Administração de Sistemas Operacionais Unix (I)

Centro de Informática de Ribeirão Preto - CIRP

MSc. Eng. Ali Faiez Taha

Sumário

Capítulo 1

Introdução

O Sistema Operacional **Unix** é bastante complexo e apresenta características específicas diversas.

O administrador de S.O. Unix deverá conhecer Linguagens de Programação (C, C++, etc), shell scripts, Redes e um pouco de Hardware.

No cotidiano o administrador Unix utiliza softwares, shell scripts e ferramentas administrativas.

Muitas tarefas administrativas são efetuadas em shell scripts.

Criar, modificar e adaptar **shell** scripts para determinadas situações faz parte da administração do S.O. Unix.

É bastante comum encontrar Softwares específicos para administração de serviços dedicados do servidor.

É fundamental conhecer o Hardware do equipamento servidor, ou Desktop, seus componentes, modelos e características técnicas.

A administração de um servidor baseado em Sistema Operacional Unix se resume em três pontos:

1.1 Instalação

Instalar um S.O. Unix ou algum S.O. baseado em Unix já foi uma tarefa difícil.

Com a popularização do Linux esta tarefa se tornou bastante simples.

Conhecer a arquitetura do servidor que vai ser usado, seu elementos de hardware e suas características é extremamente fundamental.

O administrador deve escolher o S.O. Unix que mais se adapta às suas necessidades.

Deve levar em conta as necessidades de Software e Hardware e, assim, dimensionar o servidor.

Numa instalação de Linux já há opções de tipos diferentes de instalação.

Dimensionar a instalação e procurar conhecer o significado de cada item que será instalado.

Conhecendo o Hardware existente não vai haver problemas no momento da instalação dos drivers de cada item.

Não são raras as vezes que certo elemento de Hardware tem que ser trocado por causa da falta de driver específico.

Não se aconselha a instalação de todos os pacotes de uma ditribuição Linux.

Escolha a distribuição Unix/Linux que vai ser instalada na plataforma que você possui.

Se não houver uma distribuição Linux para a sua plataforma específica, outro UNIX deve ser escolhido.

Um bom planejamento deve ser estabelecido e seguido. Só assim uma instalação Unix/Linux pode ser bem efetuada.

1.2 Operação

O administrador e os usuários de um S.O. Unix devem conhecer os recursos oferecidos pelo servidor, Softwares instalados, limites estabelecidos (espaço em disco, serviços de rede, etc), periféricos instalados, sistemas de backup, etc.

A operação do Unix/Linux hoje é bastante facilitada devido ao grande número de aplicativos disponíveis.

Numa instalação Linux o número de aplicativos encontrados hoje supera o que se encontra em equipamentos servidores profissionais, tais como Sun, IBM, Silicon Graphics, etc.

Operar um S.O. de uma linhagem profissional não é muito diferente de se operar um Linux. Com um bom treinamento em Linux, o usuário não terá dificuldades em operar máquinas e Unix mais profissionais, e mais caros.

O administrador de um S.O. Unix deve prezar pelo bom funcionamento do servidor e para isso ele deve se manter sempre atualizado, participar de treinamentos e cursos de aperfeiçoamento.

Os usuários precisam conhecer seu limites também, participar de treinamentos e conhecer as ferramentas que utilizam, seus limites, características e detalhes envolvidos.

A cada ferramenta ou recurso novo instalado o usuário final deve ser informado.

1.3 Manutenção

A atualização das instalações é fundamental.

Manter um S.O. Unix funcionando corretamente, ou próximo disso, é uma tarefa que exige muita prática e bom conhecimento do sistema todo.

Os itens de Software e Hardware necessitam constantemente de manutenção e atualização.

Não é somente o hardware que apresenta problemas. Software também apresenta defeitos, falhas e erros.

Atualizar os Softwares instalados é hoje tarefa considerada fundamental para um administrador de S.O. Unix. Se os Softwares são proprietários ou não, sua atualização sempre deve ser feita.

No caso de Software Livre a atualização também deve ser feita.

Atualização de Kernel e dos serviços, dos aplicativos, políticas de troca de senha, backup, redes, etc.

O administrador deve estar atualizado e conhecer os meios para fazer as atualizações necessárias, fazer o gerenciamento dos serviços e recursos do servidor.

Gerenciar servidores de E-Mail, FTP, WEB, Serviços de Rede, etc. Atualizar estes serviços constantemente. Administrar contas de usuários, sistemas de Backup e instalação de Hardware e Softwares.

É aconselhável participar de listas de discussão, troca de informações e treinamento.

Na instalação de programas proprietários os recursos exigidos merecem atenção especial, tais como licença, senhas e documentação.

A análise dos recursos disponíveis para atender a instalação de novos elementos de Software e Hardware deve ser criteriosa.

Deve-se adotar critério na especificação de novas necessidades, recursos, adaptações, expansões, Backup, e segurança.

Todo equipamento precisa de manutenção. A manutenção preventiva deve ser efetuada constantemente.

1.4 Instalação do Sistema Operacional Debian Linux

Ao se instalar um S.O. Linux, todo o Hardware compatível com o Linux deverá ser examinado. Muitos elementos de Hardware já são suportados pelos Linux existentes. Praticamente todos os fabricantes de Hardware já disponibilizam os drivers necessários para o Linux.

Os elementos de Software que compõem uma distribuição Linux devem ter suporte aos elementos de Hardware.

Se isso ocorre numa distribuição Linux, então ela é compatível com a arquitetura, seja ela **i386, Alpha, Sparc, m68k**, etc.

O URL abaixo mostra um completo guia de compatibilidade de Hardware para Linux:
http://www.tldp.org/HOWTO/Hardware-HOWTO/

As preocupações estão mais voltadas aos seguintes itens: CPU : **Intel, AMD, Cyrix**, etc
Motherboards : **Intel, Soyo, Supermicro, Tyan,** etc
Placas de vídeo e de rede de diversos fabricantes.

Ao se escolher os elementos de Hardware vale a pena uma consulta às listas de compatibilidade com o S.O. Linux. Na WEB há muitas referências para consulta.
http://www.linorg.cirp.usp.br/Debian.refs/Install/ch-hardware-req.pt.html

Os requerimentos mínimos necessários devem ser atendidos, tais como: espaço em disco e quantidade de memória.

Servidores ou Desktops voltados a aplicações mais dedicadas e para realização de tarefas bem específicas devem ser mais elaborados em relação a Hardware.

No mercado já é possível se encontrar equipamentos com boa capacidade de processamento. Vale a pena uma análise dos produtos encontrados e fazer uma comparação ao produto que ser pretende ter. As chamadas **máquinas de grife** apresentam certas vantagens aos equipamentos **convencionais**, seja em relação aos preços e qualidade dos elementos de Hardware.

1.5 Modos de Instalação

Instalar um S.O. Linux hoje se tornou uma tarefa bastante simples. Dependendo da distribuição do Linux a complexidade pode ser resumida a simples e elementares 'clicks' ou preenchimentos de alguns pequenos menus de configuração.

A instalação do Debian Linux pode ser efetuada via Diskettes, CD-ROM, Disco rígido, HTTP e FTP.

O URL abaixo mostra os preparativos necessários e os métodos de instalação do Debian Linux:
http://www.linorg.cirp.usp.br/Debian.refs/Install/ch-preparing.pt.html

O mais comum hoje em dia é fazer uma instalação via CD-ROM, onde os requisitos mínimos têm que ser atendidos:

BIOS com recursos de Boot pelo CD-ROM além de um bom drive de CD-ROM, boa quantidade de memória e elementos de Hardware de boa qualidade.

A instalação via diskettes também é bastante utilizada e vai exigir apenas um Diskette, um drive de CD-ROM, um disco rígido com a distribuição Linux e, possivelmente, uma conexão com rede TCP/IP.

Se no equipamento já há uma particção com o S.O. Windows, é aconselhável fazer um Backup desta partição. Assim, o equipamento pode funcionar em dual-boot. É aconselhável que se tenha uma Disco rígido para cada Sistema Operacional.

Depois de analisado todo o Hardware e exigências de Software, o próximo passo é instalar o Debian Linux.

Os URLs abaixo mostram os passos necessários para efetuar a instalação:
1 - Obtenção da mídia de instalação do S.O. Linux Debian:
http://www.linorg.cirp.usp.br/Debian.refs/Install/ch-install-methods.pt.html
2 - Métodos de instalação:
http://www.linorg.cirp.usp.br/Debian.refs/Install/ch-rescue-boot.pt.html

1.6 Esquema de partições

É necessário pelo menos uma partição para se instalar o Linux. Se houver mais partições a instalação fica melhor. A vantagen de se ter mais de uma partição se resume na facilidade de operação e manutenção do S.O. (correção de falhas nas partições). Se houver falhas em alguma partição, a mesma pode ser isolada e trocada por outro disco evidamente particionado.

Dimensionar as partições é de extrema importância, pois, para muitas aplicações, é do espaço disponíveisem disco que oAdministrador deverá dimensionar e administrar seu servidores. Os tamanhos devem ser bem dimensionados para se evitar problemas no futuro. Hoje os discos são baratos, não há mais a necessidade de se economizar em discos.

