# *APOSTILA TREINAMENTO AVANÇADO EM LINUX*

V1.5


Instrutor: Rodrigo Rubira Branco

Empresa: Firewalls Security Corporation Site : www.firewalls.com.br <http://www.firewalls.com.br/>
Contatos: cursos@firewalls.com.br
Data : 15/01/2002
Duração : 40 horas

## Conteúdo do Curso:

**-ChangeLog v1.5**
**-Introdução**
**-Objetivos**
**-Listas de Discussões de Bauru**
**-Recapitulando o módulo Básico**
- **História do Linux (não será revisado)**
- **Instalação do Conectiva, do Slackware, via Rede (não será revisado)**
- **Comandos Básicos do Sistema**
  - *Man*
  - *Projeto LDP, help, info (não será revisado)*
  - *Logout*
  - *Shutdown -r now*
  - *Su*
  - *Ls (-la)*
  - *Cd (deslocamento relativo x absoluto)*
  - *Cp*
  - *Mv*
  - *Alias*
  - *Clear*
  - *Mkdir (-p)*
  - *Rmdir (-p)*
  - *Rm (-rf)*
  - *Who (w, who is god, whoami)*
  - *Df*
  - *Free*
  - *Cat /proc/cpuinfo*
  - *Setterm*
  - *Tput*
  - *Uptime*
  - *Arquivos da Inicialização*
  - *Variáveis*
  - *Variável PS1*
  - *Arquivo /etc/motd*
  - *Arquivo /etc/issue e /etc/issue.net*

- *Ps (aux)*
- *Kill*
- *Killall*
- *Sinais (1, 9, 15)*
- *Touch*
- *Find*
- *Locate*
- *Top*
- *Vi (:w :x :q :q! /)*
- *Background (&)*
- *Jobs*
- *Fg*
- *Ln (-s)*
- *Chmod (ugo +rwx)*
- *Umask*
- *Chown (user:group)*
- *Chgrp*
- *Caracteres Especiais (~,[],*,?)*
- *Home do Usuário*
- *Adicionando um Grupo*
- *Adicionando um Usuário*
- *Senhas SHADOW*
- *Arquivo /etc/passwd*
- *Arquivo /etc/group*
- *Porque saber editar manualmente usuários e grupos?*

**oCaracterísticas da Shell BASH**

**oShell Script**

- *#!/bin/bash*
- *If [ condicao ] then fi*
- *For variavel in lista do done*
- *Script Adiciona 1000 usuários*
- *Script Abrir Interface que Escolher*

**-Aprofundando-se em SHELL Script**

- *oCase*
- *oReescrevendo o script de abrir interface usando o case*
- *oWhile*
- *oO comando CUT*
- *oEscrevendo um script para acertar as permissões do HOME dos usuários*
- *oPara aprofundar: Script LogRotate.sh*

**-Utilizando o VI**

- *oEditando um texto*
- *oIndo para a última linha*
- *oIndo para a primeira linha*
- *oIndo para a enésima linha*
- *oRemovendo uma linha*

- *Removendo a partir do cursor*
- *Colocando número nas linhas*
- *Substituindo linhas*
- *Salvando*
- *Saindo e salvando*
- *Saindo sem salvar*
- *Salvando para outro arquivo*
- *Saindo e salvando em outro arquivo*
- *Forçar salvação*
- *Inserir linha abaixo do cursor*
- *Inserir linha acima do cursor*
- *Copiar e colar linhas*

**-Entendendo o sistema de arquivos do Linux**

- *Estrutura de Diretórios*
- *MOUNT*
- *FSTAB*
- *FDISK*
- *FSCK*
- *FDFORMAT*
- *MKFS*
- *MKSWAP*

**-Configuração e Instalação do Kernel**

- *Entendendo o conceito de módulos*
- *Gerenciando Módulos*
- *Adicionando Suporte a um Novo Hardware*
- *Personalizando o kernel*

**-Ferramentas Básicas de Rede**

- *Ifconfig*
- *Route*
- *Netstat*
- *Ping*
- *Traceroute*
- *Nslookup*
- *Editando os arquivos de configuração manualmente*

**-Compactadores e Empacotadores de Arquivos**

- *Gzip*
- *Zip*
- *Bzip*
- *Compress*
- *Tar*

-**Agendamento de Tarefas**
- *Cron*
- *At*

-**Entendendo o Super Daemon Inetd**
- */etc/inetd.conf*
- */etc/hosts.allow*
- */etc/hosts.deny*

-**Configurando o boot da máquina: /etc/lilo.conf**

-**Instalação e compilação de aplicativos**
- *O gcc*
- *O comando ./configure*
- *Módulos do perl*
- *O comando make*
- *O comando make install*
- *Quando ocorrem os problemas...*

- **Gerenciamento de pacotes RPM**
  - *Instalando*
  - *Removendo*
  - *Atualizando*
  - *Pesquisando*
  - *Erros Comuns*

-**Configurando o NFS**
- *Escolhendo os diretórios a compartilhar*
- *Atualizando tabela de arquivos compartilhados*
- *Portmap*
- *Acionando o Portmap*
- *Acionando o NFS*
- *Verificando arquivos compartilhados pela máquina*
- *Verificando clientes que estão acessando os arquivos compartilhados*

-**SAMBA**
- *Entendendo o Funcionamento do NetBIOS*
- *Entendendo o Funcionamento do Samba*
- *Configurando um servidor SAMBA*
- *Compartilhando Diretórios através do SAMBA*
- *Acessando arquivos compartilhados através do Linux*
- *Fazendo uma máquina Windows se autenticar no Linux*

-**Configurando o SSHD**
- *Retirando a compatibilidade com a Versão 1*
- *Retirando o uso da diretiva UseLogin*
- *Aumentando a chave criptográfica*

# ChangeLog v1.5

As mudanças da versão 1.0 para a versão 1.5 são:
-Corrigido erro pequeno na parte de While
-Incluído o comando tput
-Incluído o comando setterm
-Incluído o desafio logRotate.sh
-Incluída a parte de NFS
-Incluída a parte sobre SAMBA - 17/06/2002
-Adicionada a seção de SEGURANÇA - 17/06/2002
-Mudado um pouco o comando TOP - 17/06/2002

-Corrigido pequeno erro na parte de samba referente a valid users, que antes estava como @usuario o que na verdade poderia ser @grupo ou simplesmente usuario - 19/06/2002
-Corrigido pequena falha na parte sobre man, já que ficava um número 9 auto-colocado pelo OpenOffice como continuação da ordenação
-Corrigidos erros de acentuação - 19/06/2002
-Faltou a opção count=8139 no comando DD - 19/06/2002
-Séria incongruência na OBS do comando MOUNT foi corrigida, já que / é um ponto de montagem de uma partição de nosso HD e não nosso HD como foi dito - 19/06/2002
-Corrigidos erros de escrita - 19/06/2002
-Corrigido problema sério no comando MKFS onde se dizia que o mkfs DEVERIA suportar o sistema de arquivos a ser montado, mas quem realmente deve suportar este sistema é o kernel - 19/06/2002
-Adicionada a seção: Arquivo /etc/group - 19/06/2002
-Adicionada a seção: Porque saber editar manualmente usuários e grupos? - 19/06/2002
-Disponibilizada versões em .SDW e .TXT do documento - 19/06/2002
-Reescrita e melhorada um pouco a parte sobre SAMBA (corrigido pequeno detalhe no mapeamento de diretórios via script de logon, porque utilizava-se z:, mas este diretório é utilizado pelo windows como um temporário do netlogon e isto causava problemas) - 21/06/2002
-Reescrita a parte sobre FSTAB para correção de alguns erros - 21/06/2002

# INTRODUÇÃO

É muito importante a constante atualização tecnológica, principalmente nos dias atuais nos quais a internet se tornou não só uma realidade global, como também uma necessidade empresarial.

As instituições de ensino, sempre pioneiras no gênero UNIX e INTERNET, não devem e não podem ficar para trás nesta nova onda global: o LINUX.

Sistema operacional que surgiu e se difundiu com o apoio de UNIVERSIDADES, o LINUX vem crescendo esplendorosamente e se mostrando um sistema operacional altamente confiável e robusto, satisfazendo todas as necessidades do mercado, com exceção da falta de profissionais específicos para ele.

Devido a isto, venho com esta apostila e este treinamento (a apostila será distribuída gratuitamente na internet e eu ofereço o treinamento para qualquer empresa ou instituição que desejar) tentar reprimir a falta de profissionais e o mais importante, divulgar ainda mais o LINUX.

# OBJETIVOS

Os objetivos deste treinamento não são os de tornar seus adeptos especialistas avançados no sistema operacional, e sim usuários e administradores avançados.

Isto quer dizer que eles dominaram algumas características do sistema, mas devem estudar e se aprofundar muito para se tornarem especialistas e profissionais no mesmo.

Espero converter os usuários que leiam e façam o treinamento e mostrar-lhes quão fácil é entrar no mundo LINUX.

Percebam que a apostila é técnica, não sendo altamente instrutiva isoladamente, por isso, façam os exemplos desta e vejam os manuais para maiores entendimentos. Ela é apenas um complemento do treinamento completo, mas pode facilmente se tornar um curso através de um documento para as pessoas mais esforçadas e interessadas.

NÃO ficarei preso a uma única distribuição, mas gostaria de lembrar que pessoalmente eu utilizo o FwSec Linux, um Slackware Linux modificado e com a segurança reimplementada pela minha empresa (Firewalls Security), portanto, todos os exemplos mostrados serão em Conectiva Linux (no qual sou certificado) e Slackware Linux.

Também NÃO mencionarei o uso de Configuradores Gráficos (tais como LinuxConf e WebMin).

Os exemplos aqui apresentados são EXEMPLOS, embora as configurações funcionem podem não ser as ideais para um servidor e devem ser revistas e acrescentadas.

Esta apostila PODE e DEVE ser distribuída livremente, desde que não seja alterada.

Esta pode ser encontrada online em <http://www.firewalls.com.br/cursos/linux>.

## Listas de Discussões de Bauru

Estou tentando formular algumas listas para discussão sobre o sistema operacional Linux em Bauru.

Atualmente, existem as seguintes listas criadas:

**senac**
**cti**
**tcpdump**
**seguranca**
**linux**

Para se cadastrar acesse:

www.firewalls.com.br/mailman/listinfo/nomeDaLista <http://www.firewalls.com.br/mailman/listinfo/nomeDaLista>

Coloque seu email e uma senha e você receberá uma mensagem com instruções.

O objetivo destas listas é ter discussões de qualidade, portanto serão monitoradas e assuntos infundados serão proibidos.

## Recapitulando o módulo Básico

- **Man**

Comando que nos mostra o manual de um outro comando.
Uso: man [seção] comando
Observe que seção é opcional, mas pode ser:

1à Comandos do usuário
2à Chamadas ao sistema
3à Biblioteca de funções
4à Dispositivos
5à Formatos de arquivos
6à Jogos
7à Informações Gerais
8à Administração do sistema

Exemplo de uso:
man 1 ls
Utilize /palavra para procurar e q para sair.

- **Logout**

Fecha a shell do usuário. Este comando é utilizado quando se termina sua sessão ou para se trocar de usuário.

- **Shutdown -r now**

Comando utilizado para reiniciar a máquina. Possui alguns similares:

Reboot
Init 6
Ctrl + Alt + Del

Existe também o shutdown -h now que desliga a máquina. Seus similares:

Halt
Init 0

- **Su**

Utilizado para se trocar de usuário sem efetuar logout. Muito comum em acessos via rede, já que via rede por default o root não pode se logar.

Usa-se também su -c "comando a executar" para se executar um comando com poderes de root e depois retornar. Obviamente será pedida uma senha.

Uso do su:

su nomeDoUsuarioAseVirar

- **Ls**

Comando que serve para listar arquivos. Suas opções mais utilizadas são -la, onde o -l significa para listar as permissões (inclusive) e o -a para listar todos os arquivos (lembrando que para o linux arquivos começados com . são ocultos).

OBS: O Conectiva Linux possui um alias chamado l para o comando ls -la, use-o e caso a sua distribuição não contenha tal alias, crie-o. Veja mais adiante como fazê-lo.

- **Cd (deslocamento relativo x absoluto)**

Comando para mudar-se de diretório. O deslocamento absoluto se tem quando utilizamos a raiz (/) para indicarmos para onde queremos ir. Por exemplo, imaginemos que estamos no diretório /usr/src/linux e desejamos ir para o diretório /usr/src/teste. Temos duas opções, a seguir:

cd /usr/src/teste à Deslocamento absoluto, observe o uso do / no início do diretório para o qual queremos ir

cd ../teste à Deslocamento relativo, perceba que se estivéssemos em um outro diretório (/usr) por exemplo, não iríamos cair onde queremos. Daí a convenção de "relativo".

