**FACULDADE DE MINAS – FAMINAS-BH**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**EDUARDO FERREIRA DE BRITO**

# Sistema operacional GNU/Linux: um estudo sobre economia, estabilidade e segurança para tratamento das informações de microempresas

**BELO HORIZONTE**
**2008**

**EDUARDO FERREIRA DE BRITO**

# Sistema operacional GNU/Linux: um estudo sobre economia, estabilidade e segurança para tratamento das informações de microempresas

Trabalho de conclusão de curso apresentado ao Curso de Sistemas de Informação, da Faminas-BH - Faculdade de Minas, como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Ricardo Terra Nunes Bueno Villela

**BELO HORIZONTE**
**2008**

Eduardo Ferreira de Brito

# Sistema operacional GNU/Linux: um estudo sobre economia, estabilidade e segurança para tratamento das informações de microempresas

Objetivo: Evidenciar os pontos marcantes da utilização do Linux em microempresas, demonstrando aspectos econômicos, de segurança, importância dos backups e a estabilidade proporcionada pela utilização do GNU/Linux.

FAMINAS-BH – Faculdade de Minas
Curso de Sistemas de Informação

Área de concentração: Linux, GPL, segurança e *backup*

Data de aprovação: _____ / _____/ _____

_____

Prof. Ricardo Terra Nunes Bueno Villela
Orientador

_____

Prof. Paulo Henrique Fernandes de Matos
Coordenador do Curso de Sistemas de Informação

Dedico esse trabalho a meus pais, minha irmã e minha noiva Daniela por sempre estarem comigo, dando apoio e força para continuar em frente.

## AGRADECIMENTOS

Primeiramente, agradeço a Deus por me dar forças nos momentos mais difíceis e sabedoria para solucionar os problemas.

Agradeço à minha noiva, Daniela, por me ajudar nos finais de semanas, sempre me apoiando no que fosse possível.

Agradeço a meus pais e minha irmã pela condição favorável e apoio em mais uma conquista

Enfim, agradeço a meu orientador, Ricardo Terra, e a todos que me ajudaram a dar mais um passo rumo à conquista dos meus sonhos.

“*O Linux está crescendo muito, é como uma bola de neve que você não consegue mais parar*”.

Carlos Eduardo Morimoto

## RESUMO

O GNU/Linux é um sistema operacional desenvolvido a partir da fusão do núcleo do Linux com os aplicativos GNU como alternativa ao Unix. O objetivo deste estudo é demonstrar o GNU/Linux como um sistema operacional de qualidade para utilização em servidores de microempresas por ser economicamente viável, estável e seguro. Em relação à economia, foram abordados detalhes sobre sua licença GPL que garante liberdade de modificação sem imposição de restrições. Em relação à estabilidade, foram apresentadas as diversas ferramentas voltadas a manter a disponibilidade do sistema. Por fim, em relação à segurança, foram abordadas técnicas como filtragem de conteúdo, monitoramento e controle de acesso e execução. Enfim, pretendeu-se demonstrar o GNU/Linux como um sistema economicamente viável de alta confiabilidade.



**Palavras-chave:** *linux; GNU; GPL; segurança; economia; backup; firewall; senhas; falsa segurança; permissão de acesso; journaling.*

## ABSTRACT

The GNU/Linux is an operating system developed from the fusion between the Linux kernel and the GNU applications as an alternative to the Unix. The main goal of this study is to demonstrate the GNU/Linux as a quality operating system to use in main servers of small businesses in order to be economically viable, stable, and safe.Regarding the economy, details were presented about its GPL license which allows any modification without any imposed restrictions.In respect to the stability, several tools were presented aiming at ensuring the avaiability of the system. And respecting to the security, some techniques were presented such as, for example, content filtering, monitoring and control access and execution. Thus, this work aims at demonstrating the GNU/Linux as an economically viable system with high reliability.



**Keywords:** *linux; GNU; GPL; security; economy; backup; firewall; passwords; fare security; access permission; journaling.*

## LISTA DE FIGURAS

## LISTA DE ABREVIATURAS E SIGLAS

CDs - *compact discs,* em português, discos compactos.

CPD - Central de processamento de dados.

CPU - *Central Processing Unit,* em português, unidade central de processamento.

DOS - *Denial of service,* em português, negação de serviço.

DVDs - *Digital Versatile Disks,* em português, discos versáteis digitais.

EXT3 - *third extended file system,* em português, terceiro sistema de arquivos estendido.

GID - *Group Identification,* em português, identificador de grupo.

GNU - *GNU's Not Unix*

GPL - *General Public License,* em português, licença de uso geral.

HD - *Hard disk,* em português, disco rígido.

ID's – *Identifications,* em português, identificações.

IIS - *Internet Information Services*

JFS - *Journaling File System,* em português, sistema de arquivos Journaling.

MB - *Megabyte*

NIST - *National Institute of Standards and Technology,* em português, Instituto Nacional de Padrões e Tecnologia.

PC - *Personal computer,* em português, computador pessoal.

PCs - *Personal computers,* em português, computadores pessoais.

PID - *Process identifier ,* em português, identificador de processos.

REISERFS - *Reiser file system,* em português, sistema de arquivos Reiser.

SO - Sistema operacional.

TKIP - *Temporary Key Integration Protocol,* em português, Protocolo de Integridade de Senha Temporal.

UID - *User identification,* em português, identificação de usuário.

WEP - *Wired Equivalent Privacy.*

WPA - *Wi-Fi Protected Access.*

XFS - *X file system,* em português, sistema de arquivos X.

**SUMÁRIO**

# 1 INTRODUÇÃO

O Linux é um sistema operacional desenvolvido por Linus Torvalds em 1991, com propósito de ser um sistema operacional poderoso, porém livre para que todos pudessem utilizá-lo e modificá-lo como quisessem. A base para o desenvolvimento era um outro sistema operacional chamado Minix que trata-se de "uma espécie de UNIX[1] reduzido para estudos acadêmicos" (JAMIL; GOUVÊA, 2006, p.3). O Linux também foi desenvolvido como alternativa ao Unix que tinha um custo muito elevado e precisava de hardware específico, restringindo assim o seu uso. Desde sua criação, Torvalds não estava interessado em obter lucro, mas sim, em fazer um sistema operacional que atendesse às suas necessidades a fim de utilizá-lo em seu dia-a-dia.

O sistema operacional (SO) é o principal software do computador e é atribuído a ele o controle de todo o hardware e programas instalados. Para Tanenbaum (2003), o sistema operacional expõe ao usuário uma interface com o hardware mais simples do que de fato ela é. Marques e Guedes (1998) explicam que transformar o computador com todos os seus circuitos eletrônicos e periféricos em uma máquina com utilização simplificada também é de responsábilidade do sistema operacional.

Após a divulgação do Linux, muitas pessoas aderiram à idéia e começaram a propor melhorias. Contudo, antes do Linux ser concebido, Richard Stallman e sua equipe participavam do desenvolvimento do GNU[2] que almejava desenvolver um sistema operacional totalmente livre com código fonte aberto, permitindo à comunidade contribuir para o seu aperfeiçoamento. Entretanto, Matos (2007) explica que o projeto não atingiu seu objetivo, pois o GNU chegou ao final da década de 1980 sem possuir um núcleo. Na década de 1990, o Linux já estava passando por rápidas evoluções e não demorou muito para ocorrer a união dos sistemas de software do GNU com o núcleo do sistema operacional do Torvalds, evitando assim o fracasso do projeto GNU de Stallman. Segundo McCarty, em 05 de outubro de

[1] Unix: Sistema operacional de grande porte, multiusuário e multitarefa, com possibilidade de execução em diversos tipos de computadores.

[2] De acordo com GNU (2007), GNU significa "*GNU's Not Unix*", ou seja, uma sigla recursiva sem significado bem definido. GNU também é utilizada para referenciar *General Public License*.

1991, foi lançada oficialmente a primeira versão do GNU/Linux distribuída sobre licença GPL.

A GPL, de acordo com GNU (2007), garante a liberdade de compartilhar, modificar ou utilizar trechos de todos os programas e demais trabalhos que a ela se aplica. Essa licença garante a liberdade, mas não o preço, ou seja, o desenvolvedor pode cobrar pelo software ou pelo trabalho, mas não pode restringir o acesso de outras pessoas ao programa ou código-fonte.

O Linux tem estabilidade e segurança como diferencial em relação a outros sistemas operacionais. Ele provê inúmeros recursos que evitam os travamentos e a necessidade de reinicialização do sistema, além de várias opções de configurações que elevam substancialmente a segurança do sistema. É importante ressaltar que muitas vezes a segurança de um sistema está na atitude dos administradores de rede, pois o investimento em servidores, *firewall*s e *no-breaks* podem ser inúteis caso a CPD (Central de processamento de dados) não possua restrição de acesso físico, permitindo o ingresso de pessoas não autorizadas.

Ao realizar a migração para o Linux, muitos usuários passam a adotar um comportamento de risco utilizando o sistema com o usuário de administração a todo o momento, pois imaginam que após a instalação do novo sistema operacional todos os problemas de segurança estarão resolvidos. Segundo Morimoto (2006b), essa postura pode resultar em brechas de segurança. Outro agravante é quando o administrador do sistema se preocupa tanto em evitar invasões que acaba esquecendo de fazer o *backup* do sistema. Convém ressaltar que defeito em um dos discos do servidor pode acarretar mais prejuízos do que um invasor no sistema.

