamento das listas, e não o comportamento dos usuários.
- Não use auto respostas para listas de discussão. Pelos inconvenientes causados, você pode ser descadastrado ou banido de se inscrever na lista/newsgroup.
- Salve as mensagens de inscrição que recebe da lista. Ela contém detalhes sobre seus recursos, e a senha usada muitas vezes para se descadastrar dela ou modificar suas permissões de usuário. O administrador pode te ajudar nessa tarefa, mas não espere que ele esteja sempre disponível para realizar tarefas que podem ser feitas pelo próprio usuário.
- Muitas pessoas reclamam do excesso de mensagens recebidas das listas de discussão. Se você recebe muitas mensagens, procure usar os *filtros de mensagens* para organiza-las. O que eles fazem é procurar por campos na mensagem, como o remetente, e enviar para um local separado. No final da filtragem, todas as mensagens de listas de discussão estarão em locais separados e as mensagens enviadas diretamente a você entrarão na caixa de correio principal, por exemplo.

Um filtro de mensagens muito usado no **GNU/Linux** é o **procmil**, para maiores detalhes consulte a documentação deste programa.

O Netscape também tem recursos de filtros de mensagem que podem ser criadas facilmente através da opção "Arquivo/Nova SubPasta" ("File/New Subfolder") do programa de E-mail. Então defina

as regras através do menu "Editar/Filtros de Mensagens" ("Edit/Message filters") clicando no botão "Novo"("New").

---

# Capítulo 22. Apêndice

Este capítulo contém considerações sobre o guia Foca GNU/Linux.

## Sobre este guia

O Guia Foca foi lançado pela primeira vez em 12 de Novembro de 1999. A versão que está lendo contém o(s) nível(is) de aprendizado (Iniciante, Intermediário, Avançado):

- Avançado

A versão que esta lendo agora foi gerada com as seguintes opções:

- Descrição detalhada de comandos
- Opções usadas em comandos e programas
- Observações sobre comandos e configurações
- Exemplos para a melhor compreensão do assunto discutido.

A versão que está lendo tem o objetivo de servir como referência a usuários:

- *Avançado* - que já dominam grande parte do sistema operacional e procuram aprender mais sobre os seus detalhes e configurações especiais.

Uma versão que abrange um diferente nível de aprendizado ou mais completa pode ser baixada de Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>].

O *Foca GNU/Linux* é atualizado frequentemente, por este motivo recomendo que assine um dos canais de comunicação (ou todos :-)) para ser informado de novas versões:

- Siga o Guia Foca no Twitter <[@focalinux](https://twitter.com/focalinux)>
- Siga os avisos do Guia Foca no Telegram: Avisos de Atualização do Guia Foca no Telegram [<https://t.me/guiafocaAvisos>]
- Assinar o RSS presente na página do guia, e avisos de atualizações.
- Siga a página do [@guiafoca](https://www.facebook.com/focalinux) no Facebook: Página do Guia Foca no Facebook [<https://www.facebook.com/focalinux>]
- A ficha do aviso de atualizações na página web em Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>] no fim da página principal. Após preencher a ficha do aviso de atualizações, você será notificado sobre novidades do Guia Foca.

Versões diferentes deste guia podem ser geradas a partir do código fonte XML ou obtidas através da home page principal (para detalhes veja “Onde encontrar a versão mais nova do guia?”).

## Sobre o Autor

Gleydson Mazioli da Silva é Capixaba, nascido em Vila Velha. Amante de eletrônica desde criança, foi atraído para a informática através da curiosidade em funcionamento e reparo de hardware.

Se dedica ao sistema **Linux** desde 1997. determinado na realização de testes de ferramentas e sistemas avaliando pontos fortes e fracos de cada uma. Logo que iniciou em **Linux** passou a estudar exaustivamente aspectos técnicos de distribuições e rede em **Linux/BSD**.

Entre coisas que gosta de fazer/implementar em **Linux**: possibilidade de pesquisa e atualização de conhecimento constante, automatização e tomada inteligente de decisões, níveis de segurança da informação (tanto físico e lógico), firewalls, virtualização, redes virtuais, integração de sistemas, forense computacional, documentação de processos, desenvolvimento de ferramentas GPL para a comunidade, depuração, desenvolvimento de documentações, etc.

Um dos desenvolvedores da distribuição *Liberdade*, *CAETECT*, *Debian-BR* e desenvolvedor oficial da distribuição *Debian*. Atuou como tradutor do LDP-BR, traduzindo vários HOW-TOs importantes para a comunidade Linux Brasileira. Também é um tdos administradores do projeto CIPSGA, cuidando de uma infinidade de serviços que o projeto oferece a comunidade que deseja estrutura para hospedar, fortalecer e manter projetos em software livre.