Algumas considerações importantes tratando de diretórios e partições:

A partição raíz / deve sempre conter fisicamente **/etc, /bin, /sbin, /lib e /dev** caso contrário você não será capaz de inicializar.

São necessários tipicamente 100MBytes para a partição raíz, mas isto pode variar.

/usr: todos os programas (**/usr/bin**), bibliotecas (**/usr/lib**), documentação (**/usr/share/doc**), etc., estão neste diretório. Esta parte do sistema de arquivos precisa da maioria do espaço de disco. Deve-se separar pelo menos 500MBytes de espaço em disco. Se quiser instalar mais pacotes, será necessário aumentar o tamanho desta partição.

/home: cada usuário colocará seus dados em um subdiretório deste diretório. O tamanho desta partição depende de quantos usuários estarão utilizando o sistema e quais arquivos são armazenados em seus diretórios. Dependendo da utilização, deve-se dimensionar corretamente esta partição. Estabelecer políticas de quota, que definirá um espaço em disco para cada usuário.

/var: todos os dados variáveis tais como logs, e-mails, web sites, APT's cache, etc. serão colocados neste diretório. O tamanho deste diretório depende em especial de cada usuário, mas para a maioria das pessoas ele será recomendado pela ferramenta de gerenciamento de pacote. Se estiver para fazer uma instalação completa de apenas tudo que a **Debian** tem para oferecer, tudo em uma seção, a escolha de 2 a 3 gigabytes de espaço para **/var** deverá ser suficiente. Para servidores, os tamanhos das partições devem ser bem dimensionados levando em consideração o volume de dados que deverá ser armazenado, políticas de Backup, remoção de arquivos antigos, etc.

/tmp: Se um programa cria dados temporários eles provavelmente serão gravados em **/tmp**. O tamanho de 20 a 100 MBytes deve ser o bastante. Depende muito das aplicações e da utilização do servidor.

swap: Área de troca do disco. Caso a memória esteja lotada, uma área de disco pode ser usada temporariamente para troca de dados. Geralmente é do dobro da memória RAM instalada, mas, nem sempre é regra. Se for muito grande, diz-se que há muito acesso a disco para swap, o que pode comprometer a performance do Sistema Operacional. Deve ser bem dimensionada para se evitar demasiado acesso a disco.

Todas as partições devem ser bem dimensionadas levando em conta a aplicação do Servidor Unix.

Capítulo 2

Sistema Operacional Debian Linux

A **Debian** é a distribuição que mais cresce no mundo, cada versão é somente lançada após rigorosos testes de segurança e correção de falhas fazendo desta a mais segura e confiável dentre todas as outras distribuições Linux.

É reconhecida como a mais segura, maior e atualizada mais freqüentemente entre as outras distribuições Linux, além de ser a única sem fins comerciais.

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-deb.html)

2.1 Estrutura básica de diretórios do Sistema Linux

O sistema **GNU/Linux** possui a seguinte estrutura básica de diretórios:

/bin - Contém arquivos programas do sistema que são usados com freqüência pelos usuários.

/boot - Contém arquivos necessários para a inicialização do sistema.

/cdrom - Ponto de montagem da unidade de CD-ROM.

/dev - Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

/etc - Arquivos de configuração de seu computador local.

/floppy - Ponto de montagem de unidade de disquetes

/home - Diretórios contendo os arquivos dos usuários.

/lib - Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.

/lost+found - Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2.

Cada partição possui seu próprio diretório lost+found.

/mnt - Ponto de montagem temporário.

/proc - Sistema de arquivos do **kernel**.

A sessão do Sistema de Arquivos **/proc** não contém arquivos reais. Contém informações do **status** do sistema.

Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificam o funcionamento de dispositivos do sistema através da alteração em seus arquivos.

Exemplos:

Local	Informação
/proc/[número]	Processo específico que está em execução
/proc/meminfo	Quantidade de memória do sistema e quantidade que está sendo usada
/proc/cpuinfo	Qual CPU está sendo utilizada
/proc/filesystems	Sistema de arquivos que o kernel suporta
/proc/kcore	Uma imagem da memória física
/proc/net	Status da rede do servidor
/proc/pci	Dispositivos PCI encontrados na inicialização
/proc/sys	número máximo de arquivos que podem ser abertos (file max); número de arquivos abertos (file nr); tempo que o sistema está ativo (uptime)

/root - Diretório do usuário root.

/sbin - Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.

/tmp - Diretório para armazenamento de arquivos temporários criados por programas.

/usr - Contém maior parte de seus programas. Normalmente acessível somente como leitura.

/var - Contém maior parte dos arquivos que são gravados com freqüência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

2.2 Diferenças entre o Debian Linux e Linux baseados em RedHat.

O Debian Linux é totalmente Free Software e não pertence a nenhuma empresa.

Distribuições baseadas em RedHat pertencem a algumas empresas, tais como Conectiva, RedHat, etc.

O modo de manipulação de pacotes, atualização e upgrade denominado **apt-get** foi idealizado pelo **Linux Debian** e está sendo usado por outras distribuições também.

Este recurso é fácil de ser utilizado e é bastante eficiente.

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-dpkg.html#sdpkg-apt)

- Linux Demo, Kurumin, Knoppix e outros baseados no Debian Linux.

Dentre as diversas distribuições Linux encontradas, algumas merecem destaque, tais como as versões "demo", ou seja, versões utilizadas para demonstação do Linux. Estas distribuições estão baseadas no Linux Debian.

São CDs "bootáveis" que podem reconhecer diversos tipos de Hardware, placas de rede, vídeo, etc.

Ajudam na recuperação de partições, perda de senhas, cópias de partições, demonstrações, treinamento,

etc.Trazem diversos softwares para Desktop e servidores, tais como OpenOffice, GIMP, KDE, etc.

(http://www.knoppix.org)

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/)

2.3 Esquema de partições do Linux Debian

Todos os S.O. Unix trabalham com discos particionados, onde cada partição é utilizada para armazenardados de usuário ou do Sistema Operacional.

Na instalação do S.O. as partições devem ser efetuadas para atender as necessidades dos Servidor (ou Desktop) a ser montado.

As partições no Linux podem apresentar a seguinte organização (usar o comando df):

Filesystem	**1K-blocks**	**Used**	**Available**	**Use %**	**Mounted on**
/dev/hda1	241070	76846	151776	34 %	/
/dev/hda3	241070	14	228606	1 %	**/tmp**
/dev/hda5	3937036	539124	3197908	15 %	**/var**
/dev/hda6	2948892	1143988	1655100	41 %	**/usr**
/dev/hda7	4640740	1746236	2658752	40 %	**/home**
/dev/md0	35007200	31388856	1840068	95 %	**/raid**
/dev/hdc1	39389776	36900250	488632	99 %	**/pub1**

Onde:

Filesystem : O nome da partição

1k-blocks : Cada bloco do Hard Disk tem 1 kbytes

Used : Quantidade de espaço em disco em uso

Available : Quantidade de espaço em disco disponível

Use% : Quantidade de espaço em disco disponível (em %)

Mounted on : Ponto de montagem da partição

/dev/hdaN : são partições do primeiro Hard Disk, da primeira controladora IDE.

As partições IDE obedecem a seguinte sequência:

Controladora	**Master**	**Slave**
IDE 0	hdaN	hdbN
IDE 1	hdcN	hddN
IDE 2	hdeN	hdfN
IDE 3	hdgN	hdhN

Onde **N** é o número máximo de partições e vale: **N=1,2,3,...,20.**

O número máximo é de oito **Hard Disks IDE (hda, hdb, hdc, hdd, hde, hdf, hdg e hdh)**.

Veja a listagem das possíveis partições no diretório **/dev**. Use o comando **ls -la /dev/hd***

Partições **SCSI** são indicadas por **/dev/sda, /dev/sdb**, etc, e podem ter, no máximo, 15 partições.

Use o comando **ls -la /dev/sda*** . O número máximo é de oito **Hard Disks SCSI (sda, sdb, sdc, sdd, sde, sdf, sdg e sdh)**, por controladora SCSI. Pode-se também usar partições do tipo RAID (Redundant Array of Inexpensive Disk), ou seja, é um arranjo redundante de discos baratos. Arranjos com RAID são usados para obter maiores velocidades de acesso aos discos, redundância de discos, correção de erros, espelhamento de discos, etc.

A partição /dev/md0 é uma partição em RAID 0 e é formada por duas partições SCSI (sda1 e sdb1).

O arquivo **/etc/raidtab** mostra como é formada a partição /dev/md0:

raid dev	**/dev/md0**
raid-level	0
nr-raid-disks	2
persistent-superblock	0
chunk-size	4
device	/dev/sda1
raid-disk	0
device	/dev/sdb1
raid-disk	1

Esta partição (/dev/md0) é a soma das duas partições (/dev/sda1 e /dev/sdb1) e o seu tamanho é a soma destas duas partições SCSI.

Comandos para particionamento de discos : **fdisk, cfdisk e sfdisk**.

Todos podem ser utilizados para manipular as partições dos discos (**IDE e SCSI**) .