- **Cp**

Copia arquivos. Use: cp arquivoASerCopiado novoArquivo

Opções interessantes:

-i à Pede confirmação antes de substituir um arquivo existente

-R à Cópia recursiva. Serve para copiar diretórios e seu conteúdo.

- **Mv**

Move arquivos. Use-o também para renomear.

Uso: mv arquivo novaLocalizacao/

mv arquivo novoNome

Recomendado:

-i à Confirma antes de substituir um arquivo existente.

OBS: No Conectiva Linux existe um alias tanto para o comando cp como para o mv com a opção -i.

- **Alias**

Cria um apelido para um comando. Tem precedência sobre o comando, ou seja, pode-se criar um alias do tipo: alias ls="ls -la". Toda vez que digitarmos ls na verdade ele executará ls -la.

- **Clear**

Limpa a tela. Recomenda-se a criação de um alias chamado c para este comando.

### - Mkdir (-p)

Comando para a criação de diretórios. Usa-se o -p caso se queira criar uma "árvore" de diretórios.

### - Rmdir (-p)

Complemento do comando mkdir. Serve para remover um diretório vazio. A opção -p serve para remover uma árvore de diretórios vazia (sem arquivos).

### - Rm (-rf)

Comando utilizado para apagar arquivos. Observe que o rm simplesmente não apaga diretórios. Sua opção -r indica para apagar recursivamente, ou seja, ir apagando todos os arquivos em subdiretórios e inclusive os próprios diretórios. A opção -f força apagar, e não emite mensagens de erro caso não exista um arquivo.

Ex: rm -rf arquivoQueNaoExiste

Não acontecerá NADA. Nenhuma mensagem de erro será informada.

### - Who (w, who is god, whoami)

O commando who e w listam os usuários que estão logados na máquina. O w tem uma saída um pouco mais complexa, mostrando mais informações.

O comando who is god é uma sátira e retorna o nome de seu usuário.

O comando whoami (pode ser escrito who am i) também retorna o nome de seu usuário e é utilizado para saber com qual usuário você está logado, muito usado quando se utiliza o su e acaba se confundindo quem é você.

### - Df

Mostra informações de sistemas de arquivos montados (mesmo CDRom e Disquete).

### - Free

Mostra informações de memória (swap inclusive).

- **Cat /proc/cpuinfo**

Informações muito completas de seu processador.

- **Setterm**

Este comando serve para modificar configurações do terminal do Linux, tais como cor de fundo e cor da letra.

Ex:

setterm -background green --> Fundo Verde

setterm -foreground yellow --> Letra "amarela". O amarela está entre aspas devido ao fato de que a cor não parece ser amarelo não.

OBS: Este comando mudará a cor a partir do momento em que ele for dado, ou seja, você precisa imprimir algo na tela ou dar um clear para realmente mudar a cor.

- **Tput**

Utilizaremos este comando para posicionar o cursor na tela, onde quisermos. Ele será muito útil quando estivermos construindo shell scripts.

Ex:

tput cup 5 10 à Posiciona o cursor na linha 5 coluna 10.

- **Uptime**

Mostra a quanto tempo o sistema está ligado. Os maiores uptimes da internet são com máquinas UNIX.

- **Arquivos da Inicialização**

Alguns arquivos são executados quando o sistema reinicializa. O que nos será conveniente falar por agora será o arquivo /etc/rc.d/rc.local (todas as distribuições devem tê-lo implementado).

Este arquivo será o último a ser executado quando da inicialização do sistema.

Diversos arquivos são executados no processo de entrada de um usuário no sistema.

São eles:

.bashrc
.profile
.bash_login

Os 3 se localizam no HOME do usuário

E:

/etc/bashrc
/etc/profile

Observe que enquanto os 3 primeiros são exclusivos dos usuários (cada usuário pode ter suas configurações), os últimos 2 são globais a todos os usuários que entrarem no sistema.

Não recomenda-se alterar o arquivo /etc/profile já que este é de configurações e variáveis.

## - Variáveis

Variáveis nada mais são do que espaços na memória que armazenam valores. O linux possui variáveis do próprio sistema, que armazenam valores de configurações ou informações da máquina.

Para vê-las utilize o comando set.

Para darmos um valor a uma variável e torna-la global ao sistema, fazemos export variável=valor.

Para retirarmos uma variável fazemos unset variável.

## - Variável PS1

Esta variável guarda os valores para o PROMPT do Linux. Observe que estes valores podem ser variáveis interpretadas pela SHELL, e por default o são, ou seja, se você utilizar uma shell que não a shell BASH, eles podem ficar sem sentido.

A variável PS1 é uma variável normal do sistema, qualquer valor que for dado a ela irá ficar no prompt.

No entanto ela possui alguns valores especiais interessantes, eis alguns:

- \h à Host da máquina
- \W à Diretório Corrente
- \w à Caminha completo do diretório corrente
- \u à Nome do usuário
- \t à Hora
- \\$ à Fica $ se for usuário normal e # se for root

Exemplo:

```
export PS1="[\h@\w]\\$ "
```

## - Arquivo /etc/motd

Este arquivo é lido pelo sistema quando um usuário loga na máquina e seu conteúdo é enviado para a tela do usuário, como uma mensagem de boas vindas ou algo do tipo. Preste atenção que comandos NÃO serão interpretados.

## - Arquivo /etc/issue

Tela vista ANTES do usuário se logar. Seria a própria mensagem antes do login. Observe que o arquivo /etc/issue.net é o mesmo que o issue mas é válido para conexões via rede (telnet). Em algumas distribuições este arquivo é apenas um link para o /etc/issue.

- **Ps (aux)**

Comando que lista os processos em execução no sistema. Recomenda-se sempre utiliza-lo com as opções AUX, para que liste TODOS os processos ativos no sistema.

- **Kill**

Serve para matar um processo em execução. Deve-se utilizar um dos sinais existentes para esta tarefa. O sinal padrão é o sinal 15. Após o sinal, deve-se informar o PID (identificador único de processos) do processo que se deseja matar (encerrar).

- **Killall**

Implementação do linux muito interessante. Permite-se que se mate diversos processos com o mesmo nome de uma única vez. Observe que pode utilizá-lo para matar um único processo pelo nome, desde que se tenha o cuidado de perceber se não existem outros processos com este nome.

Ex: killall httpd
Killall -9 vi

- **Sinais (1, 9, 15)**

É importante se lembrar destes 3 sinais principais:

1-) SigHUP à Manda a aplicação reiniciar

9-) SigKILL à Manda o kernel tirar a aplicação da lista de processos ativos (mata mesmo!)

15-) SigTERM à Manda um sinal para que a aplicação termine normalmente

Os sinais são utilizados para comunicações INTER-PROCESSOS, ou seja, quando um processo deseja indicar algo para outro processo.

Neste caso, o processo kill (killall é a mesma coisa), envia o sinal que pedirmos para a aplicação.

Existem outros sinais (como SigINT) que são utilizados pelo sistema. Para visualizá-los utilize o comando kill -l.

- **Touch**

Cria um arquivo texto vazio. Muito interessante na hora de se testar alguma coisa.

Uso: touch nomeDeArquivoaCriar nomeDeArquivoaCriar2 ...

Pode-se criar diversos arquivos de uma única vez.

## - Find

Busca arquivos. Muito avançado.

Uso: find DirAProcurar opções

Exemplos de uso:

find / -name Rodrigo.tar à Procura a partir da raiz (no sistema todo) o arquivo chamado Rodrigo.tar

find /home -exec grep "teste" {} \; -exec ls -la {} \: à Procura a partir do diretório /home arquivos com o conteúdo teste (grep teste) e lista este arquivo (ls -la).

find /usr -type l -ok rm -rf {} \; à Procura no diretório /usr links (-type l) e caso encontre, confirma se deve ou não apagar (-ok rm -rf).

Consulte o manual para informações mais interessantes.

## - Locate

Busca arquivos, mas utiliza uma base de dados como padrão, o que o torna muito rápido. Cuidado!! Atualize sempre sua base de dados, ou irão aparecer arquivos que já foram removidos em suas buscas.

Outro problema do locate é o fato de que ele busca qualquer ocorrência da palavra a buscar, ou seja, se você fizer locate a, ele irá listar TUDO no sistema que contém a palavra a.

Para atualizar sua base de dados utilize: updatedb

Para buscar utilize: locate oqbuscar

## - Top

Método interessante de se vizualizar os processos ativos na máquina.

Use: M --> Ordenar por consumo de memória

P --> Ordenar por consumo de CPU

## - Vi (:w :x :q :q! /)

Ótimo editor de textos que recomenda-se e muito saber. As opções vistas foram:

- ØModo Comando, Fim de Linha e de Edição
- Ø:w à Salva arquivo
- Ø:x à Salva e sai
- Ø:q à Sai quando você não alterou nada
- Ø:q! à Sai sem salvar
- Ø/palavra à Procura palavra
- Øn à Procura pela próxima ocorrência de palavra

## - Background (&)

O linux possui uma opção interessante que é a de mandar processos para o segundo plano, liberando assim o ambiente do usuário.

Pode-se fazer isso através do sinal & após qualquer comando.

## - Jobs

Lista os processos que estão em segundo plano, retornando o número do processo de segundo plano, que deverá ser utilizado para traze-lo de volta.

## - Fg

Comando que trás de volta um processo do segundo plano.

Uso: fg numeroProcessodeSegundoPlanoRetornadoPeloJobs

## - ln (-s)

Este comando cria um link (atalho) entre diretórios e arquivos. Um link simbólico (opção -s) nada mais é do que um arquivo no HD que aponta para a área onde está o arquivo original. Se o original é apagado, o link fica "quebrado". Já um link direto (apenas ln) dá um outro nome para a mesma área do HD. Como um backup contra remoção indevida, no entanto usa-se o mesmo espaço do HD, referenciando-no de duas maneiras diferentes. Crie e compare. Um link direto não pode ser feito entre diretórios.

Uso: ln -s Original Link
ln Original Link

## - Chmod (ugo +rwx)

Comando que muda as permissões de um arquivo. Estas podem ser vistas através do comando ls -l.

A esquerda dos arquivos aparecerá uma cadeia de caracteres de difícil compreensão inicial, mas prestem atenção:

PrimeiroCaractere à Indica o tipo de arquivo, pode ser:

- \- à Arquivo normal (executável ou texto)
- d à Diretório
- c à Dispositivo orientado a caracteres (modem, portas seriais)
- b à Dispositivo orientado a blocos (hd)
- s à Socket mapeado em arquivo ("Em Unix tudo é arquivo")
- p à FIFO, comunicação inter-processos
- l à Link ***Simbólico***

3 próximos Caracteres à Permissões Válidas para o DONO do arquivo, 1ª coluna com nome de usuário.

- R à Permissão de leitura. Para diretórios, pode listar seu conteúdo.

W à Permissão de escrita.

X à Permissão de execução. Para diretórios, pode entrar nele.

3 próximos Caracteres à Permissões Válidas para o GRUPO dono de um arquivo, 2ª coluna, do lado da do DONO do arquivo. Observe que o GRUPO dono não necessariamente tem o DONO como membro.

3 próximos Caracteres à Permissões para o restante dos usuários do sistema.

Ex: chmod ugo+rwx -R arquivoOudiretório.

A opção -R manda dar a permissão recursivamente a todos os arquivos e subdiretórios deste diretório em questão.

Esclarecendo a Sintaxe UGO:

U à Refere-se as permissões para o DONO

G à Refere-se as permissões para o GRUPO DONO

O à Refere-se as permissões para o restante dos usuários

A à Refere-se as permissões para TODOS os usuários (mesmo que Ugo junto)

+rwx à Está-se dando todas as permissões

-rwx à Está-se tirando todas as permissões

OBS: Pode utilizar apenas -r ou -w sozinhos, por exemplo.

=rw à Estaria-se igualando as permissões a +RW-X, ou seja, quando se utiliza o sinal de igual, as permissões se IGUALAM as que o sinal indica, sendo retiradas as que não forem mencionadas.

## - Umask

Comando que muda a máscara de permissões padrão para a criação de arquivos e diretórios.

Seu uso será explicado mais adiante, apenas em modo OCTAL e não CARACTERE. O modo CARACTERE foi explicado no curso básico, mas não o será nesta apostila.