Atualmente, as empresas necessitam gerenciar uma maior quantidade de informações e, para isso, é necessário a adoção de sistemas informatizados. Essa necessidade crescente de informações resulta em novas preocupações. Tecnologias que serão utilizadas, recursos de software e hardware passam a ocupar um papel trivial no que diz respeito a custos. Como o Linux, segundo Hunt (2004), disponibiliza um alto poder computacional e um custo acessível, ele se torna economicamente viável na adoção em servidores.

Este estudo demonstra o Linux como uma alternativa eficiente para utilização em microempresas, contudo grande parte do conteúdo abordado pode ser utilizado em empresas com diferentes estruturas.

A constituição federal, no seu decreto 5.028, define microempresa como “a pessoa jurídica e a firma mercantil individual que tiver receita bruta anual igual ou inferior a R$ 433.755,14 (quatrocentos e trinta e três mil, setecentos e cinqüenta e cinco reais e quatorze centavos)”. (BRASIL, 2004); fica assim, delimitado o ambiente deste estudo.

O objetivo geral deste trabalho é evidenciar os pontos mais relevantes da utilização do Linux em uma microempresa, exemplificando seus maiores benefícios. Para alcançar essa meta, os seguintes objetivos específicos foram traçados:

- Abordar fatores econômicos;
- Demonstrar a segurança envolvida na utilização do Linux;
- Salientar a importância dos *backups* e a estabilidade proporcionada pelo sistema.

A necessidade de abordagem do tema surgiu da ausência de informações sobre a escolha do sistema operacional de servidores, observando que a maioria dos autores aborda somente como fazer e configurar determinado serviço, deixando de lado informações sobre as vantagens que um sistema operacional tem em relação ao outro. Para tratar justamente dessa carência, este trabalho visa demonstrar ao leitor porque o Linux é um sistema operacional com maturidade suficiente para atender às microempresas.

## 2 GNU/LINUX

O Linux é um sistema operacional desenvolvido em 1991 por Linus Benedict Torvalds, até então estudante de Ciência da Computação da Universidade de Helsinki, na Finlândia. Torvalds tinha em mente construir um sistema operacional poderoso, porém livre para que todos pudessem utilizá-lo e modificá-lo como quisessem. O Linux foi baseado em outro sistema operacional chamado Minix, que foi concebido por Andrew Tanenbaum. De acordo com Matos (2007), ele foi desenvolvido para fins acadêmicos e era baseado na arquitetura 8086 devido ao preço mais acessível.

O Minix teve como inspiração um outro sistema operacional chamado Unix, que segundo o próprio autor, era muito caro e com necessidade de hardware específico ficando inacessível para muitos. Jamil e Gouvêa (2006) concluem que o Minix provia de código fonte aberto. Ainda, Matos (2007) salienta que o Minix só poderia ser utilizado em ambiente acadêmico graças as limitações técnicas, na qual o mesmo só poderia endereçar 1MB de memória e não contava com memória virtual.

Frente às limitações do Minix, Torvalds proveu um *kernel* (núcleo do sistema operacional) multitarefa, multiusuário que provia de memória virtual batizando-o de Linux. Matos (2007) explica que o nome Linux surgiu da união dos nomes Linus e Unix.

O Linux é gratuito, porque Torvalds ao desenvolvê-lo não estava interessado em obter lucro, mas sim, em fazer um sistema operacional que atendesse suas necessidades, ou seja, ele iria utilizá-lo em seu dia-a-dia. Em agosto de 1991, uma mensagem circulou para um grupo utilizando a Usenet (grupo de notícias) na qual o próprio Linus dizia.

> Como eu mencionei há um mês, estou trabalhando em uma versão livre de um sistema operacional similar ao Minix para computadores AT-386. Ele finalmente alcançou o estágio onde pode ser utilizado (ou não, depende do que você deseja), e eu estou disposto a colocar os fontes disponíveis para ampla distribuição. Ele está apenas na versão 0.02, mas eu tenho executando nele, sem problemas, programas como *bash*(Shell), *gcc*(compilador), *gnu-maker, gnu-sed, compress*,etc. (TORVALDS, 1991 apud JAMIL; GOUVÊA, 2006, p.6).

Logo após sua divulgação, muitas pessoas começaram a participar e propor novas funcionalidades, incluindo assim rápidas e constantes melhorias, o que resultou, segundo Jamil e Gouvêa (2006), em uma fusão do *kernel* do Linux com os programas do GNU, resultando na primeira versão do Linux em 5 de outubro de 1991.

GNU que por sua vez trata-se de um projeto iniciado por Richard Stallman em 1984, tendo como principal objetivo, ser um software totalmente livre, com código fonte aberto, podendo assim, a comunidade dar sua contribuição para melhorar o sistema. Matos (2007) explica que o projeto não deu certo, pois chegaram ao final da década de 1980 sem cumprir seu objetivo que era desenvolver um clone do Unix. O fracasso da GNU não foi total porque ocorreu a fusão de seus programas com o núcleo do Linux. Com essa união, o problema foi resolvido e o sistema operacional resultante passou a ser conhecido com GNU/Linux, que, a partir deste ponto, será referenciado apenas como Linux.

Atualmente, o Linux é distribuído por uma licença GPL (*General Public License*), que se trata de "um mecanismo de fornecimento de software livre" (JAMIL, 1999, p. 30), ou seja, ninguém precisa pagar para utilizá-lo.

> O Linux é aderente, por decisão de Linux Torvalds, à GPL. É um produto livre, não de domínio público. Você poderá obtê-lo de graça, alterá-lo, cobrar pelas alterações, implementos, melhorias e modificações que fizer, mas não pode restringir os direitos aos seus clientes e detentores. (JAMIL, 1999, p. 32).

Matos (2007) conclui que nem o próprio criador do Linux tem a permissão para alterar a licença ou exigir o software para ele. Em outras palavras, o Linux, de acordo com o GPL, sempre será disponível a todos.

Após o Linux se tornar um SO[3] completo, ele começou a ser utilizado em várias situações. Hunt (2004) explica que o alto poder computacional e o baixo custo funcionam como impulso para utilização cada vez maior em servidores. Provando ser eficaz em alternativa aos altos custos de licenças para servidores, o autor

[3] O SO ou Sistema operacional é um software que controla todo o hardware e aplicativos instalados em um sistema informatizado, em síntese, ele "é o responsável por dar vida ao PC" (MORIMOTO, 2006a, p. 9). Marques e Guedes (1998) explicam que ele tem a capacidade de transformar circuitos eletrônicos, discos e periféricos em uma máquina simples de utilizar. Tanenbaum (2003) conclui que o sistema operacional deve fornecer aos programas do usuário uma interface com hardware mais simples do que realmente é.

completa que o Linux é uma escolha perfeita para construção de servidores de rede, pois contém uma vasta gama de serviços, em sua maioria também livres, para todas as necessidades que a rede possa precisar. Ele comenta ainda que muito da popularidade que o Linux conquistou, ocorre devido ao uso disseminado na construção de servidores *web*.

## 2.1 Licenças, Custos e Suporte

Como já mencionado o Linux é distribuído pela licença GPL que segundo GNU (2007) garante a liberdade de compartilhar, modificar ou usar trechos de todos os programas e demais trabalhos que a ela se aplica. A GPL garante a liberdade, mas não o preço, ou seja, o desenvolvedor pode cobrar pelo software ou pelo trabalho. A GNU explica também que todas as versões modificadas, devem ser assim marcadas evitando atribuições errôneas ao autor.

A licença GPL[4] impõe algumas condições para um software ser classificado como software livre, GNU (2007) descreve que o software deve contar com alguns princípios de liberdade:

- O direito de executar o programa para qualquer finalidade;
- O direito de estudar como o programa funciona e modificá-lo conforme as necessidades;
- O direito de fazer novas cópias e distribuí-las;
- O direito de aperfeiçoar o programa e disponibilizar ao público;

Para garantir esses direitos, Matos (2007) informa, deve-se cumprir os seguintes requisitos principais expressos na licença:

- Devem estar claramente expostos os avisos sobre os direitos autorais e a desobrigação de garantia. Ainda deve-se manter sem alteração qualquer outra informação sobre licença ou garantia. Uma cópia da licença deve sempre ser entregue junto à cópia do programa e possíveis traduções da GNU (ou GPL) deve ser acompanhada da cópia original em inglês;

[4] Verifique sempre a versão da licença GPL utilizada.

- Alterações ou transformações na obra podem ser distribuídas, somente se mantiver uma licença idêntica;
- Se copiar ou distribuir, deve-se incluir uma forma de acesso ao código fonte completo válido por pelo menos três anos.

Contudo essas condições podem ser alteradas, desde que em negociação com o autor do programa.

Como o Linux é um sistema operacional gratuito as distribuições passaram a adotar políticas comerciais alternativas. Jamil e Gouvêa (2007) explicam que o indivíduo ou a empresa que fizer a adesão do produto via Internet, não terá ônus algum para efetuar o *download* (copiar um arquivo de uma página da web), porém o mesmo usuário pode optar em comprar o pacote de uma das distribuições. Nesse caso, a empresa estará vendendo o serviço e não o Linux. Os autores completam que a distribuição Linux também pode cobrar por suporte ou treinamento quando solicitado.