Veja **man fdisk, man cfdisk, man sfdisk**

Por razões óbvias somente o usuário root pode fazer uso destas ferramentas.

Os sistemas de arquivo do Linux são : **ext2, ext3 ou swap**.

O arquivo **/etc/fstab** mostra os tipos de sistemas de arquivos que está sendo usado na instalação.

file system	**mount point**	**type**	**options**	**dump**	**pass**
/dev/hda1	/	ext2	defaults,errors=remount-ro	0	1
/dev/hda2	none	swap	sw	0	0
proc	/proc	proc	defaults	0	0
/dev/fd0	/floppy	auto	defaults,user,noauto	0	0
#/dev/sr0	/cdrom	iso9660	defaults,ro,user,noauto	0	0
/dev/hda3	/tmp	ext2	rw	0	2
/dev/hda5	/var	ext2	rw	0	2
/dev/hda6	/usr	ext2	rw	0	2
/dev/hda7	/home	ext2	rw	0	2
/dev/md0	/raid	ext2	rw	0	2
/dev/hdc1	/pub1	ext2	rw	0	2
/dev/sdc1	/Linorg.C	ext3	rw	0	2
/dev/sdd1	/Linorg.D	ext3	rw	0	2

Onde:

file system : Partição que se deseja montar

mount point : Diretório do GNU/Linux onde a partição montada será acessada.

type: Tipo de sistema de arquivos usado na partição que será montada. Para partições GNU/Linux use

ext2 (ou ext3), para partições DOS (sem nomes extensos de arquivos) use msdos, para partições Win 95

(com suporte a nomes extensos de arquivos) use vfat, para unidades de CD-ROM use iso9660

options: Especifica as opções usadas com o sistema de arquivos:

defaults - Utiliza valores padrões de montagem.

noauto - Não monta os sistemas de arquivos durante a inicialização (útil para CD-ROMS e disquetes).

ro - Monta como somente leitura.

rw - Monta para escrita e leitura.

user - Permite que usuários montem o sistema de arquivos (não recomendado por motivos de segu-

rança).

sync é recomendado para uso com discos removíveis (disquetes, zip drives, etc) para que os dados

sejam gravados imediatamente na unidade

Os campos **(dump e pass)** são **flags** para os comandos **dump** e **fsck**, respectivamente, e indicam as prioridades para que aquele sistema de arquivos seja verificado periodicamente com esses comandos. O valor 1 é o prioritário, seguido por 2, 3, e assim por diante. Um valor 0 indica que os comandos **dump** e **fsck** não devem fazer verificação periódica nesses volumes.

pass - Define a ordem que os sistemas de arquivos serão verificados na inicialização do sistema. Se usar 0, o sistema de arquivos não é verificado. O sistema de arquivos raíz que deverá ser verificado primeiro é o raíz **"/" .**

Os tipos de partições podem ser:

minix : Sistema de arquivos que suporta nome de arquivos de 14 ou 30 caracteres.

ext : Sistema de arquivos com aquivos com nomes grandes e grande número de **inodes**. Substituido pelo **ext2** e não é usado mais usado.

ext2 : Sistema de arquivos com aquivos com nomes grandes e grande número de **inodes** e muitas outras características.

xiafs : Sistema de arquivos com aquivos com nomes grandes e grande número de **inodes** e muitas outras características.

xfs : Sistema de arquivos com **journaling**, escalabilidade e outras características.

msdos : Sistema de arquivos MS-DOS.

hpfs : Sistema de arquivos HPFS.

iso9660 : Sistema de arquivos para CD-ROMs

nfs : Sistema de arquivos para montagem de sistemas remotos

swap : Partição de disco usada para SWAP.

Mais detalhes podem ser obtidos nos seguintes manuais:

man fs, fstab, mount, umount, fsck, e2fsck, etc.

Inodes:

Cada arquivo é representado por um inode, muitas vezes chamado de "serial number" do arquivo. O inode contem informacoes sobre: Tipo de arquivo (ordinario, diretorio, FIFO, dispositivo de bloco, etc).

Dono do arquivo (UID) Tamanho (em Bytes) Data de criação, acesso e modificação Grupo do arquivo (GID) Permissões Mapeamento do seu conteudo (data sectors).

Os layouts dos **inodes** variam de acordo com o S.O.

Listagem dos arquivo e de seus **inodes**.

No comando **ls** o parâmetro **(-i)** mostra o **inode** number.

Cada arquivo encontrado no S.O. Unix tem um **inode**. Este número nunca se repete.

$> ls -i /etc/*.conf

952 /etc/apmd.conf

953 /etc/auth.conf

975 /etc/pam.conf

958 /etc/dhclient.conf

O **inode** de cada arquivo da lista acima é a coluna mais à esquerda (números 952, 953, etc.)

Comando **stat** : Mostra o conteúdo do inode dos arquivos (permissões, tamanhos, links, access times, etc.)

$> stat /bin/ls

File: "/bin/ls"

Size: 300136 Allocated Blocks: 640 Filetype: Regular File

Mode: (0555/-r-xr-xr-x) Uid: (0/root) Gid: (0/wheel)

Device: 116,131072 Inode: 124 Links: 1

Access: Sun Feb 23 12:47:21 2003

Modify: Sun Feb 23 12:47:21 2003

Change: Sun Feb 23 12:47:21 2003
$> stat /etc/dm.conf
File: "/etc/dm.conf"
Size: 478 Allocated Blocks: 4 Filetype: Regular File
Mode: (0644/-rw-rr) Uid: (0/root) Gid: (0/wheel)
Device: 116,131072 Inode: 959 Links: 1
Access: Tue Jun 24 06:01:13 2003
Modify: Wed Oct 9 12:46:52 2002
Change: Fri Feb 21 16:23:41 2003
Manuais e mais detalhes: **man inode e man stat**

Numa instalação do Linux Debian o esquema de partições recomendado pode ser encontrada no seguinte URL: (http://www.linorg.cirp.usp.br/Debian.refs/Install/ch-partitioning.pt.html)

2.4 Manipulação das partições

Uso dos comandos mount e umount

Para montar uma partição ext2:
$> mount -t ext2 /dev/hda3 /tmp
Para desmontar :
$> umount /dev/hda3
ou
$>umount /tmp
Para montar um diskette no formato MSDOS:
$> mount -t msdos /dev/fd0 /mnt/floppy
Para montar um CD-ROM:
$> mount -t iso9660 /dev/hdc2 /cdrom
Para se formatar um diskete em **ext3**:
$> mkfs.ext3 /dev/fd0
Não se deve retirar o **diskette** antes de desmontá-lo.
Procure sempre conhecer os drivers dos periféricos antes de utilizá-los.
Veja os diversos argumentos que podem ser usados nos comandos **mount e umount** :
mount -h, mount help ou info mount
man mount, man umount, info umount
O arquivo **/etc/mtab** mostra as partições que estão montadas.
A manipulação de partições envolve muitas tarefas, dependendo do S.O. utilizado.
Os tópicos são:
Criação de partições, usando os comandos **mkfs, mkfs.ext2, mkfs.ext, etc.**
Verificação das partições, usando os comandos f**sck, e2fsck, badblocks, dumpfs.**
Criação de partições **ext2, ext3, msdos, reiserfs, etc**, com os comandos **mke2fs, mkreiserfs, etc**.

O URL abaixo possui bom material sobre este assunto.
(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-disc.html)
Veja também os comandos utilizados:
fs(5), badblocks(8), fsck(8), mkdosfs(8), mke2fs(8), mkfs.ext2(8), mkfs.minix(8), mkfs.msdos(8),
mkfs.xfs(8), mkfs.xiafs(8), dumpfs(8)

2.5 Documentação do Linux Debian

Diversos comandos para visualização da documentação existente no **Unix**:

Manual de programas : o clássico **man pages**, obrigatório para todos usuários. **(/usr/man)**

info : páginas em Hypertexto, fácil de usar. Uma melhora do **man pages.**

/usr/doc/programa : Documentos em ASCII/HTML instalado junto com o programa.

HOW-TO : Tutoriais dos tópicos relacionados ao Linux, disponível on-line se instalado (geralmente em **/usr/doc**)

WWW : Documentação disponível na WEB. Inclui Howtos, manuais do Unix e manuais dos programas diversos.

Comando **apropos** : este comando procura a linha de descrição no **man pages**.

Os manuais do Unix são divididos em oito sessões:

(1) User commands
(2) System calls
(3) Subroutines
(4) Devices
(5) File formats
(6) Games
(7) Miscellaneous
(8) System Administration

Exemplos de utilização:

$> man 5 fs

significa que será mostrado o manual da sessão **file formats** do comando **fs**

$> man 1 ls

simplesmente mostra o manual da sessão **user commands** do comando **ls**

$> apropos printer (mostra os man pages relacioandos a printer)

banner(6) - print large banner on printer
grolbp(1) - groff driver for Canon CAPSL printers (LBP-4 and LBP- 8 series laser printers)
lp(4) - printer port Internet Protocol driver
lpc(8) - line printer control program lpd(8) - line printer spooler daemon
lprm(1) - remove jobs from the line printer spooling queue
lpt(4) - generic printer device driver
lptcontrol(8) - a utility for manipulating the lpt printer driver
lptest(1) - generate lineprinter ripple pattern
mcprint(3) - ship binary data to printer
pac(8) - printer/plotter accounting information
printcap(5) - printer capability data base
psroff(1) - send troff to PostScript printer
ulpt(4) - USB printer support

Descobrindo o que é um comando:

$> whatis tac tac

(1) - concatenate and print files in reverse

Algo também muito útil e interessante é a capacidade (na versão GNU) destes comandos de manipular **wildcards**. Assim:

$> apropos -w 'l?'

irá mostrar todos os comandos de 2 letras comecados com **"l"**.