## - Chown (user:group)

Utilizado para mudar o DONO e o GRUPO dono de um arquivo ou diretório.

Uso: chown novodono:novogrupo arquivoOudiretorio

Observe que a opção :novogrupo pode ser omitida ou trocada por .novogrupo.

Também aqui existe a opção -R.

## - Chgrp

Utilizado para mudar apenas o grupo dono de um arquivo.

Uso: chgrp novogrupo ArquivoOuDiretório

## - Caracteres Especiais (~,[],*,?)

São também conhecidos como METACARACTERES.

Os mais comuns e utilizados são:

* à Simboliza TUDO
? à Simboliza QUALQUER CARACTERE
~ à Simboliza o HOME do usuário Corrente
[AB]* à Qualquer arquivo (*) começado com A ou com B.

## - Home do usuário

Diretório que pertence ao usuário, onde ele pode tudo. O comando cd isolado leva o usuário até este diretório. Ao logar no sistema, o usuário cai também em seu home.

## - Adicionando um Grupo

Um grupo nada mais é do que a união de diversos usuários com as mesmas características. Por exemplo, poderíamos ter um grupo estudantes ou alunos.

Para adicionarmos este grupo, devemos utilizar o comando:

groupadd alunos

No arquivo /etc/group será adicionada uma entrada alunos, e será dado um GID (identificador de grupo) a este grupo.

## - Adicionando um Usuário

Qualquer pessoa que for utilizar o linux deve necessariamente possuir um usuário válido na máquina. Lembrando que NÃO devemos utilizar o root a menos que necessário, esta tarefa é importantíssima mesmo para usuários caseiros.

Adicionando:

useradd rodrigo -g alunos

Adicionamos o usuário rodrigo no grupo alunos. Observe que a opção -g nomegrupo não se faz necessária, e caso seja omitida, teremos comportamentos diferentes em algumas distribuições:

RedHat e familiares (incluindo Conectiva):

Será criado um grupo com o mesmo nome do usuário e este será adicionado neste grupo

Slackware:

O usuário será adicionado em um grupo chamado users

Isto ocorre devido a não padronização deste ato e ao fato de um usuário NECESSARIAMENTE pertencer a algum grupo

## - Senhas SHADOW

O esquema de senhas chamado SHADOW foi criado devido ao fato de o Linux (e os Unix-Like da vida) utilizarem em suas senhas um método de criptografia chamado DES (Data Encryption Standard). Este método é fraco (utiliza chaves de apenas

64 bits) e pode ser facilmente quebrado (Veja o livro Cracking DES para entender melhor sobre este assunto). Como o arquivo /etc/passwd necessita ter permissão de leitura para todos, qualquer usuário facilmente conseguiria obter a senha de ROOT do sistema.

Com isso criaram o SHADOW, onde as senhas criptografadas com o DES ficam no arquivo /etc/shadow que só pode ser visto pelo root.

Sobra então no arquivo /etc/passwd apenas um * ou ! no lugar da senha criptografada do usuário.

Todas as distribuições linux trazem o SHADOW por padrão.

## - Arquivo /etc/passwd

Este arquivo contém os usuários cadastrados na máquina e informações sobre eles. Sua sintaxe é:

login:UID:GID:Descrição:Home:Shell

Onde:

- login à Nome do usuário na máquina
- UID à Identificador do usuário. O Linux utiliza este número para dar ou tirar permissões. Pode ser repetido entre usuários.
- GID à Identificador do grupo principal do usuário.
- Descrição à Qualquer coisa, se for omitido, deve-se deixar ::. Geralmente coloca-se nome e cargo do usuário
- Home à Diretório pessoal do usuário. Não necessariamente, mas recomenda-se que ele fique no /home e tenha o mesmo nome do usuário.
  Ex: Usuário: Rodrigo
  Home: /home/Rodrigo
- Shell à Shell que o usuário irá utilizar para se logar no sistema. Use /bin/false caso o usuário não deva logar. E /bin/bash caso deva.

OBS:

Deve ter ficado na cabeça do leitor atento o fato de CASO O USUÁRIO NÃO DEVA LOGAR.

Mas quando isto acontece?

Digamos que temos uma aplicação que deve ser executada com as permissões de um usuário. Criamos um para ela, mas este não é um usuário válido, como minha aplicação iria entrar na máquina? Este é um caso.

## - Arquivo /etc/group

Similarmente ao /etc/passwd este arquivo possui as configurações dos grupos (o /etc/passwd possui dos usuários)

Sua sintaxe geral é:

<grupo>:x:<gid>:<usuario1>,<usuario2>

Os usuários que pertencerem a este grupo estarão listados neste arquivo, a menos que o grupo seja primário do usuário, neste caso apenas estaria referenciado em /etc/passwd no campo gid.

Preste atenção que é neste arquivo que o GID dos grupos está especificado, sendo que o /etc/passwd apenas consulta ele.

- **Porque saber editar manualmente usuários e grupos?**

  Esta é uma pergunta bem simples, já que teremos de editar manualmente em diversas situações onde desejarmos modificar opções de usuários e desejarmos fazer isto de uma forma rápida e segura.

## Características da Shell BASH

O TAB completa, tanto comandos como nomes de arquivos ou diretórios, use-o.

Setas para cima e para baixo movimentam-no entre os comandos que já foram digitados.

Shift+PageUP sobe a tela.

Ctrl+D efetua LOGOFF.

Cuidado!! O Linux FAZ DIFERENÇA ENTRE MAIÚSCULAS E MINÚSCULAS.

Cuidado!! Não existe UNDELETE no Linux porque seu sistema de arquivos se auto-defragmenta durante seu uso.

## Shell Script

Um shell script nada mais é do que se utilizar diversos comandos encadeados em um arquivo. Estes comandos serão executados na ordem em que forem vistos.

Use: sh arquivo à para executar.

- **#!/bin/bash**
  - Deve ser utilizado no início dos shell script
  - Indica qual shell deverá ser utilizada para a execução
  - Não é necessário
  - Caso omitido, o script será executado usando-se a shell que o usuário estiver utilizando no momento.
- **If [ condicao ] then fi**
  - Comando condicional, caso a opção for verdadeira ele executa o que estiver entre o then e o fi

- Opcionalmente pode-se utilizar o else.
- Ex:

```
if [ $valor = "1" ]
then
    echo "Valor = 1 "
    echo " legal!"      à Ocorrerá um erro, porque as " permitem que valores dentro dela sejam interpretados. Neste caso o ! será interpretado como o operador NOT e causará um erro. Portanto deve-se utilzar ´´ que impede que qualquer coisa seja interpretada.
else
    echo ´ Valor != 1´  à Agora esta certo
fi
```

## ØFor variavel in lista do done

- Comando de loop
- A variável irá assumir a cada itereção do loop um valor da lista
- Ex:

```
for  nome in  Rodrigo Alberto Gilberto
do
    echo $nome
done
```

- Irá ter a saída:

```
Rodrigo
Alberto
Gilberto
```

## ØScript Adiciona 1000 usuários

- Cria-se um arquivo chamado /tmp/nomes.txt
- Neste arquivo coloca-se o nome dos 1000 usuários.
- Faz-se o seguinte script:

```
for  user in `cat /tmp/nomes.txt`
do
        §useradd $user
done
```

- Observe o `cat /tmp/nomes.txt` no lugar da lista. A crase faz com que o comando seja executado.
- O comando CAT lista o conteúdo de um arquivo, no caso nomes.txt que fica no lugar da LISTA.
- A variável user assume valor por valor desta lista a cada iteração e é utilizado o comando useradd para adicionar o usuário.
- Também pode ser escrito assim:

Øfor user in `cat /tmp/nomes.txt` ; do useradd $user;
done

**ØScript Abrir Interface que Escolher**

§Lembre-se que o arquivo .xinitrc localizado no home do usuário é executado assim que a interface gráfica abre.

```
#!/bin/bash
echo "Escolha a interface"
echo " 1- KDE"
echo " 2- Wmaker"
echo " 3- Gnome"
read iface
if [ $iface = "1" ]
then
        kde
        exit
 fi
if [ $iface = "2" ]
then
        echo "wmaker" > ~/.xinitrc
fi
 if [ $iface = "3" ]
 then
         echo "gnome"  > ~/.xinitrc
 fi
  startx
```

OBS: Observe o echo "gnome" > ~/.xinitrc, utilizei o metaractere ~ para indicar o home do usuário, assim independente do usuário que executar, o script funcionará.

O comando exit serve para fechar o script, porque eu não desejo que o startx seja executado caso kde tenha sido.

Coloquei um único startx ao final do script, ao invés de um por if.

Utilizei o comando read para acessar um valor digitado pelo usuário e armazenei este valor na variável iface, que não precisa ser previamente declarada.

# Aprofundando-se em SHELL Script

- **Case**

Usado quando se necessita de muitos If´s.

Funciona assim:

```
case  $Variável in
        valor) comando
                comando
```

```
                comando
                ;;
        valor2) comando
                comando
                ;;
        *)  comando
                comando
                ;;
esac
```

Exercício: Reescrever o script de abrir interface usando o case

## - While

Faz um loop que só sairá quando encontrar o comando break ou a condição for FALSA.

Sintaxe:

```
while  Condição
do
        comandos
done
```

Ex:

```
a=0
while [ $a  -le 10 ]
do
        echo $a
        expr $a + 1
done
```

Primeiro indiquei que a variável a tem valor 0.
O While testa se a variável a é menor ou igual a 10 (-le é um operador)
Se for, e é, imprime seu valor, primeiramente 0.
O comando expr soma 1 ao valor de a.
O loop é refeito, novamente se testa se a é menor que 10 e assim por diante.
Perceba o operador -le, temos outros:

- -eq à Igual
- -ne à Diferente
- -gt à Maior que
- -ge à Maior ou igual
- -lt à Menor que
- -le à Menor ou igual

## - O comando CUT

Este comando embora pouco conhecido é realmente MUITO útil. Sua

função é a de capturar apenas uma parte de uma linha, ou expressão.

Quando iríamos querer isto?

Vamos pensar, lembrem-se do arquivo /etc/passwd, nele temos diversas informações.

Porque não simplesmente capturar o nome do usuário e nada mais?

Como faríamos isto? Listar apenas o nome dos usuários do sistema?

Veja:

cat /etc/passwd |cut -f1 -d":"

Mas o que fizemos?

Primeiramente listei o conteúdo do arquivo com o cat.

Canalizei ele para o comando cut.

-f1 à Indica que quero o 1º campo. Nossa que primeiro campo?? Vou especificar com o -d

-d":" à Indiquei que o que separa cada campo é o caracterer dois pontos (:).

Exercício: Escrever um script para acertar as permissões do HOME dos usuários.

## *DESAFIO:* *Construir um script que faz o seguinte:*

*Lê um arquivo chamado LOGS.txt que possui a seguinte sintaxe e será criado a parte:*

*ArquivoDelog TAMANHO*

*Ex:*

*/log/tudo.log 10000*

*Após ler este arquivo, ele irá ser rodado via CRON (ver mais adiante) e de tempos em temos ele irá checar os arquivos que estiverem no arquivo LOGS.txt e seus respectivos tamanhos.*

*Se o tamanho for atingido ou for passado, ele irá compactar o arquivo de log, gerando um chamado nomedoarquivo.log.DIAMESANO.tar.gz.*

*No nosso exemplo, o arquivo /log/tudo.log ao atingir o tamanho 10000 irá ficar:*

*/log/tudo.log.19012002.tar.gz* à *Dia 19 de janeiro de 2002*

# Utilizando o VI

**Editando um texto:** vi nomedoarquivo
vi +linha nomedoarquivo à Abre direto na linha
**Indo para a última linha:** :$
**Indo para a primeira linha:** gg ou :1
**Indo para a enésima linha:** :n
**Removendo uma linha:** dd
**Removendo a partir da linha n até a útima:** :n,$ d
**Colocando número nas linhas:** :set number
**Tirando número das linhas:** :set nonumber
**Substituindo linhas:** :n,n2 s/palavraSerSubstituida/palavraVaiSubstituir/g
**Salvando:** :w
**Saindo e salvando:** :x
**Saindo sem salvar:** :q ou :q! caso tenha modificado
**Salvando para outro arquivo:** :w outroarquivo
**Saindo e salvando em outro arquivo:** :x outroarquivo
**Forçar salvação:** :x! à Útil quando o arquivo está como RO
**Inserir linha abaixo do cursor:** o
**Inserir linha acima do cursor:** O
**Inserir:** [INSERT] ou i
**Substituir:** [INSERT 2x] ou r
**Inserir no fim da linha:** A
**Inserir após o cursor:** a
**Copiar e colar linhas:** y à copia linha
P à cola linha

# Entendendo o sistema de arquivos do Linux

**Nome:** *ext2*
**Características:** *Pesquisa Binária*
*Não fragmentação*
*Permissões de Arquivo (podem ser extendidas)*
*Arquivos com 255 caracteres no nome*
*Qualquer caractere especial no nome*
*Por default, não síncrono*

**Nome:** *ext3*
**Características:** *Novo sistema*
*Journaling FS (assim como o do Aix e o ReiserFS)*
*Pesquisa Binária*
*Grava o que foi feito, não necessita FSCK mesmo caso caia a energia*
*Pode ser aumentado em tempo real, sem perda de dados*
*Idem a ext2 no restante*

## •Estrutura de Diretórios

É importantíssimo a qualquer administrador de sistemas entender a estrutura de diretórios do sistema. Isso porque manter a padronização definida, o ajudará a saber onde as coisas estão e a futuros administradores ou auxiliares acharem tudo no sistema.