A Red Hat disponibiliza vários planos para o usuário, dentre eles, segundo a Red Hat (2008), Red Hat Enterprise Linux 32/64-bit x86 Itanium2, em versões Basic, Standard e Premium. Valores praticados em dezembro de 2008, €337,59, €773,19 e €1257,19 (valores em Euro) respectivamente.

Embora o suporte Linux seja cobrado na maioria das vezes, a facilidade em obter informações gratuitas sobre configurações, novas funcionalidades e segurança é muito grande, pois existem *sites* e fóruns especializados em ajudar os usuários. No Brasil, o *site* Viva o Linux[5] é uma referencia muito utilizada para sanar várias das dúvidas dos usuários.

## 2.2 Atualizações

Sempre quando nos deparamos com sistemas informatizados, surge necessidade de atualizações, seja por *bugs* (erros em software que impedem seu funcionamento adequado), por falhas de segurança ou, até mesmo, por incluir uma

[5] Endereço: http:// www.vivaolinux.com.br

nova funcionalidade. Com o Linux essa situação não é diferente, principalmente quando falamos em servidores Linux pois, uma simples atualização em um determinado pacote de algum programa pode ter ganhos significativos em relação à segurança.

Da mesma forma que as distribuições Linux disponibilizam suporte para sanar dúvidas, elas também mantêm disponíveis atualizações de programas do sistema. Hunt (2004) explica que verificar sempre a página do fabricante pode manter o usuário informado sobre *bugs* e falhas de segurança até então desconhecidos, geralmente as atualizações disponíveis são importantes, mesmo que não seja correção de segurança, pois também existem outros quesitos importantes como estabilidade e funcionalidade. Convém salientar que se você não utilizar um determinado serviço em seu servidor e for descobertas falhas de segurança ou *bugs* nesses serviços, a atualização não é necessária, pois essas falhas não afetarão os outros serviços de seu sistema.

Ao atualizar o software deve-se ter conhecimento do que será atualizado. Hunt (2004) explica que consultores de segurança como os da NIST (*National Institute of Standards and Technology)* dos EUA (Estados Unidos da América) normalmente descrevem o problema e informam a solução e em muitos casos avisam qual é a atualização necessária.

Atualizar um software do sistema pode corrigir muitos problemas, da mesma forma que pode causar outros. Falhas durante o processo de atualização podem resultar em um problema crítico em seu servidor, evitando que o mesmo inicie a interface gráfica, impeça de fazer *logon* no sistema, entre outros erros. A fim de minimizar os danos causados por esse tipo de falha, sempre faça *backup* antes de qualquer atualização, principalmente as que envolvam arquivos de sistema.

Uma das maneiras mais fáceis de atualizar o Linux é utilizando o comando *apt-get* que, para Faria (2004), é um aplicativo que instala o programa desejado juntamente com todas as suas dependências, passando como parâmetro a opção *update* (*apt-get update*), ele efetua o *download* de uma lista de aplicativos atualizada e com o comando *upgrade* (*apt-get upgrade*), ele compara os aplicativos instalados e verifica se existe versão mais recente e em caso positivo, ele faz o *download* e instala automaticamente a atualização. Vale alertar que não são todas as distribuições Linux que utilizam *apt-get,* por exemplo, a distribuição Suse utiliza o *Yast* como gestor de atualizações.

## 2.3 Estabilidade

Um dos principais pontos que diferem o Linux de outros sistemas operacionais é a sua estabilidade. Ele conta com inúmeros recursos para evitar travamentos e reinicialização do sistema, claro que alguns cuidados devem ser tomados para manter todo o sistema funcionando, desse modo Hunt (2004) defende a seguinte posição: é melhor se prevenir a ter que consertar os erros resultantes de falta de atenção ou negligência.

Alguns passos para manter o sistema estável são bem semelhantes à mantê-lo seguro. Segundo Hunt (2004), a correção dos *bugs* detectados no servidor além de garantir em muitos casos maior segurança, podem garantir também estabilidade. Ainda para esse autor, a confiança é mais importante que novos recursos, sendo assim, ele aconselha o usuário a instalar novas funcionalidades somente após essas terem sido testadas, seja em um *desktop* (computador com função estação de trabalho) ou em um servidor de testes, certifique-se do funcionamento e somente então as melhorias devem ser instaladas no servidor.

Morimoto (2006a) explica que o núcleo do Linux é extremamente estável, porém aplicações muitas vezes não. O autor conclui seu pensamento informando que algumas distribuições apressadas em disponibilizar novas funcionalidades em seu sistema incluem programas em estágio *beta* (em estágio de teste) ou, em casos mais críticos, programas *alpha* (estágio de teste inicial), que ainda têm sua estabilidade questionável, podendo assim comprometer a estabilidade do sistema. Vale ressaltar ainda sobre a possibilidade de ocorrer conflitos entre versões de bibliotecas e sistemas compartilhados utilizados pelos aplicativos, o que podem resultar também em problemas de estabilidade.

### 2.3.1 Garantindo a estabilidade

Para garantir que o sistema continue funcionando, o Linux contém alguns recursos úteis. Morimoto (2006a) comenta que o usuário praticamente nunca precisará reiniciar o sistema completamente, basta apenas fechar o aplicativo com problema e, em casos mais graves, reiniciar a interface gráfica.

O Linux conta com ferramentas específicas para finalizar programas “travados”, a maneira mais fácil é utilizando o *xkill*, que ao “clicar sobre o ícone do programa, ou chamá-lo pelo terminal (digitando xkill), o cursor do mouse vira um ícone de caveira e basta clicar sobre o programa travado para matá-lo sem dó”. (MORIMOTO, 2006a, p. 88).

Apesar do *xkill* ser a maneira mais fácil de finalizar processos, nem sempre ele é usual, pois muitos servidores Linux não utilizam interface gráfica (X), impossibilitando assim a sua execução. Logo, nesses casos utiliza-se o *killall* ou *kill*.

O comando *killall* é utilizado sempre quando o usuário souber o comando responsável por executar o programa que deseja finalizar, bastando digitar no terminal[6] *killall* <aplicação>, ou seja, *killall* concatenado com a aplicação desejada, Morimoto (2006a) informa que o problema em utilizar o *killall*, é quando o nome do programa não é o mesmo que o necessário para finalizá-lo, por exemplo ao finalizar o Firefox o comando a ser digitado é “killall Firefox-bin”.

Por fim, o comando *kill* que, assim como o *killall,* também é utilizado em terminal, contudo utiliza o PID (***P****rocess* ***Id****entifier* ou identificador do processo, em português). Morimoto (2006a) explica que basta unir o comando kill seguido do PID do processo que deseja encerrar, ficando da seguinte forma “kill 4060”. Jamil e Gouvêa (2006) salientam que o *kill* é muito utilizado em *Shell script*[7] para controlar a execução de *daemons* que ficam rodando em segundo plano.

Os *dameons,* segundo os autores, são os programas que ficam em execução em segundo plano para servir outros programas, em outras palavras é um software que não estabelece relação direta com o usuário, ele cria uma interface com as aplicações que interagem. Para Dazs (2007), a *daemon* inicia, reinicia ou pára a execução de um serviço provido pelo sistema e também é responsável, em muitos casos, de recarregar as configurações do programa. A *daemon* do Linux é muito semelhante aos serviços[8] do Microsoft Windows XP e Microsoft Windows 2003.

---

[6] O terminal é semelhante ao prompt de comando do Microsoft Windows e também conhecido por shell

[7] Para Jargas (2008), *shell script* é uma lista de comandos que devem ser executados em seqüência. É um itinerário de comandos e parâmetros, muito utilizado para automatizar determinadas funções.

[8] O serviços estão localizados no painel de controle/ferramentas administrativas/serviços.

Para descobrir o PID do processo basta utilizar o comando *ps*, que combinado com algumas opções consegue-se obter várias informações além do PID, a opção mais utilizada é o *ps -aux,* onde a opção ***a*** exibirá todos os processos abertos, a opção ***u*** é para exibir o usuário que iniciou o processo e a opção ***x*** exibe os processos que não estão associados a nenhum terminal.

Além de contar com a opção de finalizar os processos “travados”, o Linux conta também com um sistema de prioridades, sendo uma de suas principais funções,

> evitar as famosas *Dead Locks*, impasses entre processos que em um intervalo de tempo tentam acessar recursos que já estão sendo utilizados por outro processo. Logicamente que, com o sistema de prioridades, o sistema disponibilizará o recurso para o processo que possuir maior prioridade. No caso de possuírem a mesma prioridade o sistema escolherá o processo que foi iniciado antes. (JAMIL; GOUVÊA, 2006, p.119).

Os autores concluem que as prioridades definem também várias características dos processos, como por quanto tempo cada processo poderá utilizar a CPU (*Central Processing Unit* ou unidade central de processamento, em português), quais arquivos podem ser acessados, manipulados ou modificados, entre outros.