$> apropos -w '??'

irá mostrar todos os comandos de 2 letras.

$> apropos -w 'l*'

irá mostrar todos os comandos que começam com **"l"**.

Use os comandos man, xman, info, apropos e whatis

Manuais on-line:

- http://www.linorg.cirp.usp.br/dwww/
- http://www.softlab.ntua.gr/cgi-bin/man-cgi
- http://www.gsp.com/support/man/
- http://www.freebsd.org/cgi/man.cgi

2.6 Manutenção do Sistema de arquivos

A checagem do sistema de arquivos permite verificar se toda a estrutura para armazenamento de arquivos, diretórios, permissões, conectividade e superfície do disco estão funcionando corretamente. Caso algum problema exista, ele poderá ser corrigido com o uso da ferramenta de checagem apropriada. As ferramentas de checagem de sistemas de arquivos costumam ter seu nome iniciado por **fsck** e terminados com o nome do sistema de arquivos que verifica, separados por um ponto:

fsck.ext2 - Verifica o sistema de arquivos **EXT2** . Pode também ser encontrado com o nome **e2fsck**.

fsck.ext3 - Verifica o sistema de arquivos **EXT3**

fsck.minix - Verifica o sistema de arquivos **Minix**.

fsck.msdos - Verifica o sistema de arquivos **MS-DOS**. Pode também ser encontrado com o nome dos **fsck**.

fsck.nfs - Verifição sistema dearquivos **NFS**.

Para verificar um sistema de arquivos é necessário que ele esteja desmontado, caso contrário poderá ocorrer danos em sua estrutura. Para verificar o sistema de arquivos raiz (que não pode ser desmontado enquanto o sistema estiver sendo executado) você precisará inicializar através de um disquete e executar o **fsck.ext2**.

Uma boa referência sobre este assunto :

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-manut.html)

fsck.ext2

Este utilitário permite verificar erros em sistemas de arquivos **EXT2 e EXT3 (Linux Native).**

fsck.ext2 [opções] [dispositivo]

Onde:

dispositivo É o local que contém o sistema de arquivos **EXT2/EXT3** que será verificado (partições, disquetes, arquivos).

opções

-c Faz o fsck.ext2 verificar se existem agrupamentos danificados na unidade de disco durante a checagem.

-d Debug - Mostra detalhes de processamento do fsck.ext2.

-f Força a checagem mesmo se o sistema de arquivos aparenta estar em bom estado. Por padrão, um sistema de arquivos que aparentar estar em bom estado não são verificados.

-F Grava os dados do cache no disco antes de iniciar.

-l [arquivo] Inclui os blocos listados no [arquivo] como blocos defeituosos no sistema de arquivos.

O formato deste arquivo é o mesmo gerado pelo programa badblocks.

-L [arquivo] Faz o mesmo que a opção -l, só que a lista de blocos defeituosos do dispositivo é completamente limpa e depois a lista do [arquivo] é adicionada.

-n Faz uma verificação de somente leitura no sistema de arquivos. Com esta opção é possível verificar o sistema de arquivos montado. Será assumido não para todas as perguntas e nenhuma modificação será feita no sistema de arquivos.

Caso a opção **-c** seja usada junto com **-n, -l ou -L,** o sistema de arquivos será verificado e permitirá somente a atualização dos setores danificados não alterando qualquer outra área.

-p Corrige automaticamente o sistema de arquivos sem perguntar. É recomendável fazer isto manualmente para entender o que aconteceu, em caso de problemas com o sistema de arquivos.

-v Ativa o modo verbose (mais mensagens são mostradas durante a execução do programa).

-y Assume sim para todas as questões.

Caso sejam encontrados arquivos problemáticos e estes não possam ser recuperados, o **fsck.ext2** perguntará se deseja salvá-los no diretório **lost+found**. Este diretório é encontrado em todas as partições **ext2**.

Não há risco de usar o **fsck.ext3** em uma partição **EXT2**.

Após sua execução é mostrado detalhes sobre o sistema de arquivos verificado como quantidade de blocos livres/ocupados e taxa de fragmentação.

Exemplos de utilização:

fsck.ext2 /dev/hda2
fsck.ext2 -f /dev/hda2
fsck.ext2 -vrf /dev/hda1
fsck.ext2 -p /dev/fd0
fsck.ext3 /dev/hda5

Exercícios:

1 - Formate um Diskette de 3.5" com **ext2** ou **ext3** e copie nele alguns arquivos. Passe para outro usuário verificar e alterar o conteúdo deste diskette. Como você pode montá-lo ? Desmontá-lo ?

2 - Use os comandos **mount** e **umount** para montar o **diskette** e examine o seu conteúdo. Verifique se há problemas com o **diskette**.

3 - Use os comandos do pacote **mtools** para comparação com os comandos do Unix descritos nestasessão.

4 - Como deve ser feita a verificação de erros numa partição ? Verifique se há erros nas partições **/boot, /usr e /tmp.**

5 - Para que serve o comando **badblocks** ? Como deve ser utilizado ?

6 - Monte o CD-ROM e copie alguns arquivos para o **diskette**.

a - Use os comandos do pacotes do **mtools** e os comandos do Unix.

b - Compare os comandos do **mtools** com os comandos encontrados no Unix.

c - Utilize a ferramenta **mc (Midnight Commander)** e verifique as facilidades disponibilizadas.

d - Como deve ser utilizado esta ferramenta para ser fazer um **UNDELETE ?**

7 - Como você faria para recuperar arquivos apagados ? Veja a ferramenta **recover**, do Debian Linux.

a - Instale o pacote **recover** e tente utilizar o comando **debugfs.**

b - O site : **http://www.r-tt.com/** mostra ferramentas para esta finalidade.

c - Instale uma para o Debian e faça os testes.

8 - Para que serve o diretório **lost+found** ? Como ele é utilizado ?

9 - Explique os comandos para uso geral e administração do Unix. Mostre algumas diferenças, prvilégios, etc, entre eles.

10 - O que é encontrado no diretório **/proc** ? Comente o conteúdo de alguns subdiretórios.

a - Explique os conteúdos dos arquivos **/dev/null , /dev/zero , /proc/kcore.**

b - O quê representam os arquivos e diretórios **numéricos** encontrados no **/proc** ?

c - O quê representam os arquivos e diretórios **não numéricos** encontrados no **/proc** ?

d - Dê exemplos de como utilizar os controles de serviços de rede nos arquivos encontrados no **/proc.**

11 - Suponha que você tem um **Servidor Unix/Linux.**

Num certo dia o **S.O. Unix** não inicializa e você não consegue utilizar o servidor. Ninguém consegue utilizá-lo.

Houve prolemas com o suprimento de energia elétrica e o **No-break** não suportou a carga por muito tempo e não havia um esquema de **shutdown** apropriado.

Enfim, seu servidor sofreu uma pane geral !!!

a - Quais seriam os procedimentos para recuperar os dados do servidor ?

b - Você tem que recuperar os arquivos de senhas dos usuários, dados de cada usuário, arquivos de log, programas instalados e configurações que foram efetuadas durante a existência deste servidor.

c - Que ferramentas e recursos (Software e Hardware) você utilizaria ?

d - Elabore um procedimento para recuperar tudo que for necessário.

Capítulo 3

Manipulação de pacotes

O **apt** é sistema de gerenciamento de pacotes de programas que possui resolução automática de dependências entre pacotes, método fácil de instalação de pacotes, facilidade de operação, permite atualizar facilmente sua distribuição, etc. Ele funciona através de linha de comando e ainda não existe nenhuma interface amigável para uso deste programa, mesmo assim sua operação é muito fácil.

(http://linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-dpkg.html#s dpkg-apt)

Além do **apt-get** há também os programas **dpkg** e o **dselect** específicos para o gerenciamento de pacotes. A ferramenta **alien** faz a conversão de pacotes **RPM** (utilizados nos Linux baseados em RedHat) para padrões **tgz, deb, cpio, etc.**

O **dpkg** (**Debian-package**) é o programa responsável pelo gerenciamento de pacotes em sistemas Debian. Sua operação é feita em modo texto e funciona através de comandos, assim caso deseje uma ferramenta mais amigável para a seleção e instalação de pacotes, prefira o **dselect** (que é um **front-end** para o **dpkg**) ou o **apt**.