O Linux possui uma estrutura muito bem organizada e realmente a devemos seguir.

Irei colocar cada um dos diretórios principais e o que neles devem conter:

/etc à Configurações do sistema
/lib à Bibliotecas compartilhadas necessárias ao sistema
/mnt à Montagem de discos e periféricos
/opt à Pacotes adicionais NÃO fornecidos com o sistema (não utilizado quase)
/sbin à Diretório usado na inicialização do sistema, pacotes essenciais para manutenção. Demais pacotes para a administração do sistema devem ficar em /usr/sbin ou /usr/local/sbin.

/usr à Segundo maior diretório (logo após o /), recomenda-se monta-lo como RO para evitar danos. A grande maioria dos aplicativos do sistema e instalados com ele ficam a partir deste diretório.

/var à Diretório que contém arquivos variáveis, tais como spool (filas de email, crontab, impressão) e logs. Este diretório existe para que os arquivos que necessitem ser modificados fiquem nele e não no /usr, liberando assim sua montagem como RO.

/root à Alguns Unix-Like não o utilizam (utilizam /home/root). É o diretório que contém os arquivos do administrador (seu home).

/proc à Diretório VIRTUAL onde o kernel armazena suas informações. Alguns dados deste podem ser modificados, como os abaixo de /proc/sys que contém informações muito pertinentes a performance tunning do sistema.

/tmp à Diretório que contém arquivos temporários. Todos os usuários necessitam poder escrever neste diretório (para gravar seus temporários), no entanto um não pode apagar o temporário do outro (se não teríamos um problema), devido a isto este diretório possui uma permissão especial, que possibilita escrever mas só apagar aquilo que for seu. Esta permissão chama-se STICK BIT.

/home à Os diretórios pessoais dos usuários devem ficar a partir daqui.

/bin à Aplicativos e utilitários usados durante a inicialização do sistema, antes de qualquer sistema de arquivos ser montado. As shells dos usuários costumam ficar aqui também. Binários de usuários devem ficar em /usr/bin.

/boot à Contém a imagem do kernel e tudo o que for necessário ao processo de boot, menos configurações.

/dev à Dispositivos do sistema.

- **MOUNT**

Comando utilizado para se acessar qualquer dispositivo no Linux.
Como dispositivo entenda disquete, cdrom, e o próprio HD.
Sua sintaxe é:

mount -t tipofs /dev/dispositivo /ponto/de/montagem

Ponto de montagem: Diretório que será utilizado para acessar o dispositivo.
TipoFS: Tipo do sistema de arquivos do dispositivo, geralmente:

| | |
|---|---|
| Vfat | à Fat32 |
| Msdos | à Fat16 |
| NTFS | à NTFS |
| Ext2 | à Linux |
| Iso9660 | à Cdrom |
| Umsdos | à FS especial, sistema Linux sobre FAT. |

Monta FAT32 e FAT16, além de ext2.
OBS: Nossas partições são montadas durante o boot do sistema, por isso / é um ponto de montagem que indica uma das partições de nosso HD.

## ? FSTAB

Como foi dito, alguns sistemas de arquivos devem ser montados durante o boot, para que todo o resto funcione. É o caso de nossas partições ativas do linux e a própria swap.

No arquivo /etc/fstab estão as informações do que deverá ser montado automaticamente no boot da máquina e algumas opções de dispositivos que são muito acessados e seu FS não muda (ex: cdrom).

Isto faz com que utilizemos apenas mount /ponto/de/montagem para acessar tais dispositivos, já que o resto das informações o linux busca no fstab.

Sua sintaxe geral:

*Dispositivo ponto de montagem FS opções OrdemBackup OrdemFSCK*

Onde opções podem ser:

**Auto**
**Async**
**Atime**
**Dev**
**Exec**
**Noatime**
**Noauto**
**Nodev**
**Nosuid**
**Nouser**
**Remount**
**Ro**
**Rw**
**Suid**
**Sync**
**User**
**Defaults** --> rw,suid,dev,exec,auto,nouser,async

**Ordem Backup:**

Usado pelo comando dump para fazer backup do FS.
Caso seja 0 não fará backup.

Números 1 acima serão feito backups na ordem (1 primeiro e assim por diante)

Números repetidos será feito backup do que estiver primeiro no fstab.

**Ordem FSCK:**

Usado pelo comando FSCK para checar o FS.

Caso seja 0 não será checado.

Números 1 acima serão checados na ordem (1 primeiro e assim por diante)

Em caso de números repetidos será checado primeiro, o que aparecer antes no FSTAB.

## •FDISK

Utilizado para se criar ou destruir partições. Observe que seu particionamento é SEMPRE destrutivo.

Uso: fdisk /dev/hdx à X indica o HD e não a partição

Será visto em aula suas opções:

_____
_____
_____
_____
_____
_____
_____

## •FSCK

Não se necessita passa-lo, já que o sistema o faz automaticamente quando precisar, durante o boot.

Observe que caso algum pane aconteça na passagem automaticamente, será solicitada a senha do administrador e deverá ser passado o fsck manualmente.

Digite a senha e faça:

fsck -c -v /dev/hdxy à X seu HD, Y partição que deu problema.

-c à Manda checar

-v à Informa tudo o que está fazendo e pede confirmação

## •FDFORMAT

Utilizado para formatar ***DISQUETES.***

Use: fdformat /dev/fd0.

Ele não irá criar um sistema de arquivos no disquete. Use o mkfs.

- **MKFS**

Utilizado para criar um sistema de arquivos.
Perceba a necessidade de se especificar o sistema a ser criado e o fato de que o kernel DEVE suportar este sistema.
Uso:
mkfs -t tipofs /dev/dispositivo
Onde tipofs é um dos tipos já mencionado e dispositivo pode ser qualquer dispositivo de armazenamento. Ex: fd0, hda1.

- **MKSWAP**

Utilizado para se "formatar" uma swap. Gerar seu sistema de arquivos. Utilizamos ele assim:
mkswap -c /dev/hdxy
A opção -c é opcional mas recomendada, já que checa o HD antes de criar a swap.
Após isto feito, deve-se necessariamente ativar a swap para que o sistema a reconheça.
Faça isso com o comando:
swapon /dev/hdxy
Alguns alunos mais "fuçados" já terão visto:
swapon -a
Que indica para se ativar a swap de todos os dispositivos de swap encontrados em /etc/fstab.
Lembre-se de adicionar a nova swap lá para que no próximo boot seu sistema a reconheça automaticamente.
Bem, como eu já disse o Linux é poderoso, Linux é Linux, portanto temos uma opção de caso necessitemos de mais swap, mas não tenhamos uma partição a parte para isto.
É o caso da swap em arquivo. Sua desvantagem é o fato de ser mais lenta, já que estará dentro do sistema de arquivos ext2 além do seu próprio, mas é uma maneira de se ganhar memória SEM TER QUE REINICIAR ou qualquer coisa do tipo.
Para criar este "arquivo de swap", faça os seguintes passos:
1-) dd if=/dev/zero of=/swap bs=1024 count=8139
2-) mkswap -c /swap
3-) swapon /swap
Passo1: O comando dd "copia transformando". Ele foi utilizado para se criar o arquivo de swap sem conteúdo (/dev/zero) e com um tamanho equivalente a 8139 blocos (8MB). Obviamente você pode adeqüar este tamanho para o que necessitar.

Passo2: Cria-se o sistema de swap neste arquivo
Passo3: Ativa-se a swap deste arquivo.
Use o comando free e você verá que o espaço de swap foi incrementado.

# Configuração e Instalação do Kernel

A grande característica do Linux é o fonte aberto, que facilita o desenvolvimento. Ele realmente está muito desenvolvido e constantemente são lançados muitas versões novas de seu núcleo.

Algumas destas novidades são importantes para nós e por isso desejamos atualizar nosso sistema. Mesmo que não queiramos atualizar, devemos no mínimo recompilar nosso kernel que vem no sistema para que se adapte apenas a nossa máquina, fazendo assim uma grande economia de memória e desempenho.

O primeiro passo é baixar o novo kernel.

Faça isso acessando: www.kernel.org <http://www.kernel.org/> ou em caso de pessoas de dentro da Unesp: ftp.feb.unesp.br/pub/Linux

## oEntendendo o conceito de módulos

O kernel do linux é muito bem elaborado e projetado.

Obviamente foi pensado em diversas possibilidades do uso de um sistema operacional, como por exemplo o fato de não utilizarmos o cdrom 100% dos momentos em que estamos no computador.

Se não é assim, porque o suporte ao CD deve estar o tempo todo na memória, ocupando espaço?

Com isto inventaram o conceito de módulos, ou seja, quando o sistema operacional precisa, ele carrega o código para memória, usa e descarrega.

## oGerenciando Módulos

Existem alguns comandos para o gerenciamento de módulos do sistema operacional. São eles:

modprobe -l à Lista os módulos disponíveis
modprobe módulo à Carrega o módulo na memória
insmod módulo à Carrega o módulo na memória mas não checa

lsmod à Lista os módulos carregados

## oAdicionando Suporte a um Novo Hardware

Para isto, devemos recompilar nosso kernel

## oPersonalizando o kernel (recompilando)

Após baixar o arquivo do kernel, vamos supor chamado:
linux-2.4.17.tar.bz2
Copie-o para /usr/src e entre neste diretório
Mova o kernel antigo para outro nome (mv linux linux.old)
Descompacte-o (veja compactadores nesta apostila).
Move ele para um nome mais descritivo (mv linux linux-2.4.17)
Crie um link chamado linux (alguns aplicativos precisam)
ln -s linux-2.4.17 linux
Entre no diretório dos fontes
cd linux
Inicie a configuração:
make menuconfig à Recomendado, pode ser make xconfig ou make config também
Escolha o que deseja inserir (m para módulo) e saia.
make dep
make clean
make bzImage
make install
make modules
make modules_install
Vá para a raiz (cd /)
Mova o arquivo vmlinuz (imagem do kernel) para o diretório /boot à Padrão do FS do linux
Mova o arquivo System.map para o diretório /boot também
Edite o lilo.conf para que o sistema boote pelo novo kernel
Digite lilo
PRONTO!! Reinicie sua máquina.
OBS: TODOS OS DETALHES SERÃO VISTOS EM AULA. ATENTE-SE PARA CASO NO SEU SISTEMA JÁ EXISTA O ARQUIVO /BOOT/VMLINUZ (FAÇA UM BACKUP DE SEGURANÇA E PREVINA ISTO NO LILO.CONF TAMBÉM).
O LILO.CONF SERÁ EXPLICADO MAIS ADIANTE NESTA APOSTILA.

## Ferramentas Básicas de Rede

- Ifconfig à Este comando retorna informações sobre as placas de rede (mesmo virtuais) e a interface lo.
  Ele pode ser utilizado para mudar informações de rede assim:
  ifconfig interface ip netmask mascaraREDE
  Ex:
  ifconfig eth0 192.168.0.1 netmask 255.255.255.0
  ifconfig eth0:0 192.168.0.2 netmask 255.255.255.0
  Primeiro, dizemos que o ip de nossa placa será 192.168.0.1 classe C.
  Depois, configuramos um ip virtual para esta placa também classe C.
- Route à Comando que gerencia a tabela de roteamento estática do Linux. Apenas iremos utilizar este comando (em casos básicos, claro, roteamentos mais avançados podem ser feitos, lembrando que será um roteamento estático e limitado. Veja: Advanced Routing HOWTO para entender melhor o uso do iproute) para definir o gateway de nossa máquina:
  - route add default gw IPdoGATEWAY
  - Use: route -n para listar a tabela de roteamento.
- Netstat à Lista conexões ativas. Use netstat -na para listar todas as conexões ativas e que estão escutando.
- Ping à Famoso comando que utiliza o protocolo ICMP para verificar o delay de uma máquina a outra na rede.
- Traceroute à Comando muito utilizado para se verificar o caminho efetuado por um pacote de uma máquina a outra na rede.
- Nslookup à Retorna o nome da máquina a partir de seu número IP.
  Use: nslookup IP
  Uma opção interessante seria:
  nslookup -type=mx ip à Retorna o servidor de email deste domínio. Veja manual.