Trabalhou para algumas empresas do Espírito Santo, no Governo Federal e de estados na implantação de sistemas em software livre. Atualmente atua como gerente de tecnologia da Spirit Linux, uma empresa focada na inovação, avaliação e integração de ferramentas de código aberto junto a seus clientes.

Concorda com certificações, mas destaca que o mais importante é aproveitar a oportunidade dada pela certificação para estudo e auto avaliação de seus pontos fracos e assim procurar melhora-los. Possui certificação LPI nível 3 e um ISO9001 internacional em Administração Linux, como primeiro no ranking Brasileiro.

E-mail: E-mail: <gleydson@guiafoca.org>, Twitter: <@gleydsonmazioli>.

## Referências de auxílio ao desenvolvimento do guia

- As seções sobre comandos/programas foram construídas após uso, teste e observação do comportamento das opções dos comandos/programas, help on line, páginas de manual, info pages e documentação técnica do sistema.
- How-tos do Linux (principalmente o *Networking Howto*, *Security-Howto*) ajudaram a formar a base de desenvolvimento do guia e desenvolver algumas seções (versões *Intermediário* e *Avançado* somente).
- Todos os exemplos de comandos e seções descritivas do guia são de minha autoria. Quanto a exemplos de configurações e utilização de programas, será citada a origem que foram baseados em cada capítulo, valorizando trabalho de seus respectivos autores.
- Uso de programas e macetes aprendidos no dia a dia para gerenciar máquinas, controlar redes e automatizar sistemas.
- As seções do nível avançado foram construídas com base em testes e usando metodologia própria com o objetivo de oferecer uma documentação objetiva e clara ao administrador de rede, auxiliando-o na tomada de decisões para solução de problemas. Partes extraídas de outras documentações no nível Avançado terão as referências explicitamente citadas em seus respectivos capítulos.
- Manual de Instalação da *Debian GNU/Linux* - Os capítulos contendo materiais extraídos do manual de instalação da Debian são muito úteis e explicativos, seria desnecessário reescrever um material como este. O texto é claro e didaticamente organizado, o documento aborda detalhes técnicos úteis sobre hardware em geral e o Linux ausentes nos manuais de outras distribuições Linux.

## Onde encontrar a versão mais nova do guia?

Novas versões deste guia, avisos de lançamento, outros níveis de aprendizado (Iniciante, Intermediário e Avançado), versões para outras distribuições Linux podem ser encontradas em: Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>].

Se quiser receber notificações de novas versões, use uma das formas a seguir:

- Siga a página do @guiafoca no Facebook: Página do Guia Foca no Facebook [<https://www.facebook.com/focalinux>]
- Siga os avisos do Guia Foca no Telegram: Avisos de Atualização do Guia Foca no Telegram [<https://t.me/guiafocaAvisos>]
- A ficha do aviso de atualizações na página web em Página Oficial do guia Foca GNU/Linux [<http://www.guiafoca.org>] no fim da página principal. Após preencher a ficha do aviso de atualizações, você será notificado sobre novidades do Guia Foca.
- por E-Mail: envie uma mensagem para <[gleydson@guiafoca.org](mailto:gleydson@guiafoca.org)> pedindo para ser incluído na lista de atualizações do guia ou preencha o formulário encontrado no final da Home Page do guia.
- Twitter: Assine o Twitter do guia Foca: @focalinux
- RSS: Assine o RSS na página oficial do guia (citado acima) para receber atualizações e novidades.

## Colaboradores do Guia

Entre as principais colaborações até a versão atual, posso citar as seguintes (classificação cronológica):

- Djalma Valois <[djalma@cipsga.org.br](mailto:djalma@cipsga.org.br)> - Pela atual hospedagem do Foca GNU/Linux. Estou muito feliz vendo o Foca GNU/Linux fazendo parte de um projeto tão positivo como o CIPSGA é para o crescimento e desenvolvimento do software livre nacional.
- Bakurih <[bakurih@yahoo.com](mailto:bakurih@yahoo.com)> - Revisão inicial do guia, após suas primeiras versões.
- Eduardo Marcel Maçan <[macan@debian.org](mailto:macan@debian.org)> - Pela antiga hospedagem, na época do site metainfo.
- Michelle Ribeiro <[michellemazioli@gmail.com](mailto:michellemazioli@gmail.com)> - Por dispensar parte de seu atencioso tempo enviando revisões e sugestões que estão melhorando bastante a qualidade do guia. Entre eles detalhes que passaram despercebidos durante muito tempo no guia e página principal.