No URL abaixo há uma boa documentação sobre o **dpkg**:

http://linorg.cirp.usp.br/Debian.refs/Foca.Linux/Intermediario/index.html/ch-dpkg.html

O **dselect** é usado para uma configuração avançada do sistema Debian. Permite que sejam consultadas as dependências e não é muito amigável para instalação de pacotes. Os URLs abaixo mostram um bom manual sobre a utilização do **dselec**t:

http://www.debian.org/doc/manuals/dselect-beginner/index.pt.html

http://linorg.cirp.usp.br/Debian.refs/dselect-begginers/index.en.html

http://linorg.cirp.usp.br/Debian.refs/dselect-begginers.pt/index.pt.html

3.1 Configuração do apt-get

O programa para a configuração do **apt** é o **apt-setup**. Nele você vai definir o repositório fonte dos pacotes e a versão do S.O. Linux Debian.

Estas fontes podem ser **mirrors** do Debian ou o CD-ROM. O arquivo que será configurado é o **/etc/apt/sources.list.**

Cada linha contém a indicação do URL, distribuição e componente:

deb cdrom:/cdrom

deb cdrom:/mnt/cdrom

deb file:/mnt

deb file:/debian

deb file:/cdrom woody main contrib

deb http://www.us.debian.org/debian woody main contrib non-free

deb http://non-us.debian.org/debian-non-US woody non-US

deb ftp://ftp.debian.org/debian

deb ftp://nonus.debian.org/debian-non-US

Feita a configuração do arquivo o próximo passo é fazer as instalações do pacotes usando o apt-get.

3.2 Instalação de pacotes via apt-get

Inicialmente é necessário se fazer uma atualização das informações dos pacotes disponíveis, para isso use o comando:

$> apt-get update

Depois faça um **upgrade** dos pacotes instalados, usando o seguinte comando:

$> apt-get upgrade

Todos os pacotes que necessitarem de Upgrade serão atualizados. Este processo pode levar tempo. Depende das conexões de rede.

O manual do comando descreve bem a sua utilização.

Use também o comando **apt-get help** para ver como os parâmetros devem ser utilizados:

$> apt-get help

apt 0.5.4 for linux i386 compiled on Aug 19 2001 01:02:26

Usage: apt-get [options] command

apt-get [options] install | remove pkg1 [pkg2 ...]

apt-get [options] source pkg1 [pkg2 ...]

apt-get is a simple command line interface for downloading and installing packages.

The most frequently used commands are update and install.

Commands:

update - Retrieve new lists of packages

upgrade - Perform an upgrade install - Install new packages (pkg is libc6 not libc6.deb)

remove - Remove packages

source - Download source archives

build-dep - Configure build-dependencies for source packages

dist-upgrade - Distribution upgrade, see apt-get(8)

dselect-upgrade - Follow dselect selections

clean - Erase downloaded archive files

autoclean - Erase old downloaded archive files

check - Verify that there are no broken dependencies

Options:

-h This help text.

-q Loggable output - no progress indicator

-qq No output except for errors

-d Download only - do NOT install or unpack archives

-s No-act. Perform ordering simulation

-y Assume Yes to all queries and do not prompt

-f Attempt to continue if the integrity check fails

-m Attempt to continue if archives are unlocatable

-u Show a list of upgraded packages as well

-b Build the source package after fetching it

-c=? Read this configuration file

-o=? Set an arbitary configuration option, eg -o dir::cache=/tmp

See the apt-get(8), sources.list(5) and apt.conf(5) manual pages for more information and options.

This APT has Super Cow Powers.

$> apt-get -d install pacote

Faz apenas o download do **pacote. Não instala.**

3.3 Remoção de pacotes

A remoção de pacotes é simples de se fazer, porém tem que tomar cuidados com as dependências de outros pacotes.

Não se pode remover pacotes que quebrem as dependências, isso gera muitos problemas e pode ser necessário reconfigurar todo o sistema instalado.

$> apt-get remove pacote

Inicialmente tem que se saber qual pacote deverá ser removido.

O comando a seguir mostra como saber quais pacotes estão instalados:

$> dpkg -l "a*"

O resultado será a lista de pacotes que têm nome que começam com a letra **a**, por exemplo: **awk, aspell**, etc. Escolha o pacote e remova-o.

Cuidado com a quebra das dependências !!!

Quando se remove um pacote, os seus arquivos de configuração não são removidos, o que pode ajudar na re-instalação do pacote. Se for necessário remover integralmente o pacote os seguintes comandos deverão ser utilizados:

$> dpkg remove pacote (ou apt-get remove pacote)

$> dpkg purge pacote

3.4 Configuração de pacotes

Geralmente a sequência é instalar pacotes e depois configurá-los. Para isso os comandos relacionados ao **dpkg** devem ser estudados e bem assimilados. Eles são fundamentais no Debian Linux. Eis a lista completa destes comandos:

dpkg, dpkg-deb, dpkg-genbuilddeps, dpkg-parsechangelog, dpkg-scansources, dpkg-statoverride,
dpkg-architecture, dpkg-depcheck, dpkg-genchanges, dpkg-preconfigure, dpkg-shlibdeps,
dpkg-www, dpkg-buildpackage, dpkg-distaddfile, dpkg-gencontrol, dpkg-reconfigure,
dpkg-source, dpkg-www-browser, dpkg-checkbuilddeps, dpkg-divert, dpkg-name, dpkg-scanpackages,
dpkg-split, dpkg-www-installer

3.5 Utilização do comando dpkg

A utilização do comando **dpkg -l** mostra muita informação importante sobre os pacotes instalados.

Mostra o status de cada pacote, ou seja, se está instalado corretamente, se tem erros, etc.

Use o comando **dpkg -l | more** e veja, com atenção, as primeiras linhas. Elas indicam o cabeçalho de status de cada pacote :

Observe bem à esquerda da listagem acima, as linhas que começam com uma barra vertical (|), apontando para os sinais de mais (+).

Estes são os status dos pacotes.

* A primeira linha mostra o status do pacote:

Unknown: indica que o status é desconhecido.

Install: indica que o paote está marcado para instalação.

Remove: indica que o pacote está marcado para remoção.

Purge: indica que o pacote e seus arquivos de configuração estão marcados para remoção.

* A segunda linha mostra o status da instalação do pacote:

Not: pacote não instalado.

Install: pacote instalado com sucesso.

Config-files: indica que apenas os arquivos de configuração do pacote estão instalados.

Unpacked: indica que o pacote está pronto para ser instalado.

Failed-config: indica que o pacote está instalado mas seu script de configuração falhou.

Half-installed: indica que a instalação do pacote falhou
* A terceira linha mostra o status de erro do pacote:
None: não há erros associado ao pacote.
Hold: indica que o pacote está guardado, seguro, mas ele não pode ser instalado e nem removido.
Reinst-required: o pacote deve ser reinstalado.
* A quarta linha mostra o nome, a versão e uma breve descrição do pacote.
Se você quer ver a lista de pacotes instalados :
$> dpkg -l 'nome do pacote' | grep ^i | more
Se você quer ver todos os pacotes instalados:
$> dpkg -l '*' | grep ^i | more
Se você quer ver os pacotes removidos que deixaram os arquivos de configuração:
$> dpkg -l | grep "^rc"
Se pretende remover os pacotes e seus arquivos deconfiguração, use o parâmetro **purge**:
$> dpkg -purge pacote

3.6 Verificação dos pacotes

Mostra informações dos pacotes instalados, arquivos componentes, etc.
$> dpkg status pacote
$> dpkg status apache
$> dpkg listfiles pacote
$> dpkg -L pacote
dpkg help para maiores detalhes do comando
Veja também os outros comandos relacionados ao dpkg, tais como os citados no item 5.4.
Use o **dpkg-www** para procurar pacotes e informações sobre os pacotes.
A quantidade de comandos para a manipulação de pacotes é grande e um bom manual ajuda bastante.
Veja os seus manuais, argumentos e modos de utilização para poder fazer uma boa configuração e manutenção dos pacotes instalados.
(http://linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-dpkg.html)

Bons livros sobre Linux podem ser encontrados nos seguintes URL:
http://www.oreilly.com/openbook/
http://www.oreilly.com/catalog/debian/chapter/book/index.html
(http://www.linorg.cirp.usp.br/livros.html)

3.7 Consulta de pacotes

Para consultar a lista de pacotes disponíveis no Debian para instalação:
$> apt-cache search 'pacote'
Procura por pacotes com o nome **'pacote´**
$> apt-cache show 'pacote'
Mostra a versão dos pacotes procurados
Mais informações sobre os pacotes podem ser obtidas no site:
(http://packages.debian.org)
$> dpkg -s arquivo
Diz a qual pacote pertence determinado arquivo
$> dpkg -L pacote

Mostra os arquivos que pertencem a determinado **pacote.**

Download dos pacotes : o arquivo **.deb** são colocados em **/var/cache/apt/archives/**
$> **apt-get clean** faz uma limpeza neste diretório.