# Editando os arquivos de configuração manualmente

Esta parte do treinamento se torna um pouco mais complicada, devido ao fato de as muitas distribuições existentes trabalharem diferentemente com os arquivos de configuração.

No Slackware esta tarefa é muito mais simples, bastando editar-se o arquivo /etc/rc.d/rc.inet1. Neste arquivo, existem variáveis com nomes bem demonstrativos que bastam ser colocados seus valores corretos para trabalhar perfeitamente. Preste atenção que estamos ensinando apenas com uma única interface de rede, não cabendo a este treinamento aprofundar-se no uso de duas. Caso seja da sua necessidade, por favor mande um email para duvidas@firewalls.com.br <mailto:duvidas@firewalls.com.br> que não exitaremos em auxiliar-lhe.

No Conectiva Linux:

Edite o arquivo /etc/sysconfig/network à Variáveis bem demonstrativas
Edite o arquivo /etc/sysconfig/network-scripts/ifcfg-eth0 à Variáveis bem demonstrativas.

Estes dois scripts servem para setarmos o IP, GATEWAY e NOME da máquina. Observe que ifcfg-eth0 se trata da primeira interface, se tivéssemos um ifcfg-eth1 seria a segunda e assim por diante.
Podemos também setar uma virtual com ifcfg-eth0:0.
Caso queira setar um ip virtual, apenas copie o ifcfg-eth0 para ifcfg-eth0:0 e edite suas opções.
Após as alterações, faça ifdown eth0 e ifup eth0 para que elas sejam validadas.

## Agora as opções são válidas para todas as distribuições:

**Arquivo /etc/hosts**

Este arquivo contém os nomes de hosts conhecidos da máquina e que não seriam necessários checar o servidor de DNS.

É muito útil. Sua sintaxe é:

IP FQDN APELIDO

Para quem não sabe, FQDN seria o nome completo da máquina, com a seguinte sintaxe:

NOMEMÁQUINA.NOMEDOMÍNIO.DOMÍNIOPRIMEIRAGRANDEZA.

Ex:

www.firewalls.com.br <http://www.firewalls.com.br/>
| | | |
| | DOMÍNIO PRIMEIRA GRANDEZA
| |

| NOME
| DOMÍNIO
NOME
MÁQUINA

Um exemplo da sintaxe do arquivo /etc/hosts seria:

200.158.118.169 www.firewalls.com.br <http://www.firewalls.com.br/>
web

Quando fizermos por exemplo, ping web, automaticamente minha máquina saberá que WEB seria a máquina com ip 200.158.118.169.

**Arquivo /etc/resolv.conf**

Este arquivo contém o nome do domínio que nossa máquina pertence e os servidores DNS de nossa máquina.

Um exemplo deste arquivo seria:

domain firewalls.com.br à Domínio firewalls.com.br
nameserver 200.158.118.169 à DNS primário
nameserver 200.206.116.2 à DNS secondário
nameserver 200.145.150.1 à DNS terciário
nameserver 200.145.160.1 à DNS quaternário

**Arquivo /etc/host.conf**

Define a ordem de resolução de um HOSTNAME, ou seja, se primeiro o sistema deve procurar o nome de hosts no arquivo /etc/hosts ou se deve requisitar ao servidor de DNS.

Um exemplo deste arquivo:

order hosts,bind à Primeiro vê o arquivo hosts depois o DNS
multi on à Permite mais de um endereço por host no arquivo /etc/hosts

**Adição de ROTAS**

Recomendo SEMPRE utilizar o comando route.

**Configurando sua placa de REDE**

Obviamente este seria o primeiro passo, mas como ele é genérico a todas as distribuições quis coloca-lo aqui primeiro, onde o aluno já estará familiarizado com as configurações de rede do linux.

1-) Adicione suporte a sua placa de rede no kernel (recompilar), geralmente adicionamos como módulo, mas não necessariamente.

2-) Edite o arquivo /etc/modules.conf. Este arquivo contém configurações lidas antes de se carregar qualquer módulo no kernel.

3-) Coloque a seguinte linha:

alias eth0 nomemódulo

Para a segunda placa, coloque alias eth1 nomemódulo e assim por diante para outras placas de rede.

Perceba que o nome do módulo é apenas seu nome, sem extensão .o ou seu caminho no sistema.

Um exemplo seria:

alias eth0 dmfe
alias eth1 rtl8139

Ok, tudo pronto, agora basta testar.

Primeiramente de um ping em seu GATEWAY (definido com o comando route).

Caso não responda cheque novamente as rotas, digite ifconfig para ver se a placa de rede está no ar.

Se tudo parecer correto mas continuar sem responder, verifique o cabo de rede.

# Compactadores e Empacotadores de Arquivos

Tópico importantíssimo para a utilização de qualquer sistema operacional, compactadores e empacotadores são SEMPRE utilizados na internet para diminuir o tamanho de arquivos (transferências mais rápidas) e unir diversos arquivos em um único.

Compactador à Diminui o tamanho de um arquivo. Arquivos texto são mais facilmente compactados e tendem a diminuir entre 60 e 70% de seu tamanho.

Empacotador à Une diversos arquivos ou diretórios em um único. Os usuários não lidam muito com este termo, já que no sistema operacional concorrente os compactadores são também empacotadores.

**oGzip**

Ótimo compactador.

Para utilizar:

gzip arquivoAcompactar

O arquivo será substituído pelo mesmo nome mas com extensão .gz, que indica que ele foi compactado com o gzip.

Para descompactar:

gzip -d arquivoAdescompactar.gz

gunzip arquivoAdescompactar.gz

As duas opções acima são idênticas.

### oZip

Pouco utilizado no mundo Linux, é um compactador/empacotador.

Utilize zip arquivo.zip Arquivo1 Arquivo2.

Cuidado ao zipar diretórios!!

Para dezipar, use:

unzip arquivo.zip

### oBzip

Compactador mais atual e mais eficiente que o GZIP. Não está sendo muito utilizado e atualmente está na versão 2.

Use:

bzip2 arquivoAcompactar

Este arquivo será substituído por um com o mesmo nome e extensão .bz2.

Para descompactar:

bunzip2 arquivoAdescompactar.bz2

bzip2 -d arquivoAdescompactar.bz2

As 2 opções são iguais.

### oCompress

Muito antigo, mas ainda hoje encontramos uso deste compactador.

Para compactar:

compress arquivoAcompactar

Será gerado um arquivo com a extensão .Z

Para descompactar:

Uncompress arquivoAdescompactar.Z

### oTar

Praticamente um MITO do mundo UNIX. Este excelente empacotador serve para a criação de BACKUPs, tendo muitas e muitas opções. Veremos aqui as mais importantes, tais como as opções de criação de um arquivo empacotado e já compactado (utilizando gzip, bzip2 ou compress).

Criando um arquivo .tar:

tar cvf nome.tar ArquivosouDiretoriosATARGIAR

Voltando um arquivo .tar:

tar xvf nome.tar

Criando um arquivo .tar.gz:

tar zcvf nome.tar.gz ArquivosouDiretoriosATARGZIPAR

Voltando um arquivo .tar.gz:

tar zxvf nome.tar.gz

Criando um arquivo .tar.bz2:

tar Icvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Conectiva 6
tar jcvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Conectiva 7
tar ycvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Slackware

Voltando um arquivo .tar.bz2:

tar Ixvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Conectiva 6
tar jxvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Conectiva 7
tar yxvf nome.tar.bz2 ArquivosouDiretoriosATARBZIPAR à Slackware

Existem muitas outras opções do TAR. Veja o manual para maiores instruções.

# Agendamento de Tarefas

O agendamento de tarefas faz parte do dia-a-dia dos administradores de sistemas. Sempre é necessário automatizar as tarefas. Isto nós aprendemos através da criação de scripts. Mas nossos scripts muitas vezes necessitam ser executados de tempos em tempos. Como fazer isso? Programando sua execução. Utilize o cron e o at para isto.

## oCron

Agenda tarefas a serem executadas sempre. Isto quer dizer, por exemplo, tarefas a serem executadas todo dia a meia noite por exemplo. Ou todo dia 10 as 10:00 horas e assim por diante.

O cron permite o agendamento de tarefas por usuário. Cada um pode ter suas tarefas e estas serão executadas com as permissões deste usuário.

Comumentemente agendaremos tarefas como root.

Agendando uma tarefa:

crontab -e

Será aberto o editor vi, editando o arquivo correto. Ali dentro, siga a seguinte sintaxe (uma por linha):

MinutoHoraDiadomesMesDiadasemana comando

Ex:

12345600***LsDentro do crontab deve-se colocar a segunda linha da tabela acima. A primeira linha será utilizada para a descrição dos campos, a seguir:

CampoValorIntervalo1Minutos0 a 592Hora0 a 233Dia do mês1 a 314Mês1 a 125Dia da semana0 a 6 (0=domingo)6Comando a ser executado-No nosso exemplo, o comando ls será executado exatamente a meia noite em ponto (0 horas e 0 minutos) todos os dias. O * significa todos os valores possíveis.

### oAt

Agendamento de tarefas a serem executadas uma única vez. Por exemplo, uma tarefa que necessite ser executada hoje a meia noite.

Uso:

at hora -f arquivo

Ex:

at midnight -f /root/Instalados/logrotate.sh

Utilize o atq para visualizar as tarefas agendadas e atrm numeroDatarefa para desagendar uma tarefa.

## Entendendo o Super Daemon Inetd

A função do inetd é a de economizar memória. Como? Imaginemos um servidor ftp que não recebe muitas requisições, digamos umas 5 por dia, por exemplo.

Este servidor ftp fica o tempo todo na memória da máquina, ocupando um espaço que não é necessário. No entanto, poderia haver alguma maneira de fazer com que este serviço não ficasse no ar o tempo todo, apenas quando necessário.

Daí surgiu o inetd. Ele fica no ar, esperando conexões para os mais diversos serviços (configurados em /etc/inetd.conf). Quando a conexão chega, ele apenas repassa para o serviço específico.

Infelizmente, como inconveniente temos o fato de que será necessário carregar o serviço para a memória, o que torna o processo mais lento.

### o/etc/inetd.conf

Sua sintaxe é:

ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a

Explicando os campos:

ftp à Serviço que escuta na porta 21 (ver /etc/services)

stream à Modo para conexões TCP (dgram para udp)

tcp à Protocolo de conexão

nowait à Não ficará esperando

root à Rodará como usuário root

/usr/bin/tcpd in.ftpd -l -a à Programa a ser chamado e seus parâmetros

### o/etc/hosts.allow

Observe no exemplo acima, que o programa a ser chamado é TCPD. Este é o programa que chamamos de TCP Wrappers. Como parâmetro é passado o serviço a ser protegido, no caso in.ftpd.

TCP Wrappers implementa um controle de acesso aos serviços que rodam sob o inetd através do número ip ou nome da máquina.

O arquivo /etc/hosts.allow contém os serviços que serão permitidos. Caso esteja sendo permitido, ele nem sequer irá checar o que está sendo negado.

Sua sintaxe é:

Serviço:IP

Ex:

in.ftpd:localhost à Libera localhost a conectar no serviço FTP

All:200.158.118.169 à Libera tudo para a máquina 200.158.118.169

### o/etc/hosts.deny

Neste arquivo proibimos os serviços. Sua sintaxe é igual a do arquivo /etc/hosts.allow.

## Configurando o boot da máquina: /etc/lilo.conf

Lilo (LInux LOader) é o gerenciador de boot mais utilizado e antigo no sistema operacional Linux.

Atualmente temos também ganhando muito prestígio o chamado GRUB.