E também por cuidar do fonte do guia ;-)

- Augusto Campos <[brain@matrix.com.br](mailto:brain@matrix.com.br)> - Descrição sobre a distribuição **Suse** e constantes divulgações do guia no antigo site *br-linux.org*
- Paulo Henrique Baptista de Oliveira <[baptista@linuxsolutions.com.br](mailto:baptista@linuxsolutions.com.br)> - Pelo apoio moral oferecido durante os freqüentes lançamentos do guia, acompanhamento e divulgação.
- Diego Abadan <[diego@hipernet.ufsc.br](mailto:diego@hipernet.ufsc.br)> - Envio de correções significativas, novos endereços de listas de discussão.
- Alexandre Costa <[alebyte@bol.com.br](mailto:alebyte@bol.com.br)> - Envio de centenas de patches ortográficos nas versões Iniciante e Intermediário do guia que passaram despercebidas durante várias versões do guia...

- Christoph Simon <ciccio@prestonet.com.br> - Pela pesquisa e a gigantesca coletânea de textos sobre o Linux enviada. Eles estão sendo muito úteis tanto para mim quanto no desenvolvimento do guia.
- Gustavo Noronha <dockov@zaz.com.br> - Vem enviando freqüentes correções, contribuições construtivas ao desenvolvimento além de apoio ao desenvolvimento do guia. Vale a pena destaca-lo por sua atual dedicação junto a distribuição Debian/GNU, sua tradução e a comunidade Open Source.
- Pedro Zorzenon Neto <pzn@debian.org> - Envio de diversas atualizações para o nível Avançado, principalmente sobre o firewall iptables.
- Rafael Gomes (<@gomex>) - Muito a agradecer a ele, mas em especial por me introduzir do jeito certo ao mundo DevOps, mostrando que a colaboração e espírito de comunidade se sobressaem acima de tudo e pelo incentivo a voltar a lançar atualizações do guia.
- Andrea González - Dedicou horas revisando todo o conteúdo do guia Iniciante, o que mais impressionou foi o nível de detalhamento das correções e sugestões enviadas. Algumas passaram batidas por anos (talvez desde o início do guia).

## Marcas Registradas

Todas as marcas registradas citadas neste guia são propriedades de seus respectivos autores.

## Futuras versões

Estes são os materiais que pretendo adicionar em futuras versões do guia:

- Acrescentar mais detalhes sobre o sistema gráfico X-Window.
- Entre outros itens que venho estudando para verificar se encaixam no perfil do guia.

Esta é uma futura implementação que venho estudando para acompanhar o crescimento do guia. Sugestões são bem vindas e podem ser enviadas para <gleydson@guiafoca.org>.

## Guia do Linux

O *Guia do Linux* é uma versão fork do guia Foca Linux 6.40 WikiLivros [[http://pt.wikibooks.org/wiki/Guia\\_do\\_Linux/Capa](http://pt.wikibooks.org/wiki/Guia_do_Linux/Capa)] e *Iniciante*, *Intermediário* e *Avançado*. Ele foi portado para WikiLivros graças ao dedicado trabalho de Raylton P. Souza e outros colaboradores.

A versão *Guia do Linux* atualmente encontra-se desatualizada, requerendo o trabalho de voluntários para se igualar em conteúdo com o *Guia Foca*, pois os mesmos possui em conteúdos não sincronizados automaticamente.

Acesse e confira mais este excelente lançamento: Wikibook do Guia do Linux [[http://pt.wikibooks.org/wiki/Guia\\_do\\_Linux/Capa](http://pt.wikibooks.org/wiki/Guia_do_Linux/Capa)].

## Chave Pública PGP

Chaves PGP são usadas para criptografar arquivos, e-mails ou qualquer outra coisa que desejamos que somente uma pessoa tenha acesso. O PGP segue o padrão de chave pública/privada; a chave pública é

distribuída a todos e a chave privada permanece na posse do criador para que ele seja o único a ter acesso aos dados criptografados após digitar a "frase de acesso" correta.