3.8 Reconfiguração de pacotes

Permite que os pacotes instalados sejam reconfigurados. Refazer as configurações do pacote específico.

dpkg-reconfigure pacote
Exemplos :
dpkg-reconfigure xserver-xfree86 : vai reconfigurar o Ambiente Gráfico XFree86
dpkg-reconfigure xserver-svga : vai reconfigurar o Ambiente Gráfico SVGA
dpkg-reconfigure etherconf : vai reconfigurar os parâmetros da placa de rede
dpkg-reconfigure ntpdate : vai reconfigurar o servidor de horário.

Exercícios:

1 - Use os comandos **apt-get e dpkg** para as seguintes tarefas:

a - instalar um pacote

b - remover um pacote

c - fazer apenas o download do pacote

d - instalar este pacote usando o **dpkg**

e - consultar as informações sobre o pacote

f - ver a lista de arquivos componentes do pacote

g - examinar as dependências do pacote

h - configurar ou reconfigurar o pacote

i - examinar o status do pacote

j - listar pacotes relacionados a XFree86

2 - Faça o upgrade de todos os pacotes instalados.

3 - Configure o **apt** de modo a obter pacotes de outras fontes, de outros mirrors Debian.

4 - Faça o **download** de um determinado pacote **.deb** e instale-o. Para isso, consulte o site http://packages.debian.org

5 - Instalar o pacote **synaptic** e fazer uso dele.

6 - Faça uma comparação entre os gerenciadores : **dselect, dpkg e apt**.

Capítulo 4

Gerenciamento de usuários

4.1 Introdução

Gerenciar usuários é uma tarefa que todo Administrador de Sistemas deve tomar cuidado.

Cadastrar usuários é simples. O mais difícil é fazer com que os usuários respeitem os limites, obedeçam as regras e cuide de seus dados.

O Administrador deve estabelecer estas regras e fazer o controle da utilização dos recursos.

Há muitos comandos para manipular as contas:

adduser, addgroup, passwd, vipw, newgrp, userdel, groupdel, lastlog, last, sg, chfn, chsh, id, logname, users e groups.

O detalhe de cada comando pode ser visto no seguinte URL:

(http://linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-cmdc.html)

4.2 Criação de grupos, contas de usuários, shells, senhas

Na criação de usuários todo o cuidado é pouco. Dependendo da finalidade do servidor, muitos usuários não precisam nem ter shells estabelecidos, só usarão o serviço de E-Mail, enquanto que outros vão fazer programas em uma linguagem de programação específica para o seu trabalho e precisarão ter acesso a diversas bibliotecas do sistema, comandos Unix, acesso a fitas DAT, Diskettes, Serviço de FTP, etc.

Em relação aos shells (interpretador de comandos do Unix), pode-se usar uma grande variedade, tais como:

sh, csh, ksh, tcsh, bash, esh, zsh, sash

Os shells : nologin, false são usados para negar login ao servidor e para retornar um valor falso quando o usuário tenta fazer um login.

Pode-se também deixar usuários sem shells, o que não é aconselhável.

Personalizar shells também é uma opção onde o usuário fica restrito a utilizar apenas alguns serviços do servidor, tais como:

Ler e-mails, trocar a senha e sair da sessão de login.

Eis um exemplo de shell script personalizado:

-

```
#!/bin/sh
clear
while true do
clear
echo ""
echo ""
echo " ============================================================="
echo ""
echo " Centro de Informática de Ribeirão Preto" echo " Servidor de Mail - xyzw.cirp.usp.br"
```

```
echo ""
echo "================================================================="
echo ""
echo " 1) Executar o PINE "
echo ""
echo " 2) Executar o comando passwd (alterar senha)"
echo ""
echo " 3) Finalizar telnet"
echo ""
echo " Digite a opcao : "
read opc case "$opc" in
1) pine;;
2) passwd || echo "*** Ocorreu erro na digitação da senha ***" && sleep 3;;
3) exit;; *) sleep 3; echo "\a *** Opção inválida***";;
esac done
```

-

Os cuidados com as contas de usuários são muitos, eis alguns:

Contas com as seguintes características:

- **mesmo UID ou GID do root (UID=GID=0)**
- **sem shell**
- **sem diretório $HOME**
- **sem senha**
- **sem arquivo /etc/shadow**
- **contas repetidas**
- **sem identificação do usuário**
- **sem grupo**
- **contas em LOCK**
- **contas com o mesmo ID e GID do root**
- **etc.**

4.3 Controle de senhas

Todos os usuários devem ter uma senha. O Administrador troca sua senha e a de todos os usuários.
Senhas de grupo podem ser utilizadas, apesar de ser um pouco raro.
Uma política de senhas deve ser estabelecida:
- periodicidade (troca e mensagens)
- bloqueio de contas
- eliminação de contas inativas
- verificação de quebras de senhas com softwares apropriados, tais como o Crack 5
- senhas fracas
- backup dos arquivos de usuários e senhas.

Exercícios:

1 - Como usuário **root**, criar um grupo de usuários denominado curso e cadastrar nele 5 usuários:

João da Silva Júnior, Pedro da Costa, Maria das Dores, Ermengalda Cremência e Rolando Lero.

a - Estabelecer o **login** e senha para cada um deles.

b - Estabelecer diretórios **$HOME** diferentes.

c - Colocar **shells** personalizado para apenas dois deles. Instalar este **shell** e fazer os testes.

d - Estabelecer uma política de troca de senhas.

e - Verificar as ocorrências dos logins destes usuários e salvar em arquivos.

f - Estabelecer um administrador para este grupo de usuários.

2 - Explicar porque não se pode apenas editar o arquivo de usuários e senhas para se alterar os dados dos usuários.

3 - Para quê serve o comando **vipw** ? E o comando **vigr** ?

4 - Usando o Software WEBMIN (www.webmin.com), faça o controle dos usuários de seu servidor.

5 - O que pode acontecer se um usuário possuir o mesmo **UID** ou **GID** do usuário **root** ?

6 - Para quê servem os comandos pwconv e **pwunconv** ?

7 - Qual a função do arquivo **/etc/shadow** ?

8 - Instale o programa **Crack** e veja se as senhas estabelecidas estão satisfatórias.

9 - Se você não tem a senha do **root** de um servidor, como você pode recuperá-la ou estabelecer outra senha para este usuário ?

10 - Como deve ser a conversão de senhas de um Sistema Operacional Windows NT para um Sistema Operacional baseado em Unix/Linux ?

Explique o que deve ser necessário e quais os recursos exigidos.

11 - Examine os dois **shell scripts** encontrados seguinte URL abaixo e explique as suas funcionalidades:

ftp://linorg.cirp.usp.br/FreeBSD.Utils/

Capítulo 5

O diretório /etc

Este diretório traz muitos arquivos de configuração de muitos serviços. É praticamente padrão em muitos sistemas Unix.

Facilita na administração dos sistemas Unix. Nele são encontrados todos os arquivos necessários para administração.

Dependendo do Unix utilizado, os arquivos de configuração dos serviços podem estar localizados em diferentes locais, podem ser colocados de acordo com as exigências dos serviços ou de acordo com as preferências do Administrador de Sistemas. No FreeBSD há muitos arquivos localizados no **/usr/local/etc** que fazem parte de **pacotes** extras instalados.

O número de arquivos encontrados no **/etc** varia de acordo com a distribuição Unix utilizada.

Não há uma padronização em relação a nomes dos arquivos encontrados no **/etc** .

No Linux Debian, os arquivos encontrados no **/etc** são:

Diretório /etc/alternatives

Este diretório contém links para diversos aplicativos padrões utilizados pelo sistema. Dentre eles são encontrados **links** para o **editor** do sistema e o **xterm** padrão usado pelo sistema.

Por exemplo, se você quiser usar o **editor jed** ao invés do **ae** ou **vi**, remova o **link editor** com o comando **rm editor**, localize o arquivo executável do **jed** com **which jed** e crie um **link** para ele **ln -s /usr/bin/jed editor**. De agora em diante o editor padrão usado pela maioria dos aplicativos será o **jed**.

Arquivo /etc/default/devpts

Este arquivo contém algumas configurações para os pseudo terminais em **/dev/pts**.

Arquivo /etc/default/rcs

Contém variáveis padrões que alteram o comportamento de inicialização dos scripts em **/etc/rcS.d**

Por exemplo, se quiser menos mensagens na inicialização do sistema, ajuste o valor da variável **VERBOSE** para **no**.

OBS: Somente modifique aquilo que tem certeza do que está fazendo, um valor modificado incorretamente poderá causar falhas na segurança de sua rede ou no sistemas de arquivos do disco.

Arquivo /etc/kbd/config

Este arquivo contém configurações padrões do pacote **kbd** para as fontes de tela e mapas de teclado usados pelo sistema. A fonte de tela é especificada neste arquivo (as fontes disponíveis no sistema estão localizadas em **/usr/share/consolefonts**).

O arquivo do mapa de teclados pode ser copiado para o diretório **/etc/kbd** com o nome **default.kmap** para que seja utilizado na inicialização do sistema ou escolhido interativamente através do utilitário **kbdconfig**.

Diretório /etc/menu-methods

Este diretório contém uma lista de arquivos que são executados pelo programa update-menu para criar os menus dos programas.