Um exemplo de sua configuração seria:

boot=/dev/hda à HD em que o LILO será instalado

map=/boot/map à Mapa do HD
install=/boot/boot.b à Geometria do HD
prompt à Exibir o prompt do lilo
timeout=50 à 5 segundos de espera
image=/boot/vmlinuz-FwSec-1.0 à Imagem do kernel a ser carregada
label=linux à Quando for escolhido o nome linux
root=/dev/hdb1 à Partição raiz = /dev/hdb1
read-only à Monta-se como RO
password=teste à Necessita-se digitar a senha TESTE
other=/dev/hda3 à Na partição /dev/hda3
label=outroSO à Será carregado caso outroSO seja escolhido

# Instalação e compilação de aplicativos

No Linux a grande maioria dos aplicativos serão encontrados em seu estado natural, ou seja, em código fonte, geralmente na linguagem C.

Nesta parte do treinamento aprenderemos como compilar tais aplicativos e torná-los executáveis.

Sempre leia o arquivo README e/ou INSTALL.

Geralmente fazemos apenas:

```
./configure
make
make install
```

## oO gcc

Em minha opinião o melhor compilador C existente. O kernel do Linux foi feito para ser compilado no compilador C da GNU (o gcc).

Ele possui diversas características importantíssimas:

- Total compatibilidade com o ANSI C
- Recursos de otimização de código automáticos

Para compilar um programa em C, use:

```
gcc arquivo.c -o arquivoBinárioAserGerado
```

## oO comando ./configure

A grande maioria dos aplicativos possui uma opção de configuração que se trata do comando ./configure. Este comando faz com que a aplicação verifique todas as necessidades para a compilação de um programa, tais como bibliotecas e outros aplicativos e

o sistema operacional em uso. Geralmente podemos utilizar ./configure -prefix=/dir/a/ser/instalado para especificar um diretório onde a aplicação deverá ser instalada.

Observe atentamente possíveis erros aqui, como a falta de bibliotecas instaladas no sistema.

Quando falta uma biblioteca, devemos instalá-la antes de prosseguir.

Para isto, basta acessar www.freshmeat.net <http://www.freshmeat.net/> e procurar pela biblioteca faltante.

Ao baixá-la, descompactá-la e entrar no diretório, dando ./configure.

Após isto, make e make install.

Verificar se o diretório em que a biblioteca foi instalada está em /etc/ld.so.config, caso não esteja, inserir.

Após isto, digite ldconfig.

## oMódulos do perl

Perl é uma grande linguagem de programação e muitas aplicações necessitaram dele. Ele funciona modularizado (assim como o kernel do Linux). Para instalar um novo módulo, entre no diretório do mesmo e digite:

perl Makefile.pl
make
make install

## oO comando make

O comando make lê as instruções de um arquivo Makefile no diretório corrente e através destas, compila um programa com códigos separados e com as opções do compilador que se fazem necessárias. Após o ./configure a grande maioria das aplicações pedem o make.

## oO comando make install

Este comando apenas copia os binários gerados pelo comando make para seus respectivos locais dentro do sistema.

## oQuando ocorrem os problemas...

Problemas ocorrem quando estamos fazendo algo errado. Tem certeza que leu o arquivo README e INSTALL e lá não fala nada de diferente para a instalação??

Sim?

Ok, então verifique as dependências, ou seja, verifique se não faltam aplicativos ou bibliotecas, lembre-se de sempre ver as mensagens de erro que aparecem, tanto no configure quanto no make.

Dificilmente uma aplicação conterá erros e estará disponível para download, mas muitas vezes será necessário modificar alguma opção não compatível com o seu sistema. Verifique os erros de compilação e baseado em conhecimento de programação (caso você não tenha, ou peça para quem tem ou tente baixar outra versão ou já binário) tente arrumar.

## oGerenciamento de pacotes RPM

Os pacotes RPM foram inventados pela RedHat Linux e fazem parte do projeto de desmistificação do Linux, tornando-o um sistema operacional mais facilmente gerenciável e acessível a usuários caseiros.

Os pacotes possuem extensão .RPM e são pré-compilados (não estamos falando dos SRPMs).

### §Instalando

Para instalar um pacote RPM faça:

```
rpm -ivh pacote.rpm
```

### §Removendo

Para remover um pacote RPM faça:

```
rpm -e nomedopacote
```

### §Atualizando

Para atualizar um pacote RPM faça:

```
rpm -uvh pacote.rpm
```

### §Pesquisando

Para pesquisar se determinado pacote RPM está instalado faça:

```
rpm -qa |grep NOME
```

A opção -qa lista todos os instalados e o |grep procura pelo nome, já que geralmente não sabemos o nome, senão poderíamos utilizar diretamente rpm -q nome.

A saída do rpm -qa é o que deve ser utilizado pelo rpm -e quando se deseja excluir um pacote

### §Erros Comuns

*Falha nas Dependências*:

Falta algum aplicativo que este precisa para instalar. Caso você tenha instalado tal aplicativo, mas o sistema não esteja reconhecendo (você não instalou como RPM), use a opção -nodeps.

*Pacote mais atualizado já instalado*:

Quando você tenta instalar um pacote mais antigo do que um já instalado, utilize a opção -oldpackage para forçar a instalação.

*Pacote já instalado:*

Quando você deseja instalar um pacote já instalado (talvez o que esteja instalado esteja com problemas), utilize a opção -replacepkgs.

*Arquivos conflitam:*

Isto acontece quando um pacote que está sendo instalado irá substituir um arquivo existente. Nestas ocasiões use a opção -replacefiles.

*Opção -force:*

Une -oldpackage + --replacepkgs + --replacefiles.

## Configurando o NFS

Network File System (NFS) consiste em um serviço muito utilizado no mundo UNIX e obviamente portado para o Linux para o compartilhamento de arquivos em rede.

Ele permite se acessar uma máquina remota, montando-a em sua própria como um dispositivo qualquer, permitindo assim um acesso simples, como a um diretório local.

### oEscolhendo os diretórios a compartilhar

Os diretórios que desejarmos compartilhar devem ser mencionados no arquivo /etc/exports.

A sintaxe deste arquivo segue o seguinte exemplo:

/home/focker 192.168.0.*(ro)

Primeiramente temos o diretório a ser compartilhado: /home/focker

Máquinas que podem acessar o compartilhamento:192.168.0.* --> ClasseC

Permissões de acesso: ro --> Somente Leitura

Observe que por mais que você compartilhe com permissões de leitura e escrita (rw) é necessário que no sistema de arquivos local, tal permissão também seja dada a todos no sistema, isto devido ao fato de que as permissões do sistema local sobressaem-se sobre as do NFS.

oAtualizando tabela de arquivos compartilhados

Para atualizar a tabela de arquivos compartilhados devemos utilizar o comando exportfs -a.

- Portmap

Serviço necessário para o bom funcionamento do NFS e de qualquer serviço baseado em RPC (Remote Procedure Call).

RPC foi inventado pela Sun para que serviços não utilizassem uma porta fixa para rodar.

Devido ao fato de não rodar em uma porta fixa, faz-se necessário que algum serviço indique aos clientes que tentem acessar em qual porta o servidor está rodando, esta é a função do Portmap.

- Acionando o Portmap

Para ativar o portmap basta chamá-lo na linha de comando e no caso do Conectiva ainda tem-se a opção:

```
cd /etc/rc.d/init.d
./portmap start
```

- Acionando o NFS

Após todas estas configurações feitas, deve-se acionar o NFS. Chame-o na linha de comandos (nfsd).

No Conectiva:

```
cd /etc/rc.d/init.d
./portmap start
```

- Verificando arquivos compartilhados pela máquina

Para verificar quais arquivos estão sendo compartilhados por uma máquina, faça:

```
showmount -e IP
```

No caso de você estar na máquina que você quer ver, apenas faça showmount -e.

- Verificando clientes que estão acessando os arquivos compartilhados

**Para se verificar quais clientes estão acessando a máquina, faça:**

```
showmount IP
```

Caso você esteja na máquina, faça apenas showmount.

# SAMBA

Samba como todos devem saber é um sistema integrado para Linux que faz com que este suporte o protocolo NetBEUI, classicamente utilizado pelos produtos Microsoft

## Entendendo o Funcionamento do NetBIOS

Os aplicativos de impressão e compartilhamento de arquivos são baseados no NetBIOS (network basic input output system). O BIOS define a interface de aplicações para solicitar serviços de I/O no DOS. O NetBIOS extende esta interface até os serviços de rede.

A evolução do NetBIOS fez com que funcionasse sobre o TCP/IP e fosse dividido em em partes.

Não nos aprofundaremos no estudo deste protocolo e sim como fazer com que o Linux se comunique com as máquinhas Windows em uma rede.

Entendendo o Funcionamento do Samba

É muito importante lembrar que o programa se chama SAMBA devido ao nome dado a uma das partes do NetBIOS: SMB

Para entendermos melhor o funcionamento do Samba, precisaremos saber um pouco mais sobre o NetBIOS.

Ele precisa mapear os nomes dos computadores NetBIOS para um endereço IP, para isto, o NetBIOS possui 3 modos:

*Via Broadcast*
*Via lmhosts*
*Via um servidor de Nomes NetBIOS*

## Configurando um servidor SAMBA

Neste treinamento não iremos nos aprofundar no funcionamento de servidores de nomes NetBIOS, nem em opções mais avançadas do SAMBA, nos prendendo mais a parte de compartilhamento de arquivos propriamente dita.

A configuração do Samba é muito simples e se baseia exclusivamente em um arquivo de configuração: smb.conf

Vejamos aqui um exemplo da seção principal deste arquivo:

*[global]* --> Indica as configurações válidas para todos
*hosts allow = 192.168.0.* --> A rede 192.168.0.0 tem permissão de acessar
*workgroup = windows* --> Nome do Workgroup que deve ser colocado nas estações
*netbios name = server* --> Nome NETBIOS do Servidor SAMBA
*server string = Firewalls Samba Server* --> String de Apresentação
*security = share* --> Especifica que a segurança esta no compartilhamento

*log file = /log/samba/%m.log* --> Será criado um arquivo de log com o nome da máquina.log para cada máquina que acessar o servidor samba. Este arquivo será criado em /log/samba. Observar que %m é uma variável. Ela

que será substituída pelo nome da máquina que
estiver acessando o compartilhamento.
*guest account = nobody* --> Permitirá acesso a usuários não autenticados

## Compartilhando Diretórios através do SAMBA

Como pudemos perceber na seção global de nosso arquivo smb.conf, não está sendo especificado nenhum diretório que será compartilhado.

Iremos fazer isso agora:

| | |
|---|---|
| *[public]* | --> Nome do compartilhamento |
| *comment = Espaco Publico* | --> Comentário para este compartilhamento |
| *brownseable = yes* | --> Poderá ser visto por todos |
| *create mode = 0777* | --> Ao ser criado, a permissão será 0777 |
| *directory mode = 0777* | --> Mesmo para as permissõe de diretório |
| *path = /home/Backups* | --> Localização do diretório compartilhado no sistema |
| *public = yes* | --> Este diretório tem acesso público, todos podem acessá-lo |
| *only guest = yes* | --> Usuários não autenticados irão acessar diretório |
| *read only = no* | --> Não está somente para leitura |

Esta configuração acima é um exemplo sutil de um compartilhamento aberto para todos os indivíduos que desejarem acessar.

Agora, caso se deseje algo mais restrito, poderíamos ter compartilhamentos baseados em usuários (usuário + senha):

| | |
|---|---|
| *[rodrigo]* | --> Nome do compartilhamento |
| *comment = Diretorio Rodrigo* | --> Comentário |
| *brownseable = yes* | --> Poderá ser visto |
| *create mode = 0770* | --> Máscara da criação |
| *directory mode = 0770* | --> Máscara de diretórios |
| *path = /home/rodrigo* | --> Localização na máquina |
| *valid users = @rodrigo* | --> Usuários que podem acessar |
| *read only = no* | --> Pode-se escrever |

Existem diversas configurações avançadas que se podem fazer através do samba, que serão vistas mais adiantes neste treinamento.

Obviamente devido a escassez do tempo, muitas delas serão deixadas para que o aluno aprenda sozinho. O principal do caminho lhes será fornecido.

## Acessando arquivos compartilhados através do Linux

Obviamente os usuários Linux também irão querer acessar dados compartilhados pelas máquinas Windows (ou servidores SAMBA).

Nestes casos, temos três opções:

*Utilizar o SMBClient*

*Utilizar o SMBMount*

*Utilizar o comando mount -t smbfs*

A terceira opção chama a segunda quando executada no sistema, portanto não iremos explicá-la.

Nenhuma das duas opções necessitam configurações da máquina Linux que acessará os compartilhamentos.