# 3 SEGURANÇA

As empresas atualmente necessitam de um número crescente de informações e, garantir o acesso rápido a elas, exige o uso constante de tecnologias diferentes, resultando em maior flexibilidade. Em contrapartida, esses recursos causam preocupações referentes à segurança nas organizações. Morimoto (2006b) explica que essa situação torna o trabalho de manter a rede segura mais complexa, pois não adianta ter um *firewall* filtrando possíveis invasões provenientes da Internet se a empresa possuir um acesso *wireless* desprovido de senha.

Para manter um servidor seguro, em muitos casos, bastam simples procedimentos que passam muitas vezes despercebidos. Vale ressaltar que todo investimento feito em servidores *firewall*, *no-breaks*, senhas e controle de acesso podem ser inúteis se o administrador do sistema cometer um erro simples de deixar a porta da CPD (Central de processamento de dados) destrancada. Hunt (2004) explica que o importante para manter um sistema protegido é conhecer suas vulnerabilidades. O administrador deve ficar tão bem informado sobre a situação de seu sistema quanto aos possíveis invasores, evitando dessa forma brechas que comprometam a segurança.

O uso do Linux, devido às inúmeras possibilidades de configurações, promove um servidor substancialmente mais seguro. Morimoto (2006a) salienta que o Linux provê de usuários ocultos (contas de usuário sem privilégios), que tem por finalidade isolar os programas, permitindo que cada um tenha acesso apenas a os seus arquivos. Esse procedimento reduz os danos que um *bug* ou falha de segurança em determinado programa podem causar ao sistema. Esse é apenas um exemplo de regras que fazem do Linux um sistema operacional seguro. Ainda convém salientar que esse procedimento é válido somente para alguns programas, pois muitos necessitam de permissões de usuários específicos.

## 3.1 Vírus, *Worms* e *trojans*

Existem várias “pragas” virtuais que podem comprometer seriamente a estabilidade dos sistemas, o vírus é uma delas. Morimoto (2006b) explica que os

vírus de computador são muito parecidos com os da vida real, pois variam muito o potencial destrutivo. Muitos vírus são tão simples que sua única função é copiar-se o maior número de vezes possível, enquanto outros podem danificar todos os arquivos do HD[9] (*Hard disk*), sobrescrevendo-os de forma que não seja possível recuperá-los. Geralmente, os vírus são disseminados por falhas em navegadores, e-mails, arquivos infectados, ou seja, é sempre necessária da intervenção do usuário.

Existem também os *trojans* que parecem muito com os vírus, “mas o objetivo principal é abrir portas e oferecer alguma forma de acesso remoto à máquina infectada” (MORIMOTO, 2006b, p. 143). Assunção (2008) conclui que o *trojan* é considerado uma evolução do vírus devido ao fato de ser controlado a distância, a pessoa que instala o *trojan* passa, em muitos casos, a controlar o computador da vítima, podendo utilizá-lo como ponte para atacar servidores de alguma empresa sem ser descoberto.

Dentre as pragas virtuais, a que mais oferece poder de infecção é o *worm*, ele não depende da ação do usuário para ser executado. De acordo com Assunção (2008), a maior diferença entre o *worm* e o vírus é a possibilidade de infecção automática pela rede, além de contar com um maior poder “destrutivo”.

> Um worm poderia começar invadindo um servidor web com uma versão vulnerável do IIS, infectar outras maquinas da rede local a partir dele, acessando compartilhamentos de rede com permissão de escrita e, a partir delas, se replicar via e-mail, enviando mensagens infectadas para e-mails encontrados no catalogo de endereços; tudo isso sem intervenção humana. (MORIMOTO, 2006b, p. 143).

Para amenizar os estragos causados por esses programas, Morimoto (2006b) explica que, um *firewall* bem configurado pode barrar as portas de trabalho do *worm*. Por outro lado, para vírus e *trojans* devem-se criar restrições quanto às extensões dos arquivos. Mesmo com um *firewall* bem configurado a melhor forma de proteção continua sendo o antivírus instalado individualmente nas estações de trabalho.

[9] Hd é o local principal para armazenamento dos arquivos e programas.

## 3.2 Diferenças Linux em relação ao Microsoft Windows

Muitos usuários Linux, principalmente os iniciantes, imaginam que basta instalar qualquer distribuição que os problemas estarão resolvidos, contudo essa postura pode comprometer toda a segurança de um sistema. O maior problema é que durante o processo de aprendizado das soluções do Linux, os usuários, segundo Morimoto (2006b), ficam impressionados com a quantidade de recursos disponíveis e acabam abrindo brechas no servidor e, para piorar esse quadro, muitas distribuições habilitam diversos servidores durante o processo de instalação do sistema. Essa situação dificilmente acontecerá no Microsoft Windows, pois a maior parte dos usuários nem sabe da possibilidade de manter um servidor funcionando no computador de casa.

Apesar do Linux em relação ao sistema operacional da Microsoft contar com inúmeras diferenças, em alguns pontos eles têm o funcionamento parecido. Morimoto (2006a) explica que, em ambos, o sistema de permissões para os arquivos contam com três opções para acesso (leitura, gravação e execução) e três opções de grupo (proprietário, grupo e outros). A grande diferença é que no Linux você utiliza usuários comuns para praticamente todos os programas e deixa a cargo do *root* (administrador do Linux) somente as etapas de configuração. Enquanto no Windows você utiliza os usuários com privilégios de administrador, sendo que ao executá-lo como usuário comum, muitos programas podem deixar de funcionar. Esse procedimento reduz substancialmente a segurança do Microsoft Windows, pois se o usuário tem grande liberdade ao utilizar o sistema, os programas executados por ele também poderão obter os mesmos privilégios.

Outro ponto forte do Linux em relação ao Microsoft Windows é a definição dos programas. Morimoto (2006a) explica que o Linux não decide qual arquivo é executável a partir da extensão, mas sim, a partir da permissão que o arquivo tem. Enquanto no Windows para um arquivo ser executável basta possuir a extensão .exe (existem outras como .bat e .com). Esse processo aumenta muito a segurança, pois sempre que um novo arquivo é copiado para o Linux é necessário atribuir a permissão de execução à ele.

### 3.3 Falsa sensação de segurança

Os usuários do Linux ficam mais tranqüilos em relação a possíveis infecções por vírus e derivados, pois, devido sua forma de trabalho baseado em permissões de arquivos, os vírus desenvolvidos são praticamente inexistentes e falhas e vulnerabilidades em servidores que utilizam a plataforma Linux são pouco comuns se compararmos ao Windows. Devido às características de segurança que o Linux oferece, muitos usuários passam a assumir um comportamento inadequado, serviços desnecessários são instalados, utilizam-se senhas fracas e o uso de usuários com permissões de administrador no dia-a-dia, todas essas características Morimoto (2006b) define como inseguras.

Como explicado anteriormente, manter aplicativos desnecessários funcionando em seu servidor pode resultar em sérios problemas de segurança, portanto “reduza o fardo de manter software atualizado removendo todos os que você realmente não precisa” (HUNT, 2004. p. 392). Com a remoção dos aplicativos não utilizados, além de liberar recursos para o restante do sistema, diminui as possibilidades de *bugs,* falhas do sistema e de segurança.

Convém salientar que a segurança está ligada diretamente ao comportamento do usuário, Morimoto (2006b), conclui, explicando que caso um usuário que utilize o Microsoft Windows XP sempre atualizado e não abra arquivos ou programas com procedência duvidosa, mesmo sem utilizar *firewall* ou antivírus provavelmente estará mais seguro que usuários Linux que utilize vários servidores desatualizados e use o sistema como *root*.

Muitos usuários fazem a migração de outros sistemas operacionais para o Linux devido ao mito sobre imunidade a vírus. O que ocorre com o Linux é contar com sistema de permissões de arquivos, evitando que programas sejam auto-executáveis. A utilização de usuários sem permissão *root (*administrador) garantem, em caso de infecções por vírus, um pequeno poder destrutivo reduzindo a atração dos desenvolvedores de vírus.

### 3.4 Permissões de arquivos e usuários

No Linux, sempre que um usuário é criado são atribuídos a ele dois ID's (*Identifications*). O primeiro é chamado de UID (*User identification*) e refere-se à identificação do usuário e o outro é chamado GID (*Group IDentification*) e é utilizado para identificar o grupo. Hunt (2004) explica que, ao acessar um arquivo, o sistema faz a checagem dos valores UID e GID, comparando-os com os do arquivo e, caso os valores forem condizentes, o arquivo pode ser acessado. O autor ainda ressalta que todo arquivo também recebe um UID e um GID e que esses valores são atribuídos à ele no momento de sua criação recebendo os mesmos valores do usuário que o criou. Contudo esses valores podem ser alterados a qualquer momento.

As permissões dos arquivos são compostas dos seguintes valores segundo Hunt (2004):

- **Permissões de proprietário:** São as permissões para o usuário que tem o mesmo UID do arquivo;
- **Permissões de grupo:** São as permissões para os usuários que tem o mesmo GID do arquivo;
- **Permissões Globais:** São as permissões dadas para todos os outros usuários, ou seja, quando o UID e o GID do usuário são diferentes das contidas no arquivo;
- **Permissões de leitura:** Ocorre quando o conteúdo do arquivo pode ser analisado (o arquivo pode ser aberto);
- **Permissões de escrita:** Ocorre quando o conteúdo do arquivo pode ser modificado;
- **Permissões de execução:** o programa (ou *script*) pode ser executado.