Arquivo /etc/menu-methods/menu-translate

Este arquivo permite fazer a tradução de nomes de menus, identificação ou títulos usados no ambiente gráfico.

Arquivo /etc/networks

Este local contém as configurações das interfaces (placas) de rede do sistema e outras opções úteis para a configuração/segurança da rede.

Arquivo /etc/networks/interfaces

Este é o arquivo de configuração usado pelos programas **ifup** e **ifdown**, respectivamente para ativar e desativas as interfaces de rede.

O que estes utilitários fazem na realidade é carregar os utilitários **ifconfig** e **route** através dos argumentos passados do arquivo **/etc/networks/interfaces**, permitindo que o usuário iniciante configure uma interface de rede com mais facilidade.

Abaixo um exemplo do arquivo **interfaces** é o seguinte:

iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0

As interfaces e roteamentos são configurados na ordem que aparecem neste arquivo. Cada configuração de interface inicia com a palavra chave **iface**. A próxima palavra é o nome da interface que deseja configurar (da mesma forma que é utilizada pelos comandos **ifconfig e route**). Você pode também usar **IP aliases** especificando **eth0:0** mas tenha certeza que a interface real (**eth0**) é inicializada antes.

A próxima palavra especifica a família de endereços da interface; Escolha **inet** para a rede **TCP/IP, ipx** para interfaces **IPX e IPv6** para interfaces configuradas com o protocolo **IPV6.**

A palavra **static** especifica o método que a interface será configurada, neste caso é uma interface com endereço estático (fixo).

Outros métodos e seus parâmetros são especificados abaixo (traduzido da página do arquivo interfaces):

O método **loopback** é usado para configurar a **interface loopback** (**lo**) **IPv4**.

O método static

É usado para configurar um endereço **IPv4** fixo para a interface. As opções que podem ser usadas com o métodos static são as seguintes (opções marcadas com * no final são requeridas na configuração):

address endereço * Endereço IP da Interface de rede (por exemplo, 192.168.1.1).
netmask máscara * Máscara de rede da Interface de rede (por exemplo, 255.255.255.0).
broadcast endereço Endereço de Broadcast da interface (por exemplo, 192.168.1.255).

network endereço Endereço da rede (por exemplo, 192.168.0.0).
gateway endereço Endereço do gateway padrão (por exemplo, 192.168.1.10).

O **gateway** é o endereço do computador responsável por conectar o seu computador a outra rede. Use somente se for necessário em sua rede.

O método dhcp

Este método é usado para obter os parâmetros de configuração através de um servidor DHCP da rede através das ferramentas:

dhclient, pump (somente Kernels 2.2.x) ou dpcpcp (somente kernels 2.0.x e 2.2.x)
hostname nome
Nome da estação de trabalho que será requisitado. **(pump, dhcpcd)**
leasehours leasttime
Lease time preferida em horas **(pump)**
leasetime leasetime
Lease time preferida em segundos **(dhcpcd)**
vendor vendedor
Identificador do vendedor **(dhcpcd)**
cliente identificação
Identificação do cliente **(dhcpcd)**

O método bootp

Este método pode ser usado para obter um endereço via **bootp**:

bootfile arquivo

Diz ao servidor para utilizar arquivo como arquivo de inicialização.

server endereço

Especifica o endereço do servidor **bootp**.

hwaddr endereço

Usa endereço como endereço de **hardware** no lugar do endereço original.

Algumas opções se aplicam a todas as interfaces e são as seguintes:

noauto

Não configura automaticamente a interface quando o **ifup** ou **ifdown** são executados com a opção **-a** (normalmente usada durante a inicialização ou desligamento do sistema).

pre-up comando

Executa o comando antes da inicialização da interface.

up comando

Executa o comando após a interface ser iniciada.

down comando

Executa o comando antes de desativar a interface.

pre-down comando

Executa o comando após desativar a interface.

Os comandos que são executados através das opções **up, pre-up e down** podem aparecer várias vezes na mesma interface, eles são executados na seqüência que aparecem. Note que se um dos comandos falharem, nenhum dos outros será executado. Você pode ter certeza que os próximos comandos serão executados adicionando || **true** ao final da linha de comando.

Arquivo /etc/networks/options

Este arquivo contém opções que serão aplicadas as interfaces de rede durante a inicialização do sistema. Este arquivo é lido pelo **script** de inicialização **/etc/init.d/network** que verifica os valores e aplica as modificações apropriadas no **kernel**.

Arquivo /etc/networks/spoof-protect

Permite especificar os endereços IPs locais e interfaces de rede que serão protegidas contra a técnica de IP spoofing (falsificação de endereço IP).

Diretório /etc/pam.d

Este diretório possui arquivos de configuração de diversos módulos **PAM** existentes em seu sistema.

Diretório /etc/ppp

Contém arquivos de configuração usados pelo **daemon pppd** para fazer uma conexão com uma rede **PPP** externa, criados manualmente ou através do **pppconfig**.

Diretório /etc/security

Este diretório contém arquivos para controle de segurança e limites que serão aplicados aos usuários do sistema. O funcionamento de muitos dos arquivos deste diretório depende de modificações nos arquivos em **/etc/pam.d** para habilitar as funções de controle, acesso e restrições.
Arquivo /etc/security/access.conf

É lido no momento do login do usuário e permite definir quem terá acesso ao sistema e de onde tem permissão de acessar sua conta.

O formato deste arquivo são 3 campos separados por :, cada linha contendo uma regra de acesso.

O primeiro campo deve conter o caracter + ou - para definir se aquela regra permitirá (+) ou bloqueará(-) o acesso do usuário.

O segundo campo deve conter uma lista de **logins, grupos, usuário@computador ou a palavra ALL (confere com tudo) e EXCEPT (excessão)**. O terceiro campo deve conter uma lista de terminais **tty** (para logins locais), nomes de computadores, nomes de domínios (iniciando com um .), endereço IP de computadores ou endereço IP de redes (finalizando com .).

Também pode ser usada a palavra **ALL, LOCAL e EXCEPT** (atinge somente máquinas locais conhecidas pelo sistema).

Abaixo um exemplo do **access.conf**
Somente permite o root entrar em **tty1**
-:ALL EXCEPT root:tty1
bloqueia o logins do console a todos exceto whell, shutdown e sync.
-:ALL EXCEPT wheel shutdown sync:console
Bloqueia logins remotos de contas privilegiadas (grupo wheel).#
-:wheel:ALL EXCEPT LOCAL .win.tue.nl
Algumas contas não tem permissão de acessar o sistema de nenhum lugar:#
-:wsbscaro wsbsecr wsbspac wsbsym wscosor wstaiwde:ALL
Todas as outras contas que não se encaixam nas regras acima, podem acessar de
qualquer lugar
Arquivo /etc/security/limits.conf

Defini limites de uso dos recursos do sistema para cada usuário ou grupos de usuários. Os recursos são descritos em linhas da seguinte forma: #<**dominio**> <**tipo**> <**item**> <**valor**>

O domínio pode ser um nome de usuário, um grupo (especificado como **@grupo**) ou o curinga ***.

O tipo pode ser **soft** para o limite mínimos e **hard** para o limite máximo.

O campo item pode ser um dos seguintes:
*** core - limita o tamanho do arquivo core (KB)**
*** data - tamanho máximo de dados (KB)**
*** fsize - Tamanho máximo de arquivo (KB)**
*** memlock - Espaço máximo de endereços bloqueados na memória (KB)**
*** nofile - Número máximo de arquivos abertos**
*** rss - Tamanho máximo dos programas residentes (KB)**
*** stack - Tamanho máximo de pilha (KB)**
*** cpu - Tempo máximo usado na CPU (MIN)**
*** nproc - Número máximo de processos**
*** as - Limite de espaço de endereços**
*** maxlogins - Número máximo de logins deste usuário**
*** priority - Prioridade que os programas deste usuário serão executados**

Abaixo um exemplo de arquivo **/etc/security/limits.conf**:

```
#<dominio>    <tipo>    <item>    <valor>
*    soft    core    0
*    hard    rss    10000
@student    hard    nproc    20
@faculty    soft    nproc    20
@faculty    hard    nproc    50
ftp    hard    nproc    0
@student    -    maxlogins    4
```

Arquivo /etc/crontab

Arquivo que contém a programação de programas que serão executados em horários/datas programadas.

Arquivo /etc/fstab

Contém detalhes para a montagem dos sistemas de arquivos do sistema.

Arquivo /etc/group

Lista de grupos existentes no sistema.

Arquivo /etc/gshadow

Senhas ocultas dos grupos existentes no sistema (somente o usuário root pode ter acesso a elas).

Use o utilitário **shadowconfig** para ativar/desativar o suporte a senhas ocultas.

Arquivo /etc/host.conf

É o local onde é possível configurar alguns ítens que gerenciam o código do resolvedor de nomes. O formato deste arquivo é descrito em detalhes na página de manual **resolv+**. Em quase todas as situações, o exemplo seguinte funcionará:

order hosts,bind

multi on

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

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

nospoof on

spoofalert on

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

Arquivo /etc/hostname

Arquivo lido pelo utilitário hostname para definir o nome de sua estação de trabalho.