Basta fazer:

*smbclient \\\\servidor\\compartilhamento -U <usuario>*

ou

*smbmount \\\\servidor\\compartilhamento <ponto de montagem> username=<usuario>*

A grande diferença entre as duas opções anteriores está no fato do SMBMOUNT criar um ponto de montagem para o compartilhamento remoto, permitindo assim que o acessemos como um diretório local, enquanto o smbclient possui um cliente específico.

## Fazendo uma máquina Windows se autenticar no Linux

Chegamos a parte mais complexa de nosso pequeno estudo sobre o SAMBA.

Devido a grande demanda de serviços pedido para diversos consultores do mercado, relativos a utilização de um servidor SAMBA (em linux) para fazer a autenticação e compartilhamento dos dados para uma rede windows, estaremos ensinando como executar tal serviço.

Nesta etapa do treinamento, seu instrutor e você já montaram um servidor SAMBA em Linux, o acessaram através de outro Linux, e com máquinas no mesmo workgroup, trocaram arquivos entre Windows e Linux.

Agora iremos para a parte mais interessante, como fazer estas máquinas Windows se autenticarem e executarem scripts de LOGON que estão no nosso sistema Linux.

A primeira etapa envolverá a configuração dos clientes Windows:

- Entre no Registro
- Siga até o seguinte item:

*/HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/VxD/VNETSUP*

- Crie uma chave com o nome: *EnablePlainTextPassword*
- E com conteúdo: *1*

Isto foi feito para habilitar senhas em texto puro no sistema Windows, já que ele utiliza um esquema de criptografia diferente de nossas máquinas Linux.

Obviamente isto gera um certo grau de periculosidade em redes, portanto recomendamos fortemente o estudo de implementações SAMBA com criptografia habilitada.

- Agora, nas propriedades do ambiente de rede, escolha a opção de Cliente de Redes Microsoft.

- Na aba apropriada escolha a opção de efetuar logon em um domínio NT.
- Coloque o nome do Workgroup.

Pronto, sua máquina Windows está pronta, vamos para a parte mais gostosa, configurar nosso servidor Linux!

Criemos o compartilhamento NetLogon, onde ficarão nossos scripts de logon:

```
[netlogon]
  comment = Network Logon Service
  path = /home/Samba/netlogon
  public = no
  writable = no
  browsable = no
```

Feito isso, iremos ter de alterar e acrescentar algumas opções em nossa seção global do arquivo smb.conf:

```
domain logons = yes                          --> Indica que será servidor de LOGON
security = user                              --> Segurança para usuários
domain master = yes                          --> Será o mestre do domínio
password level = 5                           --> Senhas de no mínimo 5 dígitos
username level = 5                           --> Usuários com no mínimo 5 letras
smb passwd file = /home/Samba/private/smbpasswd --> Arquivo de senhas
logon script = %m.bat                        --> Script de logon
```

Vejamos alguns detalhes deste arquivo acima.

Estamos especificando um arquivo de senhas, portanto devemos criá-lo, mas antes temos de saber que o usuário a ser criado no SAMBA deve existir no sistema.

Para criarmos tal usuário, fazemos:

*smbpasswd -a <usuario>*

Será solicitada a senha do usuário.
Caso se deseje modificar, use smbpasswd <usuario> (sem a opção -a)

Como podemos perceber a senha para o sistema e para o SAMBA será diferente.

Após termos criado este arquivo de senhas, devemos nos atentar para a TAG logon script = %m.bat

Ela irá procurar o script de logon no diretório netlogon que criamos no sistema.

O arquivo terá o <nomeDaMáquina se logando>.bat

Obviamente a extensão .bat foi mantida apenas para fins de facilitar a visualização, o Linux não fará diferença para isto.

Poderíamos também substituir o %m por %u, especificando assim um script de logon por usuário específico.

Claro, também poderíamos simplesmente colocar o nome do script, sendo este único para todos.

Vamos agora ver um exemplo de script de logon:

vi scriptLogon.bat:

*net time \\192.168.0.1 /set /yes*
*net use r: \\192.168.0.1\public*

**Mas o que é isso?**

Estes comandos serão executados assim que um usuário se logar em nosso servidor, ou seja, o primeiro comando apenas acerta a hora da máquina do usuário que está se logando com a hora de nosso servidor, facilitando assim o monitoramento das atividades dos usuários.

O segundo comando mapeia o compartilhamento público que criamos para o R: do usuário que está se autenticando.

# Configurando o SSHD

O SSH veio para substituir as fraquezas encontradas com o TELNET. Ele fornece recursos para administração remota criptografada.

Seu arquivo de configuração comumente fica em /etc/ssh/sshd_config e para ativá-lo basta digitarmos sshd, já que ele vem pré-configurado.

O servidor SSH lê os arquivos de configuração /etc/hosts.allow e /etc/hosts.deny assim como o TCP Wrappers.

## oRetirando a compatibilidade com a Versão 1

É importante atentarmos para isto, já que tal compatibilidade possuía uma vulnerabilidade em algumas versões do OpenSSH.

Mude a linha:

Protocol 2,1 do arquivo de configuração para Protocol 2

## oRetirando o uso da diretiva UseLogin

A diretiva UseLogin também possuía vulnerabilidade em algumas versões do OpenSSH, portanto convém desabilitá-la (atente que esta vem desabilitada por padrão).

Para isto, basta alterar:

UseLogin yes para UseLogin no

### oAumentando a chave criptográfica

Você pode querer aumentar o tamanho da chave criptográfica a ser negociada com o servidor. Para isto basta alterar a diretiva:
ServerKeyBits 624
Para:
ServerKeyBits 2048 por exemplo, como eu costumo recomendar.
Deixe pelo menos 1024 bits de chave criptográfica.

### oPermitindo ao root se logar remotamente

Isto não é nem um pouco recomendado, mas não sei por qual motivo insano você pode necessitar disto, portanto ensinarei aqui.

Lembre-se você pode logar-se remotamente com um usuário e utilizar o su para se tornar root.

Caso deseje se logar diretamente como root, use a diretiva:
PermitRootLogin yes

## Configurando o ProFTPd

Eu recomendo fortemente o uso do ProFTPd sobre o WU FTPD (padrão na maioria das distribuições, não no Slackware 8.0).

Esta recomendação é devida ao fato do servidor Wu FTPD possuir muitas vulnerabilidades e um histórico de falhas muito grande e o proftpd ter sido feito com grande ênfase em segurança.

Abaixo coloco um exemplo de um arquivo de configuração do proftpd e explico suas diretivas mais importantes.

Preste atenção que o proftpd possui MUITOS recursos. Para conhecer todos eles, acesse o site www.proftpd.org <http://www.proftpd.org/> e veja o manual.

O proftpd lê os arquivos /etc/hosts.allow e /etc/hosts.deny assim como o TCP Wrappers.

**Arquivo: /etc/proftpd.conf**

ServerName "Firewalls Security FTP Server" à Nome do servidor
ServerType standalone à Tipo: pode ser standalone ou inetd para rodar sob o inetd
DefaultServer on
Port 21 à Escutará na porta 21
Umask 022
MaxInstances 30

```
User                    nobody à Rodará com permissões do usuário nobody
Group                   nobody
RootLogin               no              à  Não permitirá o root se logar via ftp
DisplayConnect          /etc/welcome.FTP
ServerIdent on"Bem Vindo"
<Directory /*>
        AllowOverwrite  on          à Permite sobrescrever arquivos em qualquer diretório
</Directory>
<Anonymous ~ftp>                    à Usuário anonymous cairá no home do usuário ftp
        User    ftp                 à Logará como usuário ftp
        Group   ftp
        UserAlias   anonymous ftp   à Permite entrar com user=anonymous que virará ftp
        MaxClients      10          à Máximo de conexões anônimas permitidas
  <Limit WRITE>
        Deny All        à  Não permite nenhum usuário anonymous escrever no sistema
  </Limit>
</Anonymous>
```

# Configurando o BIND

## oEntendendo o DNS

Na internet as máquinas se comunicam através de um número único que identifica cada uma (o IP).

Este número possui 32 bits, separados em 4 grupos de 8 bits cada.

Obviamente é humanamente impossível e inviável lembrar-se de tantos números.

Para isto foi criado um sistema que facilita o processo de comunicação entre as máquinas.

Este é o sistema de DNS.

Ele possibilita que demos um nome ao invés de um IP para as máquinas.

Na verdade a comunicação continua sendo via IP, mas nós fornecemos um nome e o servidor de DNS retorna o IP para a comunicação.

Claramente este é um sistema hierárquico (seria impossível um único servidor conhecer os nomes de máquinas da internet toda).

Quero dizer hierárquico porque cada DNS é responsável por uma faixa de Ips e quando recebe uma consulta a um IP que ele não é o responsável ele lança esta consulta um nível acima até chegar no servidor que seja o real responsável.

## oConfigurando uma Zona de Exemplo

Vamos iniciar a configuração de um servidor DNS de exemplo.
Primeiramente devemos conhecer o domínio que queremos configurar.
Como exemplo, configuraremos o domínio firewalls.com.br.

Edite o arquivo /etc/named.conf:

```
options {                                   à Opções globais
        directory "/etc/named".             à Os arquivos ficarão em /etc/named
};
zone "firewalls.com.br" {                   à Domínio firewalls.com.br
        type master;                        à DNS master
        file "firewalls.named";         à Arquivo /etc/named/firewalls.named
};
zone "118.158.in-addr.arpa" {               à Zona Reversa
        type master;                        à DNS master
        file "firewalls.rev";               à Arquivo /etc/named/firewalls.rev
};
zone "." {                                  à Zona .  (Primeira hierárquia=
        type hint;                          à Tipo HINT (root servers)
        file "named.ca";                    à Arquivo que já vem padrão (root servers)
};
zone "0.0.127.in-addr.arpa" {           à Zona reversa do localhost
        type master;                            à Servidor DNS master
        file "local.rev";               à Arquivo /etc/named/local.rev
};
```

## oConfigurando uma Zona Reversa de Exemplo

**Arquivo /etc/named/local.rev:**

```
$TTL 24h        ;  tempo que permite outros DNS fazerem cache
@               IN              SOA             localhost.              root.localhost.  {
                        1       ; para servidor slave
                        28800   ; ciclo de atualização do slave
                        14400   ; ciclo de tentativas do slave antes da próxima em caso de erro
                        3600000  ; tempo limite para que o slave passe a responder
                        86400   }       ; TTL padrão do domínio
                IN      NS      localhost.              ; DNS do domínio será LOCALHOST

1               IN      PTR     localhost.              ; IP 1 host localhost
```

**Arquivo /etc/named/firewalls.rev:**

```
$TTL 24h        ;  tempo que permite outros DNS fazerem cache
@               IN              SOA     ns.firewalls.com.br.            root.firewalls.com.br.  {
                        1       ; para servidor slave
                        28800   ; ciclo de atualização do slave
                        14400   ; ciclo de tentativas do slave antes da próxima em caso de erro
                        3600000  ; tempo limite para que o slave passe a responder
```

```
                86400 }         ; TTL padrão do domínio
        IN      NS      ns.firewalls.com.br.    ; DNS do domínio
        IN      NS      nss.firewalls.com.br.   ; DNS reserva
1       IN      PTR     ns.firewalls.com.br.
2       IN      PTR     nss.firewalls.com.br.
3       IN      PTR     pop.firewalls.com.br.
4       IN      PTR     email.firewalls.com.br.
10      IN      PTR     gateway.firewall.com.br.
```

- **Os arquivos de Zonas**

**Arquivo /etc/named/firewalls.named:**

```
@       IN      SOA     ns.firewalls.com.br.    root.firewalls.com.br.
                1       ; para servidor slave
                28800   ; ciclo de atualização do slave
                14400   ; ciclo de tentativas do slave antes da próxima em caso de erro
                3600000  ; tempo limite para que o slave passe a responder
                86400 }         ; TTL padrão do domínio
        IN      NS      ns.firewalls.com.br.
        IN      NS      nss.firewalls.com.br.
        IN      MX      10 pop.firewalls.com.br.    ; precedência 10 de email
        IN      MX      20 email.firewalls.com.br.  ; precedência 20 de email
                        ; quanto menor a precedência maior a preferência
ns      IN      A       200.158.118.1
nss     IN      A       200.158.118.2
pop     IN      A       200.158.118.3
email   IN      A       200.158.118.4
gateway IN      A       200.158.118.10
localhost IN    A       127.0.0.1
```

# Configurando o Apache

- **Sobre**

O apache pode ser encontrado em www.apache.org <http://www.apache.org/> e é um servidor WEB muito poderoso e gratuito.