Todas essas permissões podem ser facilmente alteradas utilizando os comandos *chmod*, *chown* e *chgrp*. Para fazer essas alterações é necessário ser o proprietário do arquivo (valor do UID do usuário igual ao do arquivo) que deseja alterar ou o usuário contar com privilégios de *root*.

O *chmod*, de acordo com Jamil e Gouvêa (2006), é utilizado para manipular as permissões de arquivos e diretórios no sistema. Utiliza-se a combinação de leitura, escrita e execução para permitir ou negar acessos provenientes dos proprietários, grupos ou outros.

O *chown* é o comando, de acordo com Magrin (2006), que tem por finalidade alterar o proprietário do arquivo. Em outras palavras, é responsável por alterar o UID de um usuário em determinado arquivo para o UID de outro usuário.

Por fim, o comando *chgrp* tem por função “modificar a afiliação de um grupo a um arquivo, ou seja, alterar o dono do arquivo e o grupo. Muda as permissões em um arquivo selecionado, de forma que o arquivo pertença aos integrantes do grupo” (MAGRIN, 2006, p.51).

## 3.5 Senhas

Um dos maiores problemas em segurança digital trata-se das senhas, pois muitos usuários não dão o devido sigilo a elas e, em outros casos, o proprietário esconde as senhas a “sete chaves”, mas trata-se de uma senha fraca, ou seja, facilmente descoberta. Hunt (2004) explica que a senha independente do tamanho pode ser uma escolha ruim, como, por exemplo, nome de pessoas ou coisas, abreviações, datas conhecidas, palavras contidas no dicionário e, principalmente, senhas totalmente numéricas devem ser evitadas.

Existem muitas maneiras de capturar senhas, Morimoto (2006b) define, como uma das formas mais comuns o uso de programas *keytrap*, que tem como objetivo capturar tudo o que é digitado no teclado, armazenando esses dados em arquivos de texto que podem ser facilmente enviados por *email*. Morimoto alerta também sobre a possibilidade de ter a senha capturada em computadores de terceiros, principalmente ao se utilizar *lan houses* ou *cyber cafés*. Quando realmente for necessário utilizar senhas em outros computadores procure utilizar teclados virtuais para evitar o problema.

No Linux as senhas ficam armazenadas no arquivo */etc/passwd* (atualmente utiliza-se /etc/shadow) de forma criptografada. Mesmo que as senhas fiquem “embaralhadas” isso resulta em um problema, pois esse arquivo pode ser lido por qualquer usuário. Hunt (2004) esclarece o problema como suscetível a ataques de

dicionário, que consiste em criptografar uma lista de palavras e compará-las uma a uma com as senhas armazenadas no arquivo */etc/passwd*. Em caso de comparação positiva a senha foi descoberta. Para resolver esse problema, o autor sugere o uso de *shadow passwords* que consiste em proteger o acesso ao arquivo de senhas somente ao usuário *root*. Hunt conclui que, além de proteger o arquivo de senhas, o *shadow passwords* disponibiliza um controle contra envelhecimento de senha. Sempre que uma senha está em uso por um determinado período de tempo, o usuário é notificado a alterá-la.

Para garantir uma segurança ainda maior foram criadas as senhas chamadas *on-time passwords* que consiste, de acordo com Hunt (2004), em senhas que podem ser utilizadas apenas uma vez e após o seu uso ela é inutilizada. Esse procedimento é muito seguro, pois se alguém apropriar de sua senha, a mesma não será mais válida.

## 3.6 Ameaças e ataques

Sistemas informatizados estão sujeitos a ameaças diversas, seja ela acidental ou intencional. Independente da sua origem, esse problema pode colocar em risco toda integridade do sistema de uma empresa.

A ameaça “consiste em uma possível violação da segurança de um sistema”, (SOARES; LEMOS; COLCHER, 1995, p.448). Soares, Lemos e Colcher (1995) discorrem sobre as ameaças acidentais como sendo as que não tiveram uma associação premeditada, no entanto as ameaças intencionais são aquelas que tiveram essa finalidade. Os autores concluem que uma ameaça intencional concretizada configura um ataque.

Existem também as ameaças passivas e ativas. Segundo Soares, Lemos e Colcher (1995), as passivas são aquelas que não modificam as informações contidas no sistema, enquanto as ativas promovem alterações de alguma forma.

Dentre as ameaças, Hunt (2004) define as mais comuns como sendo:

- **Ameaça a dados secretos:** Ocorre quando as permissões de arquivos estão incorretas. Usuários sem permissões conseguem acesso a determinados arquivos que não deveriam;

- **Ameaça a integridade dos dados:** São modificações sem autorização em determinados arquivos. Geralmente é causada devido à concretização de uma ameaça a dados secretos;
- **Ameaça a disponibilidade dos dados:** Ocorre quando o acesso a dados é negado a determinado usuário que deveria ter essa permissão. Geralmente esse problema ocorre ao executar ataques de negação de serviço (*Denial of service* ou somente DOS).

Os ataques mais comuns de acordo com Soares, Lemos e Colcher (1995) são:

- **Personificação (*masquerade*):** É quando um usuário faz-se passar por outro para obter privilégios extras;
- **Modificações:** O conteúdo de uma mensagem é interceptado, alterado e reenviado em seguida de forma que o sistema não detecte essa alteração;
- **Recusa ou impedimento de serviço**: É quando um ataque impede que determinado serviço execute de forma correta as suas funções;
- **Ataques internos**: É quando usuários reais passam a se comportar de forma não coerente com suas atividades;
- **Armadilhas (*trapdoor*)**: É quando um serviço do sistema é modificado com intuito de realizar tarefas não autorizadas a partir de um determinado comando.

## 3.7 Monitoramento

O monitoramento do sistema é muito importante para garantir a segurança. Com verificação freqüente pode-se descobrir ataques contra o seu sistema e também descobrir brechas por onde podem ocorrer possíveis invasões. Hunt (2004) explica que, além de prevenir os ataques, é possível descobrir as invasões que foram bem-sucedidas.

O autor explora a utilização de alguns comandos básicos para tentar identificar quando algo anormal está acontecendo em seu servidor:

- *who:* utilizado para verificar quem está registrado no sistema e o que estão fazendo;

- *last:* verifica a regularidade com que o usuário registra no sistema;
- *ps:* utilizado para verificar os processos em execução;
- */var/log/secure*: arquivo utilizado para analisar os erros em tentativas de *login* no servidor;

> Não espere estes comandos para pegar um intruso no ato. Esteja atento para que, se seu sistema for quebrado, todos estes comandos provavelmente serão substituídos por versões alteradas, projetadas para esconder a atividade ilícita e o arquivo de registro provavelmente será privado da informação incriminadora.
> (HUNT, 2004. p.426)

### 3.7.1 Ferramentas de monitoramento

Para monitorar o seu sistema além dos comandos básicos do Linux, alguns sistemas de software são recomendados para garantir maior sucesso na proteção do sistema.

Monitorar as portas abertas no servidor pode evitar que invasores aproveitem essas brechas e as utilize para afetar seu sistema. O *Nmap* é um aplicativo desenvolvido justamente com essa finalidade. Morimoto (2006b) define o *Nmap* como um *portscan* (verificador de portas) que pode ser usado sempre que houver a necessidade de verificar rapidamente quais portas estão abertas em determinado computador. Morimoto salienta também que ter uma porta aberta não significa estar vulnerável a ataques, mas sim, que existe algum serviço em execução na porta em questão.

Apenas verificar as portas abertas não ajuda em muito a segurança do sistema, por isso ferramentas como o *Nessus* foram desenvolvidas. O *Nessus* é uma ferramenta de “auditoria muita usada para detectar e corrigir vulnerabilidades nos PCs (*Personal computers*) da rede local. Ele realiza uma varredura de portas, detectando servidores ativos e simulando invasões para detectar vulnerabilidades”. (MORIMOTO, 2006b, p.150). O autor conclui que o uso do *Nessus* aumenta o poder de monitoramento, pois ele tem a capacidade de procurar servidores ativos em portas diferentes das utilizadas normalmente, sendo assim, ele pode verificar uma vulnerabilidade em um servidor *Apache* rodando em uma porta diferente da *default* (padrão), nesse caso a porta 80.

O *Nessus* combina o uso do *Nmap* com suas funcionalidades para detectar as vulnerabilidades conhecidas. Morimoto (2006b) explica que, após os testes realizados, é exibido uma lista com todas as vulnerabilidades encontradas em cada máquina da rede.

Outra ferramenta muito utilizada para monitoramento é o *Wireshark* também conhecido como *ethereal.* Morimoto (2006b) define essa ferramenta como um *sniffer*[10] poderoso para capturar todo o tráfego da rede, visando obter informações sobre todos os pacotes que entram e saem de determinado computador da rede, facilitando a detecção de diversos tipos de *trojans*, *spywares* e acessos não autorizados no sistema. Muitas pessoas definem o *Wireshark* como ferramenta *hacker* devido a sua capacidade de capturar todos os dados que trafegam pela rede visando obter dados confidenciais.