Minha chave PGP segue abaixo, ela também pode ser encontrada em <http://pgp.ai.mit.edu>. Se você deseja saber mais sobre o PGP, recomendo um excelente documento encontrado na seção *Apostilas* em <http://www.cipsga.org.br/>

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
mQINBF4NBqoBEAC//7HIpbF210OrrFMBOISNNMvntdeEO4Ea4YmGF2epeQYWh+aC
8eOAP0+XrjTyZV0b4V6Q9vcTd2xw2qChHHVcp2jbVjlxjaTFTGNM8rAU1/2iFFPn
vJ29svh653MexpCzL/iDrI6F9mB+t1pQN7RwjnHRFiBcEk57SwjqIQzFhdk0W78V
ihmhNR/+wVTufy7noTfTB8MYHO3eLIzy66Ck/xQLFS+liPsXymZ5UFbbk8r43AOK
x987tnrlxLlSUMxjKPwMW/WtZTu21SoQmhEweOou0U2KedMX608/UyDQ6flzyDXz
d8MUqNKDOIR7RRJTn3fLAKhXASnQCbIC5nrA6HmUAscRvg51ErE8aXvyHZa08cRhs
2yb3Gn2/RL/mNmDxHpSmJ+iFK5ELGGqhIL7cUfvrOyPOsVt5ufEa7Poc94vBRJiv
awp178zkIefR7hOeTKRCbr0a7/7UvXc0nJ2crpWTG0/iFT607gHHH/COB7X/61Pn
Oefv9wxc365iBIFN3Pg0f04tXKHs3tUH2xhfc9/qXdVgJpme0wvJbZAZHG6piJo
0H17eijz/MWpf4o6PTJPpc9r6qBKmh9pezKnnBxNmrqgg9jms+COBwsziM7/tP22
i4Ywz17RTgd7j6EbCa3Vr6SL4HSLGcxikXpYJEFF1ZXxH+nn0y/jfZf0awARAQAB
tExHbGV5ZHNvbiBNYXppb2xpIGRhIFNpbHZhIChDaGF2ZSBNYXN0ZXIgaUGVzc29h
bCkgPGdsZXlkc29ubWF6aW9saUBnbWFpbC5jb20+iQJNBMBMBCgA3FiEEhdKAjm/j
o+PVj5Eq+T22P0ZikksFAL4NBqoCGwEEcKwKBAUVCgkIAwUWAgMBAAIeAQIXgAAK
CRD5PbY/RmKSS7LxD/9aDxE8JSAKRgUrCjRMeCXTaCoryhQAZS1/V0HzTwrzSTeC
IHJfa1Vq7NoEE5shvttwrOIMK0Unjt8fXKJYnfCtjbQsFtnOd+mGsg3H9/r1DhA
TqHrBXOUb4oxrLLyB8n1lgAET+yLEYkPnrX2wyaFHpQTyJbrUsY6Nibs69eaZlJY
40tWx/eg8RiEX/GHh9SoM8BxXgxoJSLnOU+c0/QZeEoxVX1oZXE1E6lgRe9Kvv7r
6HiGc1lrauul5SZbSBQgJK+a0I96HK329C9+utDjSKrTuu7C1SmcCWbNjC0/Xxcs
SXcalMU3oj0/O+SgOxSypahjlxgYHYPlD36IhKsgqSHc8znORz0sprCtuGiMgjxm
2ENOikf0EM/FkICIS91fCfrumEm5CBRFXS0hokPh6GUFWFghGJGAEoyi7591VKcv
254j0SytPDJzML1ETZydaIfs+3y+gxOibDo2Fn0sQ2mpFt3m8x2sAWVq/fhB5KwN
KdnFYFPwfUmerOPEXh2yamaOXEC7xMkl58x6vjTnE9uh6S1CpAqxBbrVJGMPBTQ
jWv5Hbq4ux/zuraQbAnKBdhwggHD8sxkyqwxBLunR0PK9weT/lVeam905uBJXLoZ
xfOmhbL38PZmNkKEmKUHA+JvebBdFmQxLit8DqTVS7FhaAdr2QX9cfTHw/EvI7kC
DQReDQoeARAAwCQqN7oGokcrml87uqiaEq0jpLLOfmcEUneqyE8i5ek2k4EvBUHs
8y3UIKfsAGzGvM5lLmZzQnYiUutiRZjYvVVZpImMY48Ss+8ilVw6HttjhJhwOOwn
2ZX8NED39gkdNY3XimtzyjS1DpsmixpilCRZLC103L8Uzph7kJHPSUVf/9wqLV60
n9BGj7wa7EueTqsctKof7JEUFYmmdh1KRqQktoSu+ukRN5XQoH/vOqgkUP2BsPFN
m2pLZYFRfcl5dOZeeSZRWvOUwtdG8zHCQab7Ku6AYXih/qbBpzTX2Yiy5NnW/Ezg
6cjErC8OFOAumvtov5xwPH2csv4aI3YapsRZ8xoNv+00TFgGBGm9w5Cirkriu9AR
nrxnkIFDDO/C0Cb7dG/+MpJSLV03tNlXVxBhbG0UZA/1IupdtW+0uKv58PP7Nwsvw
1c6/Ejr9S63paJTv56jGyWasUIOpZxk23iqpxyHwmX3MIOey3H/qXRhgzdJreXlW
ywnaI0bZTybsLAHrE9KwukCS2mSilPKOyGVhj/pDda0w17sUFakFtipj60HswJjN
8ssqs85n6js3/nyGky02bnJEahFmQrCbTWQ3C4Wdn9h1wXpwM+KXLaBf5xjMX/rD
z/3M3b69Y2tsJoYhpW4B3u23Fuusg891GkvPvm0bKXerfVDONg7E4xMAEQEAAYke
cgQYAQoAJhYhBIXSgI5v46Pj1Y+RKvk9tj9GYpJLBQJeDQoeAhsuBQku/geAAKAJ
EPk9tj9GYpJLwXQgBBkBCgAdFiEEuRHBS7LV8V4bMGoAghcsoyJAH2cFAL4NCh4A
CgkQghcsoyJAH2dLBg//fkh8JWdEXvd2M6RrU9V5mYLMXrzp5sCXnfjTKTLo8u3C
Soe02kTPHW2cBOzS6DL/fzUw0JTqEWXJUC3RLYmlcJ6+nx8cjh/AljaW8FZ6VMaV
BmQh36AtONBp58R17RT+2qXjq1gqhNu7XaUwQHRykhxSow3Nue07m3ZpGuWYyiGJ
ESbYw2sooBD5iHMLkYIjFQdsRKivurpF82JmD3mfif+eivINNNsI0c54Ls5ElM9a
P4KjLxkkXk+bTutZGG8JWq8FNyO+oVTTdZ1jb3TiuleVYKrum8FNZymEW6DADVf8
cAohyq94UUPbRErU7c43z5DOvcV95kDhFE/RzBEau4ry3Udz6KzFMSUMn23sY/bu
```