Atualmente ele vem sendo utilizado por cerca de 60% dos servidores WEB do mundo e foi inclusive recomendado pelo GartnerGroup.

Com o atual crescimento e onda de vírus e ataques a servidores WEB no mundo, o apache ganhou muita confiabilidade devido a sua imunidade a estes ataques e a comprovação de sua extrema segurança.

- **Porque utilizá-lo**

§Robusto
§Seguro
§Rápido
§Flexível
§Administrável
§Cheio de recursos e opções
§Fácil configuração

## oExemplo de configuração

Aqui forneço um exemplo de configuração do Apache e explico os pontos mais importantes.

```
# Configurações Principais
Port 80                          à  Servidor Roda na Porta 80
User nobody                      à  Roda com permissões do user nobody
Group nobody
ServerType standalone            à  Roda standalone e  não sob inetd
ServerRoot /etc/httpd            à  Arquivos do servidor estão em /etc/httpd
MinSpareServers 2                à  Mínimo de 2 servidores ativos (e sem conexão)
MaxSpareServers 4                à  Máximo de 4 servidores ativos (e sem conexão)
StartServers 2                   à  Inicia 2 servidores no começo
MaxClients 150                   à  Máximo de 150 conexões

# Módulos  (apenas um exemplo, o servidor não funcionará se não
# forem declarados outros módulos necessários)
LoadModule php4_module    modules/libphp4.so          à  Identifica módulo php4
<IfDefine SSL>                   à  Se for definido SSL (httpd -DSSL)
        LoadModule ssl_module     modules/libssl.so     à  Identifica módulo ssl
</IfDefine>
ClearModuleList                  à  Limpa lista de módulos carregados
AddModule mod_php4.c             à  Carrega (insere) módulo no apache
<IfDefine SSL>
        AddModule    mod_ssl.c
</IfDefine SSL>

# Hosts Virtuais
NameVirtualHost  www.firewalls.com.br <http://www.firewalls.com.br/>  à  Caso uma
                                 conexão chegue sem o nome de     destino,
                                 o default será www.firewalls.com.br
<VirtualHost www.firewalls.com.br <http://www.firewalls.com.br/>>          à  Host
Virtual www.firewalls.com.br
        DocumentRoot /home/httpd/firewalls à  Documentos se encontram em...
        ServerName   www.firewalls.com.br <http://www.firewalls.com.br/>     à  Nome
do servidor a ser apresentado
```

ServerAdmin rodrigo@firewalls.com.br <mailto:rodrigo@firewalls.com.br> à Email do administrador, caso algum erro ocorra
</VirtualHost>
<VirtualHost www.br.tcpdump.org <http://www.br.tcpdump.org/>>
DocumentRoot /home/httpd/tcpdump
ServerName www.br.tcpdump.org <http://www.br.tcpdump.org/>
ServerAdmin rodrigo@br.tcdump.org <mailto:rodrigo@br.tcdump.org>
</VirtualHost>

# Diretórios
<Directory "/home/httpd/firewalls" > à Diretório /home/httpd/firewalls
Options Indexes FollowSymLinks Includes ExecCGI à Opções de acesso ao diretório
AllowOverride None à Não permite ser sobreposto pelo arquivo .htaccess
Order allow, deny à Ordem de permissão
Allow from all à Permite todos
</Directory>
<Directory "/home/httpd/tcpdump" >
Options Indexes FollowSymLinks Includes ExecCGI
AllowOverride None
Order allow, deny
Allow from all
</Directory>

# Arquivos
<Files .htaccess> à Arquivos nomeados .htaccess
Order allow, deny
Deny from all
</Files>

# Logs
ErrorLog /log/apache/ApacheError.log à Log de Erros do Apache
LogLevel warn à Logs no nível de Alerta (normal)
AccessLog /log/apache/ApacheAccess.log à Logs de Acesso do Apache

# Opções Diversas
UserDir web à Diretório nos homes dos usuários que armazenam suas páginas, será acessado assim: ww.servidor.com.br/~usuario
DirectoryIndex index.html index.htm index.php à Estes arquivos são os índices das páginas
AccessFileName htaccess à Opções de acesso também são encontradas no arquivo .htaccess

HostnameLookups Off à Não transforma Ips em Nomes
Alias /focker /home/httpd/firewalls/focker à Apelido para /focker

# Locations
<Location /focker> à Caso acessem www.servidor.com.br/focker
Order deny,allow à Ordem de permissão
Deny from all à Proíbe todos
Allow from localhost à Exceto da máquina local
</Location>

# Segurança

## oQuesitos para um sistema seguro

Muitas pessoas nos perguntam constantemente o que é necessário para se montar um sistema seguro.

Possivelmente elas se decepcionam com as respostas:

Basta se manter atualizado e ter uma boa política de segurança

Isto quer dizer que devemos agir com segurança, sempre pensar na segurança.

Não existem regras fixas. Não existe segurança 100%.

O básico que podemos fazer será visto aqui, obviamente técnicas e implementações mais avançadas de segurança existem, mas não caberiam no escopo deste treinamento.

E vale sempre lembrar:

DESABILITE tudo que não for necessário!!!

## oOnde se atualizar

A comunidade de segurança é mais lenta do que os crackers devido ao fato de segurar mais a informação.

Obviamente isto não é uma verdade 100% absoluta, mas os meios de comunicação dos atacantes realmente funcionam melhor e são mais rápidos.

Eles incluem desde programas BATE PAPO famosos como o IRC e o ICQ até listas de email e sistemas USENET exclusivos.

Os profissionais de segurança e todos os interessados podem contar com empresas e órgãos que divulgam as informações nos países.

Um deles é o CERT (Computer Emergency and Response Team).

Maiores informações e treinamentos também podem ser obtidos através do curso ISPA 600 - Segurança em Comunicação de Dados e Firewall.

Uma lista de discussão sobre o assunto, porém fechada a apenas profissionais que trabalham na área pode ser encontrada através do site:

www.firewalls.com.br/mailman/listinfo/seguranca <http://www.firewalls.com.br/mailman/listinfo/seguranca>

## oMontando um firewall linux

§*Entendendo o iptables*

Netfilter e as iptables são a mais nova inovação em Firewall para Linux. Finalmente diversos conceitos avançados de Firewall foram implementados para um sistema Linux.

Máquinas Linux podem agora substituir caríssimas "caixas pretas", ou seja, hardwares comerciais feitos exclusivamente para atuarem como Firewall de uma rede.

§*Regras de entrada, saída e passagem de pacotes*

Existem 3 possibilidades para um pacote transitando em uma com um Firewall e que envolvam este Firewall.

1-) O pacote é para o Firewall
Regra de Entrada: INPUT

2-) O pacote passará pelo Firewall
Regra de Repasse: FORWARD

3-) O pacote sairá do Firewall
Regra de Saída: OUTPUT

§*Tabelas do iptables*

O Iptables como o próprio nome indica trabalha com tabelas. Existem 3 tabelas no Iptables, a saber:

1-) Tabela Filter
Tabela default que trata os pacotes normalmente

2-) Tabela NAT
Utilizada em mascaramentos e NAT

3-) Tabela Mangle
Utilizada para reescrever informações dos pacotes

Obviamente devido ao tempo, iremos tratar apenas da tabela FILTER

§*Proibindo protocolos*

Antes de lidarmos com o iptables em si, é muito importante salientarmos que um bom Firewall consiste de um conjunto de regras bem elaboradas, com uma política de segurança consistente.

Alguns de vocês já escutaram e todos com certeza escutarão falar sobre a CHAIN default de um firewall.

Mas o que é isto?

Chain DEFAULT nada mais é do que o que o Firewall deverá fazer ao se deparar com uma condição que não está prevista, ou seja, não existe uma regra específica para esta condição.

O ideal é SEMPRE utilizarmos CHAIN DEFAULT DROP, mais ou menos aquela política:

" TUDO que não estiver explicitamente liberado está proibido"

Quando fazemos isto, apenas devemos ir colocando nossas regras para liberar o que desejamos fazer, todo o resto já estará proibido.

Vamos para a parte prática:

# Colocando as CHAIN Default
iptables -P INPUT DROP --> Chain default de entrada
iptables -P OUTPUT ACCEPT --> Chain default de saída
iptables -P FORWARD DROP --> Chain default de repasse

# Proibiremos todos os pacotes TCP:
iptables -A INPUT -j DROP -p tcp

-A INPUT --> Significa que estamos colocando esta regra ao final de nossas regras de entrada
-j DROP --> Significa que esta regra é uma regra de PROIBIÇÃO
-p tcp --> Esta regra se refere ao protocolo TCP

§*Proibindo portas e tipos de mensagens*

# Proibindo darem ping em nós
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

Novamente, agora o que mudamos?

-p icmp --> Protocolo ICMP (usado pelo aplicativo PING)
--icmp-type --> Esta regra nos permite especificar que tipo de mensagem icmp estaremos lidando, neste caso a solicitação de ping (echo-request)

# Permitindo para nós darmos ping
iptables -A INPUT -p icmp --icmp-type echo-replay -j ACCEPT

Observando que agora mudamos o tipo de mensagem icmp e aceitamos este tipo.

§*Proibindo flags TCP*

O que são FLAGS TCP?

Obviamente que um profissional de segurança precisa dominar perfeitamente os conceitos envolvidos em rede.

Não é o objetivo deste treinamento desenvolvermos toda esta hablidade nos alunos, mas sim, colocarmos eles à par do que existe no mundo Linux.

Explicando de um modo simples, flags TCP são campos de um pacote TCP que permitem identificar qual o objetivo de uma TRANSMISSÃO.

Ou seja, podemos desejar a abertura de uma conexão (Flag SYN ativada), ou o fechamento de uma conexão (Flag FIN ativada).

Podemos também especificar a aceitação de uma conexão (Flag Syn e ACK) ativadas.

Um Firewall pode ser chamado de Stateful Firewall quando possui armazenada os estados das conexões que por ele se estabelecem.

Ou seja, ele sabe quem abriu ou quem esta fechando uma conexão através dele, criando tabelas de controle (connection track).

Vamos imaginar um Firewall Linux de Antigamente (antes da série 2.4 do kernel do sistema, quando ainda utilizávamos ipchains).

Caso você tenha bloqueado um pedido de conexão na porta telnet de uma máquina de sua rede:

ipchains -A forward -j deny -p tcp -d 192.168.0.1 23 --syn

Observe que a opção final --syn especifica que a flag SYN deste pacote tem de estar ativada.

Os atacantes aproveitavam da "falta de memória" do Firewall para fazer o que chamamos de Stealth Scan, ou seja, pesquisar as portas abertas de uma máquina de um modo que o Firewall não detectava.

Como eles faziam isso?

Forjando pacotes com a FLAG ACK ativa, como se fosse uma aceitação de conexão. Devido a não ter este controle, os scans passavam despercebidos pelo Firewall.

Com o iptables podemos controlar as flags e os estados das conexões.

Controlando Flags:

iptables -A INPUT -p tcp -j DROP --tcp-flags SYN,FIN SYN,FIN

O que isto quer dizer?

Quer dizer que caso as FLAGS SYN e FIN (--tcp-flags SYN,FIN SYN,FIN) estejam setadas nosso sistema irá bloquear a conexão. Isto faz sentido, já que ninguém iria querer ABRIR e FECHAR a conexão em um mesmo pacote.

§*Proibindo estados de pacotes*

Estado de um pacote podem ser 4:

-Novo (NEW)
-Estabelecido (ESTABLISHED)
-Relacionado (RELATED)
-Inválido (INVALID)

Um pacote se diz novo quando está sendo estabelecida uma conexão
Estabelecido quando faz parte de uma conexão já estabelecida
Relacionado quando é um novo pacote, mas faz parte de uma conexão já estabelecida

O FTP por exemplo utiliza 2 portas de comunicação, caindo nesta regra
Inválido

Quando possui algo estranho com o mesmo

Uma regra interessante seria a seguir:

iptables -A INPUT -j DROP -p tcp -m state --state INVALID

Onde estamos proibindo todos os pacotes inválidos de virem para o Firewall.

§*Limitando as conexões*

Muitos ataques realizados (a grande maioria) visam apenas causar algum tipo de negação de serviço, ou seja, impedir que o servidor funcione adequadamente.

Para evitar este ataque, podemos utilizar algumas regras simples, porém práticas de controle:

iptables -A INPUT -j ACCEPT -p icmp -m limit --limit 3/m

Com esta regra limitamos o número de pacotes icmp aceitos em nosso firewall para 3 por minuto.