A utilização dessas ferramentas permite, a partir de relatórios, encontrar possíveis brechas de segurança. Hunt (2004) salienta que esses relatórios são eficientes apontadores para problemas relacionados à segurança, contudo não deve-se confiar inteiramente, pois muitos problemas indicados podem não surtir efeito para seu sistema. Portanto, sempre realize uma análise para determinar o que realmente é relevante.

## 3.8 *Firewall*

O *firewall* é um sistema de segurança que tem seu objetivo focado em proteger as redes locais das redes externas, geralmente da Internet. Hunt (2004) define *firewall* como a sentinela que monitora todo o tráfego de rede antes de entrar ou sair de determinada rede local. Completando a idéia, Magrin (2006) afirma que, o *firewall* é planejado e implantado para aumentar a segurança das redes locais em instituições que utilizam conexões de rede inseguras. A figura 1 ilustra o esquema de funcionamento básico de um *firewall*.

[10] *Sniffer* ou farejador, segundo Assunção (2008), são os programas que analisam e capturam todo o tráfego de entrada ou saída da rede ou sistema.

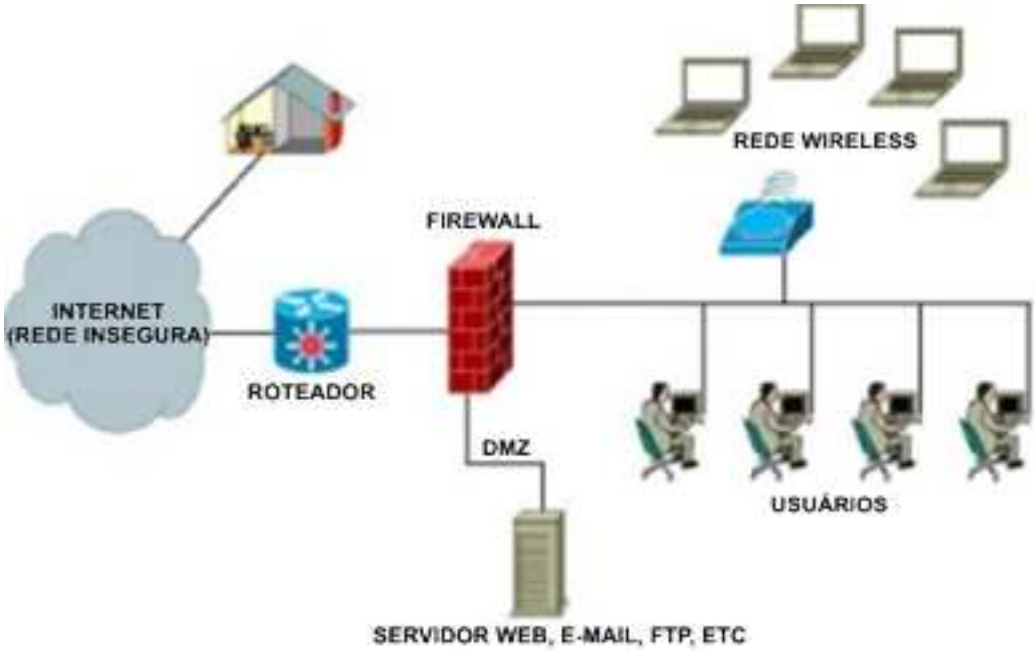


Figura 1 – Esquema básico de um *firewall*
Fonte: INTRON, 2008, adaptado.

Quando as empresas decidem implantar um sistema de segurança composto de um *firewall,* o Linux geralmente é uma boa alternativa. Hunt (2004) explica que o pingüim[11] fornece várias ferramentas de filtragem de tráfego que são essenciais para criação de um sistema de segurança. Magrin (2006) informa que o *kernel* do Linux, na sua versão 2.2, provia de um sistema chamado *ipchains*, enquanto as versões mais recentes (atualmente versão 2.6) contam com o sistema chamado de netfilter (*iptables).*

*Ipchains*, de acordo com a Revista Linux (2000), é uma ferramenta para filtragem dos dados que trafegam na rede. É utilizado para controlar todos os pacotes trafegados, impedindo acesso não autorizado a *sites* além de evitar que pessoas sem autorização fiquem investigando a rede local.

O *iptables* tem o mesmo propósito do *ipchains,* de certo modo trata-se de sua evolução, sendo substancialmente “mais ágil, seguro e robusto”. (MAGRIN, 2006, p.125).

O Linux, com o uso de pequenas configurações, abre um leque de opções referentes à segurança. Hunt (2004) salienta a possibilidade de combinar a

[11] pingüim além de ser o mascote do Linux, também é utilizado para referenciá-lo.

capacidade de roteamento com recursos de filtragem do *iptables*. Com dada combinação, é plausível filtrar todo o tráfego de dados que chegam à interface de rede de um servidor Linux antes de ser encaminhado às aplicações que executam nesse servidor. Com essa possibilidade, pode-se construir um *firewall* no mesmo servidor que controla o acesso aos serviços de rede.

Para configurar o *firewall* no Linux utilizam-se linhas de comando, isto é, toda regra deve ser digitada em um terminal (*shell*). Magrin (2006) explica que as configurações de acesso são perdidas durante a reinicialização do sistema. O Linux não disponibiliza um arquivo específico que armazene essas configurações. Para evitar digitar as regras sempre que o sistema é iniciado, a autora sugere a criação de um *script* contendo as regras necessárias, e sua posterior adição em um arquivo de inicialização do Linux, forçando assim a inicialização automática das regras do *firewall* junto ao sistema. Vale ressaltar que o *netfilter* dispõe de uma ferramenta para salvar suas configurações (iptables-save), facilitando esse procedimento.

## 3.9 Segurança *Wireless*

Ao utilizar redes *wireless* (sem fio) nas empresas aumenta-se substancialmente o risco de acesso não autorizado ao sistema. “Não existe como impedir que o sinal se propague livremente pelas redondezas [...], de forma que a única forma eficaz de proteção é encriptar toda a transmissão, fazendo com que as informações capturadas não tenham serventia”. (MORIMOTO, 2006b, p. 166 *sic*).

Reforçando a idéia sobre a baixa segurança em rede *wireless*, Horton (2003) explica que as formas de segurança em redes sem fio tiveram seu desenvolvimento tardio e foram aprimoradas de forma marginal, provendo de falhas de segurança causadas devido a mecanismos internos inadequados. Como conseqüência disso, a rede é exposta a vulnerabilidades e riscos graves.

### 3.9.1 *Wep* e *Wpa*

Ao utilizar criptografia de dados para as conexões *wireless*, geralmente emprega-se o uso das tecnologias *Wep (Wired Equivalent Privacy)* e *Wpa (Wi-Fi*

*Protected Access*), porém atualmente a utilização do *Wpa* está mais difundido devido às inúmeras falhas de segurança no protocolo *Wep*. Farias (2006) define que o *Wep* não é indicado para utilização em redes corporativas devido a utilização de chave de segurança estática, facilitando dessa forma a obtenção da senha de segurança. Farias ainda conclui que o *Wpa,* devido ao uso do *TKIP* (*Temporary Key Integration Protocol*), fornece uma codificação a partir de chaves dinâmicas e autenticação mútua[12], aumentando assim a segurança do sistema.

Horton (2003) alerta que, ao se utilizar chaves *Wep*, o atacante que conseguir capturar pacotes da rede, pode analisá-los devido a falhas existentes no algoritmo de criptografia do *Wep*. Morimoto (2006b) conclui que até as versões não vulneráveis do *Wep* podem ter suas chaves quebradas caso ocorra à possibilidade de capturar grande quantidade de pacotes da rede. O uso do *kismet* é uma das maneiras mais comuns de capturar pacotes da rede.

Uma das principais diferenças entre *Wep* e *Wpa* é o vetor de inicialização[13]. Morimoto (2006b) explica que enquanto o *Wep* utiliza apenas 24 *bits*, o *Wpa* conta com um vetor de 48 *bits,* possuindo também um conjunto de proteções contra possíveis ataques. Morimoto alerta que, embora o *Wpa* seja fundamentalmente mais seguro que o *Wep*, também existe a possibilidade de descobrir a sua senha pela obtenção de pacotes capturados da rede, entretanto esse procedimento só é aplicável caso utilize-se uma senha pequena ou uma palavra contida no dicionário.

### 3.9.2 *Kismet*

No intuito de invadir redes sem fio, pessoas mal intencionadas utilizam ferramentas para tentar burlar a segurança. O *kismet* é uma delas, e segundo Morimoto (2006b), é uma ferramenta para verificar a situação das redes *wireless*, pois ela tem a capacidade de pesquisar as redes sem fio ao redor e detectar quais

[12] Autenticação mútua ocorre quando o cliente autentica o *host* e o *host* autentica o cliente.

[13] Segundo GTA (2008), o vetor de inicialização é uma área que é enviada junto ao cabeçalho da mensagem de forma não codificada, para que a estação que a receba coloque o vetor em sua chave *WEP* ou *WPA*, conseguindo assim decodificar a mensagem enviada.

canais estão mais congestionados, porém muitos usuários utilizam essa ferramenta para invasão de redes.