```
uGEugrxvtQcuoVfuNElsvivkhNEHDlTafd35PW4jVoqZAVMOkpm/ZF+pSt0Gt5We
sOoG3BVCB9tltjanY+PChyShlTKJzkk1lzIVp9PftFUL/HZhsNxok/CjTr9dYzcl
G5jKkU5Gjp+1o2s4JNK1DtURX1ifJ5Gb2DN5pA2mGHgKBPxKbnXopjBz9kJtGrul
lMRJ7QA68z3WmUUnELaoUSPINs1Al3eNzuH+8Y/CCKFxp+LLh0Hz8UL8nBEBIE+Q
aARrOY9Tk1zIX8JUqZwBI0n2rLrFyDXEeqcYbUdb9poy233Ucul/zE48i7n2bJw8
EBAAidldMFSZbrjLvUIEoUfpgN9eKhmGMsmk/j7qi8qSrvZAg8Yev7nd7scu0jEo
H2b/q73MKDuEAZss6NkEEbfe4mKKczoGUkTIUKdFarzoL1Lgd1Mmvq0M2rt34q81
8qo0CNkR34HkYcIGcUoLsr193WPTvk7P3GKQnP6MEKOigfezII3+1GzspG929J03
ayDmzCUz7hkNp/CjXD1DM+PmIqWWORbpAyYnBOLSfZVOW31xHykZcm20UknZvt7
HHaHrxEs0grt4Nmfx7Z0Z17RLfB88FoPPNbXVS1L/wjEvaIvPNURpFEArtCF10tA
iQ0nA5ia7acluQc1CybLmCMfHuhlRHZ9ZWS+184On+o0pFGySz112uq6XZnGHnLY
OEuV72g7hGNh78rUOJSM0WeZ1LaJ9iQqHSb1AbKgfJ11kQIy7css25YQKe4MjC3/
Gahdr6njOovwALKyGB2YYTVACjdp8YCMFMMoJdb9RJsbtCAkoRMbpL2rS9T63dtM
Z1ufMuR4bYhhGGpAtHNJ2DlQFzMn/V3W7bUT3Im4MnCelFeuWBBZkuZZEVgW7ce2
l2k1Ftw9QjnLCNkJE10gKZqOzR9sHBdiORB0DTF0F++dZdQqlugcjpVfcSL5Nx5a
70+F0YKM9ontVYfkr0N8uK3CQOECp6LFbIuy/pk6Ue/XQeo=
=DkuQ
-----END PGP PUBLIC KEY BLOCK-----
```