Horton (2003) explica que o *kismet* funciona com um *sniffer* para capturar pacotes na rede que se deseja atacar. Após ser capturado uma série de pacotes, utiliza-se uma ferramenta chamada *Wepcrack.pl* para analisar a criptografia contida nos pacotes, realizando uma engenharia reversa com intuito de descobrir a senha de acesso a rede. Morimoto (2006b) ainda complementa dizendo que utilizando o um aplicativo chamado *aircrack,* também é possível descobrir a chave de encriptação da rede.

## 3.10 Sistemas de arquivos

O sistema de arquivos é uma das partes do sistema operacional que mais interage com o usuário, pois a maior parte do tempo o usuário está em contato com ela. Segundo Mello (2004), esse sistema é a união de elementos que através de vários métodos ficam acessíveis para manipular estruturas que estão compartilhadas com o *kernel*.

A finalidade dos *filesystems* (sistema de arquivos) é oferecer uma maneira de trabalhar com informações contidas nos dispositivos. Sua “funcionalidade básica está em formar estruturas lógicas na memória principal a fim de mapear dispositivos de armazenamento e seus respectivos conteúdos”. (JAMIL; GOUVÊA, 2006, p.20). Mello (2004) conclui que o sistema de arquivos é o responsável em manter uma interface entre a estrutura dos dados guardados nos dispositivos de armazenamento e as aplicações do sistema.

Ao realizar a instalação do Linux, é possível escolher diversos sistemas de arquivos diferentes, recomenda-se utilizar os que contam com a ferramenta *journaling* (os mais comuns são *JFS*, *ReiserFS, XFS* e *ext3*), para evitar possíveis perdas de dados.

O *journaling* é uma função do sistema de arquivos responsável em reservar uma área no disco para guardar todas as alterações que forem feitas nos arquivos. Esse procedimento eleva o nível de confiabilidade do sistema em caso de falhas, pois ocorrendo erros é possível retornar a um estado consistente anterior, evitando

dessa forma que arquivos sejam corrompidos, essas idéias são defendidas por Mello (2004).

Os sistemas de arquivos do Microsoft Windows não provêm um sistema de recuperação instantânea dos arquivos, mas conta com duas ferramentas utilizadas com o mesmo propósito, isto é, o de manter os dados íntegros. O *CHKDSK* e o *ScanDisk* são utilizados sempre que erros em arquivos forem detectados, realizando então uma verificação no disco em busca de possíveis falhas.

# 4 BACKUP

Muitas vezes o administrador de um servidor está tão preocupado em deixar o seu sistema seguro contra invasões de terceiros que acaba esquecendo que a perda de um dos discos rígidos do seu servidor pode causar mais prejuízos que um invasor no sistema. Devido a essa preocupação devem ser freqüentemente realizados *backups* (cópias de segurança) dos dados armazenados, evitando assim possíveis perdas de informações. A “cópia de segurança é a melhor forma de prevenção e recuperação das informações, já que os dados podem voltar fielmente para o disco, quando e se for necessário”. (FIALHO Jr., 2007, p.6).

Efetuar uma cópia de segurança não garante que seus dados fiquem seguros. Fialho Jr. (2007) explica que as mídias utilizadas para armazenar os *backups* devem ser guardadas em locais seguros, protegidos de poeira, umidade e, até mesmo, de pessoas estranhas, evitando assim que os *backups* se percam ou fiquem danificados. O autor conclui ainda que os *backups* também devem receber identificação por nome e data de realização, pois quando houver a necessidade de recuperar algum arquivo, a localização seja simples. Para realizar *backups* em Linux podemos utilizar diversos aplicativos, dentre eles *Amanda*, *FileBackup*, *Backup Mananger*, entre outros.

Outro fator importante é a privacidade dos dados que, segundo Symantec (2007), é um fator crucial criptografar os dados, pois como dito anteriormente se houver uma apropriação indevida de uma mídia de armazenamento e os dados da mesma estiverem criptografados pode dificultar ou até mesmo evitar grandes problemas à empresa. Symantec ainda conclui que os *backups* remotos também podem expor os dados da empresa ou de clientes, por isso, utilizando sistemas de criptografia, obtêm-se uma nova camada de segurança às informações.

Atualmente, as mídias de *backup* mais comuns em microempresas são os CDs (*Compact discs*), DVDs (*Digital Video Disks* ou *Digital Versatile Disks*), *pen drives* e cartões de memória. Essas mídias são recomendadas para as instituições que dispõem de volume de dados reduzidos. Porém, muitas empresas possuem grande volume de informações e então é recomendado o uso de fitas magnéticas ou discos rígidos externos devido a alta capacidade de armazenamento.

## 4.1 Compressão de dados

Espaço para armazenamento das informações é sempre um problema ao se lidar com cópias de segurança. Geralmente as mídias mais utilizadas contam com baixa capacidade de armazenamento dificultando tal atividade. Para minimizar esse problema a compressão dos arquivos é uma das melhores alternativas.

> A compressão de dados é uma técnica inventada nos anos 50. Ela baseia em eliminar os bits redundantes para o entendimento do computador, ou seja, aqueles bits presentes nos dados que são necessários somente para o entendimento humano restando apenas os dados realmente necessários para o computador. É como se nós chamássemos o “sim” e o “não” apenas com um bit 1 e 0. Concluímos que dessa maneira economizaremos espaço de memória. (JAMIL; GOUVÊA, 2006, p.134).

A compressão dos dados pode ser realizada por diversos aplicativos, Jamil e Gouvêa (2006) definem como os mais utilizados:

- *gzip*: É o comando mais utilizado em ambiente Linux e também o que contém a maior taxa de compressão. Seu funcionamento constitui na substituição do arquivo original por outro com a extensão “.*gz*”.
- *zip*: É o comando mais comum em ambiente Windows, ao utilizá-lo aumenta a portabilidade do *backup,* pois diversos sistemas operacionais contam com ferramentas para manipulá-los. Seu funcionamento constitui na criação de uma cópia compactada, mantendo os arquivos originais intactos.

Jamil e Gouvêa (2006) completam que existem níveis de compactação para os arquivos. Esse nível é dado por uma escala de 1 a 9 que define a velocidade e a taxa de compressão a ser utilizada. O valor 1 é o mais rápido, porém pouco eficaz; Enquanto o valor 9 é muito eficaz, contudo maior tempo é despendido para realização do mesmo procedimento.

Uma vantagem do *zip* é a possibilidade de proteger os arquivos por senha. Segundo Granneman (2006), ele disponibiliza duas opções, que concatenadas ao comando principal (*zip*), incluem essa segurança aos arquivos. A primeira é a opção “-e”, sempre que incluída essa opção, será solicitado a digitação da senha de

encriptação, a segunda opção é a opção “-P” você informa a senha como parâmetro para o comando, porém essa forma não é indicada, pois a senha fica visível possibilitando a sua visualização por outros

## 4.2 Forma de realização

Ao realizar cópias de segurança deve ser analisado qual tipo de *backup* é mais indicado. Ruschel (2007) explica que as formas mais utilizadas são:

- Normal: Nesse método todos os dados são copiados, desconsiderando data e horário de criação ou modificação;
- Incremental: Nesse método somente os arquivos criados ou modificados após o ultimo *backup* serão copiados.
- Diário: No *backup* diário são copiados todos os dados que foram criados ou modificados no dia da realização da cópia.

A maneira mais simples para realizar *backup* no Linux é utilizando o comando *tar* que, segundo GNU (2008), trata-se de um programa que oferece a possibilidade de criar e manipular pacotes de arquivos, direcionando-os para outros programas ou arquivos. Geralmente o uso do *tar* é concatenado com o *gzip* ou outra ferramenta de compactação, originando um arquivo de *backup* com tamanho reduzido. Esse método resulta em um pacote semelhante ao criado pelo *zip*, porém com um maior grau de compressão. Assim, o arquivo resultante pode ser facilmente copiado para diversos tipos de mídia de armazenamento.

## 5 CONSIDERAÇÕES FINAIS

O Linux, por ser um software distribuído sobre licença GPL garante aos usuários uma liberdade muito grande, pois essa licença permite a possibilidade de utilizar o sistema para qualquer finalidade e permite também que o usuário o modifique de forma a atender às suas necessidades. Isso torna economicamente viável às microempresas, pois apenas uma versão do sistema se faz necessária para a utilização em todos os servidores da sua empresa, bastando apenas algumas pequenas modificações.

Em casos mais específicos em que a empresa adote versões não gratuitas do Linux, seu custo também tende a ser reduzido, pois de acordo com os termos descritos na GPL, qualquer cópia do sistema é permitida, sendo assim, a empresa irá necessitar adquirir somente uma licença e os demais equipamentos podem receber cópias desse sistema, sem infringir as regras contidas na licença GPL. Vale ressaltar que qualquer condição presente neste tipo de licença, pode ser modificada desde que em comum acordo com o autor do software. Portanto antes de realizar modificações em seu sistema, certifique-se com o distribuidor a respeito dessa possibilidade.

Para garantir o bom funcionamento dos equipamentos, algumas empresas contratam serviços de suporte com a distribuição que mantém a versão do Linux instalada. Esse procedimento pode elevar o custo de manutenção do sistema da empresa, porém se realizarmos uma comparação com outros sistemas, o Linux em vários casos ainda continuará sendo uma boa escolha, pois muitos sistemas que não utilizam licença GPL (ou semelhante) disponibilizam suporte gratuito. Em contrapartida, cobram por sistema instalado, em empresas que dispõem de vários equipamentos, pode gerar um custo superior ao suporte não gratuito praticado por empresas que adotam o uso do Linux.

Outro fator determinante para adoção do Linux no ambiente de microempresas é a sua estabilidade, pois conta com inúmeros recursos que garantem o funcionamento contínuo do sistema. Morimoto (2006a) explica que o núcleo do Linux é extremamente estável, porém as aplicações que nele são executadas não possuem tal garantia. E para combater justamente os problemas acarretados por essas aplicações, o Linux provê de inúmeras ferramentas que visam minimizar esse problema.

Como já comentado anteriormente a reinicialização do sistema só ocorre em casos extremos. Na maioria das vezes aplicativos como o *xkill, killall* e *kill* são suficientes para a solução do problema. Isso é, em caso de travamento, o usuário pode utilizar essas ferramentas para finalizar a aplicação problemática mantendo as demais funções do sistema trabalhando sem problemas. E em casos de problemas ainda mais complexos, a reinicialização do ambiente gráfico se torna uma alternativa.

Outro ponto que garante o funcionamento contínuo do sistema é a prioridade de processos, que evita que diferentes aplicações acessem determinado recurso do sistema ao mesmo tempo, disponibilizando o recurso para a aplicação que tiver maior prioridade e, em caso de prioridades iguais, o programa que foi iniciado primeiro recebe o recurso.

Por fim tem-se a segurança que talvez ela seja o fator determinante para escolha do Linux, pois fornece várias ferramentas de filtragem de tráfego que são essenciais para criação de um sistema de segurança, o que o torna indicado para a utilização em *firewalls*. Seu núcleo inclui uma ferramenta chamada *iptables* (versões mais antigas do Linux utilizam *ipchains*) que funciona como um filtro para todos os pacotes que por ele trafegam. Com a utilização de regras apropriadas pode bloquear ou permitir determinado pacote, aumentando assim a segurança do sistema.

Existem inúmeras ferramentas para monitorar o funcionamento do sistema. Elas podem determinar qual porta do servidor está aberta e descobrir se existe uma vulnerabilidade conhecida. Esse procedimento permite que o administrador de rede conheça suas falhas e defina a correção adequada, evitando que tentativas de ataques obtenham sucesso. Por exemplo, alguns *worms* podem ser identificados com esse processo.

Outro fator importante de segurança são as permissões de arquivos. Por exemplo, o Linux não define qual arquivo é executável com base na extensão, ele utiliza a combinação de leitura, escrita e execução para permitir ou negar acessos provenientes do proprietário, grupo ou outros. Esse procedimento dificulta o desenvolvimento de vírus, *worms* e *trojans,* pois mesmo que o aplicativo malicioso esteja fisicamente no sistema, teria que ser atribuído a permissão adequada à ele para que o sistema seja afetado.

Manter a integridade dos dados também deve ser levado em consideração. Para evitar que arquivos fiquem danificados devido a erros, alguns sistemas de

arquivos do Linux contêm uma ferramenta chamada *journaling* que, ocorrendo falhas na gravação, o sistema retorna à uma posição anterior válida. Por isso, sempre ao realizar uma nova formatação, escolha um sistema de arquivos que disponibilize esse recurso.

Outra forma para garantir a integridade dos dados é o *backup* que, mesmo ocorrendo defeito físico ou alteração dos arquivos de forma errônea, permite retornar a um estado anterior válido, garantindo novamente a disponibilidade dos dados.

Embora o Linux seja um software que tem capacidade para atender empresas com as mais variadas características, este estudo abordou a utilização e adequação às necessidades das microempresas, abordando a economia, a estabilidade e a segurança adquiridas em sua utilização. Algumas dessas características são comuns, permitindo sua utilização em outras empresas e também em outros sistemas operacionais.

# REFERÊNCIAS

ASSUNÇÃO, Marcos Flávio Araújo. **Guia do Hacker Brasileiro**. Santa Catarina: VisualBooks**,** 2008

BRASIL, Decreto 5.028, de 31 de março de 2004. Altera os valores dos limites fixados nos incisos I e II do art. 2º da Lei nº 9.841 de 5 de outubro de 1999, que instituiu o Estatuto da Microempresa e da Empresa de Pequeno Porte. **Planalto**. Brasília, DF, 31 mar 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/ _ato2004-2006/2004/decreto/D5028.htm>. Acesso em: 03 set 2008.

DAZS, Ragen. **Entendendo um pouco sobre daemons**. Viva o Linux, 05 jul 2004. Disponível em: <http://www.vivaolinux.com.br/artigo/Entendendo-um-pouco-sobre-os-daemons>. Acesso em: 20 set 2008.

FARIA, Alessandro de Oliveira. **Dominando o apt-get no Conectiva**. Viva o Linux**,** 17 fev. 2004. Disponível em: <http://www.vivaolinux.com.br/artigo/ Dominando-o-aptget-no-Conectiva>. Acesso em: 14 set. 2008.

FARIAS, Paulo Cesar Bento. **Treinamento profissional em redes wireless**. São Paulo: Digerati Books, 2006.

FIALHO Jr., Mozart, **Guia essencial do backup**. São Paulo: Digerati Books, 2007.

GTA. **Segurança Wireless**. Disponível em: <http://www.gta.ufrj.br/~rezende/ cursos/eel879/trabalhos/srsf1/wep.htm>. Acesso em: 26 out. 2008.

GNU. **General public License**. GNU, 29 jun. 2007. Disponível em: <http://www. gnu.org/copyleft/gpl.txt>. Acesso em: 20 set. 2008.

GNU. **Tar**. GNU, 14 abr. 2008. Disponível em: < http://www.gnu.org/software/tar/>. Acesso em: 28 out. 2008.

GRANNEMAN, Scott. **Linux Phrasebook**: Essential code and commands. Indiana: Sams Publishing, 2006.

HORTON, Clinlon Mugge. **Hack Notes Segurança de redes**. Rio de Janeiro: Elsevier. 2003

HUNT, Craig. **Linux**: servidores de rede. Rio de Janeiro: Ciência Moderna, 2004.

INTRON. **Características técnicas do firewall**. INTRON. Disponível em: <http://www.intron.com.br/index.php?id=firewall/firewall2.php>. Acesso em: 04 nov. 2008.

JAMIL, George Leal. **Linux para principiantes:** De iniciante a intermediário em tempo recorde. Rio de Janeiro: Axcel Books, 1999.

JAMIL, George Leal; GOUVÊA, Bernardo Andrade. **Linux para profissionais:** Do Básico à Conexão de Redes. Rio de Janeiro: Axcel Books, 2006.

JARGAS, Aurélio Marinho. **Shell Script Profissional**. São Paulo: Novatec, 2008.

MCCARTY, Bill, **Learning Red Hat Linux:** A Guide to Red Hat Linux for New Users. Sebastopol: O´Reilly: 2003.

MAGRIN, Maria Heloisa. **Guia Profissional Linux**. São Paulo: Digerati Books, 2006.

MARQUES, José Alves; GUEDES, Paulo. **Fundamentos de sistemas operativos:** informática e computadores. 4. ed. Lisboa: Presença, 1998.

MATOS, Francisco Jarbas Teixeira. **Entendendo os Recursos do Linux.** São Paulo: Digerati Books, 2007.

MELLO, Leonardo Garcia de. **Validação experimental de sistema de arquivos baseados em *journaling* para o sistema operacional Linux**. 2004. 97 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade federal do Rio Grande do Sul, Porto Alegre, 2004. Disponível em: <http://www.bibliotecadigital.ufrgs.br/da.php?nrb=000448567&loc=2005&l=fdfea892cf3be797 >. Acesso em: 14 out. 2008.

MORIMOTO, Carlos Eduardo. **Linux Entendendo o Sistema:** Guia Prático. Porto Alegre: Sul Editores, 2006a.

MORIMOTO, Carlos Eduardo. **Linux Redes e Servidores:** Guia Prático. 2. ed. Porto Alegre: Sul Editores, 2006b.

Red Hat. **RedHat Online Shop**. Disponível em: <https://www.europe.redhat.com/shop/en/>. Acesso em: 17 dez. 2008.

Revista Linux. **IPChains: Conexões sem fronteiras**. Revista Linux, Dez. 2000. Disponível em: <http://augustocampos.net/revista-do-linux/012/index.html>. Acesso em: 13 out. 2008.

RUSCHEL, André Guedes. **Do cabeamento ao servidor**. São Paulo: Brasport, 2007.

SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sérgio. **Redes de computadores**: das LANs, MANs e WANs às redes ATM. 2. ed. Rio de Janeiro: Elsevier, 1995.

SYMANTEC. **Encrypting Critical Backup Data**. Symantec, Jan. 2007. Disponível em: http://www.symantec.com/business/resources/articles/article.jsp?aid=encrypting_critical_backup_data. Acesso em: 16 nov. 2008.

TANENBAUM, Andrew Stuart. **Sistemas operacionais modernos.** 2. ed. São Paulo: Prentice Hall, 2003.