Arquivo /etc/hosts

Banco de dados **DNS** estático que mapeia o nome ao endereço IP da estação de trabalho (ou vice versa).

Arquivo /etc/hosts.allow

Controle de acesso do **wrapper TCPD** que permite o acesso de determinadas de determinados endereços/grupos aos serviços da rede.

Arquivo /etc/hosts.deny

Controle de acesso do **wrapper TCPD** que bloqueia o acesso de determinados endereços/grupos aos serviços da rede.

Este arquivo é somente lido caso o **/etc/hosts.allow** não tenha permitido acesso aos serviços que contém.

Um valor padrão razoavelmente seguro que pode ser usado neste arquivo que serve para a maioria dos usuários domésticos é:

ALL: ALL caso o acesso ao serviço não tenha sido bloqueado no **hosts.deny**, o acesso ao serviço é permitido.

Arquivo /etc/hosts.equiv

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

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

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

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

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

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

Arquivo /etc/inetd.conf

http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Intermediario/index.html/ch-rede.html#s-rede-servicos-inetd-c

*** Este arquivo será descrito no capítulo 6**

Arquivo /etc/inittab

Este é o arquivo de configuração utilizado pelo programa **init** para a inicialização do sistema. Para mais detalhes sobre o formato deste arquivo, consulte a página de manual do **inittab.**

Arquivo /etc/inputrc

Este arquivo contém parâmetros para a configuração do teclado.

Arquivo /etc/isapnp.conf

Gerado pelo utilitário pnpdump e utilizado pelo utilitário isapnp para configurar os recursos de hardware dos dispositivos Plug-and-Play no GNU/Linux.

Arquivo /etc/isapnp.gone

Contém uma lista de endereços reservados que não deverão ser usados pelo isapnp.

Arquivo /etc/issue

Contém um texto ou mensagem que será mostrada antes do login do sistema.

Arquivo /etc/issue.net

Mesma utilidade do **/etc/issue** mas é mostrado antes do **login** de uma seção **telnet**. Outra diferença é que este arquivo aceita os seguintes tipos de variáveis:

* %t - Mostra o terminal tty atual.
* %h - Mostra o nome de domínio completamente qualificado (FQDN).
* %D - Mostra o nome do domínio NIS.
* %d - Mostra a data e hora atual.
* %s - Mostra o nome do Sistema Operacional.
* %m - Mostra o tipo de hardware do computador.
* %r - Mostra a revisão do Sistema Operacional.
* %v - Mostra a versão do Sistema Operacional.
* %% - Mostra um simples sinal de porcentagem (%).

Arquivo /etc/lilo.conf

Arquivo de configuração do gerenciador de partida **lilo.**

Arquivo /etc/login.defs

Definições de configuração para o pacote **login.**

Arquivo /etc/modules

A função deste arquivo é carregar módulos especificados na inicialização do sistema e mantê-los carregado todo o tempo. É útil para módulos de placas de rede que precisam ser carregados antes da configuração de rede feita pela distribuição e não podem ser removidos quando a placa de rede estiver sem uso (isto retiraria seu computador da rede).

Seu conteúdo é uma lista de módulos (um por linha) que serão carregados na inicialização do sistema. Os módulos carregados pelo arquivo **/etc/modules** pode ser listados usando o comando **lsmod.**

Se o parâmetro auto estiver especificado como um módulo, o **kmod** será ativado e carregará os módulos somente em demanda, caso seja especificado noauto o programa **kmod** será desativado. O **kmod** é ativado por padrão nos níveis de execução 2 ao 5.

Ele pode ser editado em qualquer editor de textos comum ou modificado automaticamente através do utilitário **modconf**.

Arquivo /etc/motd

Mostra um texto ou mensagem após o usuário se logar com sucesso no sistema. Também é usado pelo **telnet, ftp**, e outros servidores que requerem autenticação do usuário (nome e senha).

Arquivo /etc/mtab

Lista os sistemas de arquivos montados atualmente no sistema. Sua função é idêntica ao **/proc/mounts**.

Arquivo /etc/networks

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

Nome_da_Rede Endereço_da_Rede

Abaixo um exemplo de como se parece este arquivo:

loopnet 127.0.0.0
localnet 192.168.1.0
amprnet 44.0.0.0

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

Arquivo /etc/passwd

É o arquivo mais cobiçado por **Hackers** porque contém os dados pessoais do usuário como o **login, uid, telefone e senha** (caso seu sistema esteja usando senhas ocultas, a senha terá um ***** no lugar e as senhas reais estarão armazenadas no arquivo **/etc/shadow**).

Arquivo /etc/printcap

Banco de dados de configuração da impressora, usado por daemons de impressão como o **lpr e lprng.**

Arquivo /etc/protocols

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

nomeprotocolo número apelidos

Arquivo /etc/resolv.conf

É o arquivo de configuração principal do código do resolvedor de nomes. Seu formato é um arquivo texto simples com um parâmetro por linha e o endereço de servidores **DNS** externos são especificados nele. Existem três palavras chaves normalmente usadas que são:

domain

Especifica o nome do domínio local.

search

Especifica uma lista de nomes de domínio alternativos ao procurar por um computador, separados por espaços. A linha **search** pode conter no máximo 6 domínios ou 256 caracteres.

nameserver

Especifica o endereço IP de um servidor de nomes de domínio para resolução de nomes. Pode ser usado várias vezes.

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

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

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

Arquivo /etc/serial.conf

Configurações das portas seriais do sistema. Veja a página de manual do serial.conf e a página de manual do utilitário setserial para detalhes de como configurar adequadamente a taxa de transmissão serial conforme seu dispositivo.

Arquivo /etc/services

É um banco de dados simples que associa um nome amigável a humanos a uma porta de serviço amigável a máquinas.

É um arquivo texto de formato muito simples, cada linha representa um item no banco de dados.

Cada item é dividido em três campos separados por qualquer número de espaços em branco (tab ou espaços). Os campos são:

nome porta/protocolo apelido # comentário

name

Uma palavra simples que representa o nome do serviço sendo descrito.

porta/protocolo

Este campo é dividido em dois sub-campos.

*** porta** - Um número que especifica o número da porta em que o serviço estará disponível. Muitos dos serviços comuns tem designados um número de serviço. Estes estão descritos no RFC-1340.

*** protocolo** - Este sub-campo pode ser ajustado para **tcp** ou **udp**.

É importante notar que o item **18/tcp** é muito diferente do item **18/udp** e que não existe razão técnica porque o mesmo serviço precisa existir em ambos.

Normalmente o senso comum prevalece e que somente se um serviço esta disponível em ambos os protocolos **tcp** e **udp**, você precisará especificar ambos.

apelidos

Outros nomes podem ser usados para se referir a entrada deste serviço. comentário Qualquer texto aparecendo em uma linha após um caracter "#" é ignorado e tratado como comentário.

Arquivo /etc/shadow

Este arquivo armazena as senhas criptografadas caso estiver usando o recurso de senhas ocultas. Este arquivo somente pode ser lido pelo usuário **root**.

Arquivo /etc/shells

Contém uma lista de interpretadores de comando **(shells)** válidos no sistema.

Arquivo /etc/syslog.conf

Contém configurações para definir o que será registrado nos arquivos de **log** em **/var/log** do sistema. Veja a página de manual **syslog.conf** e dos programas **klog** e **syslogd** para entender o formato usado neste arquivo.

Arquivo /etc/timezone

Contém a sua localização para cálculo correto do seu fuso-horário local.

http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Intermediario/index.html/ch-etc.html

Capítulo 6

Processos e Daemons

Um daemon é um processo que executa em segundo plano e que realiza uma função específica ou uma tarefa relacionada ao sistema. Os daemons são programas, em vez de serem parte do kernel. Muitos daemons são ativados na inicialização e continuam a ser executados, enquanto o sistema estiver ativo e funcionando. Outros daemons são iniciados quando necessários e são executados só quando forem úteis.

O termo daemon foi utilizado pela primera vez como um termo de computador por Mick Bailey, um britânico que estava trabalhando com a equipe de programação da CTSS no MIT durante o início da década de 60. Mick citou o dicionário de inglês Oxford como suporte tanto ao significado como à ortografia da palavra. As palavras daemon e demon vêm da mesma raiz, mas daemon é uma forma mais antiga e seu significado é um pouco diferente. Um daemon é um espírito assistente que influencia o caráter ou a personalidade de uma pessoa. Os daemons não são os lacaios do mal ou do bem; são criaturas de pensamento e vontade independentes. Seguiram seu caminho do CTSS para Multics e para Unix, onde são tão populares que precisam de um superdaemon (inetd) para gerenciá-los.

O inetd é um daemon responsável por iniciar outros daemons à medida que são necessários. Todas as versões de Unix o incluem, e a maioria dos daemons mais recentes é executada sob seu controle.

Há muitos daemons com que os administradores de sistemas devem estar intimamente familiarizados, quer por exigir um bocado de administração ou porque desempenham um papel importante na operação rotineira do sistema.

6.1 Daemon init

init é o primeiro processo a rodar depois que o sistema inicializa e, de várias maneiras, é o daemon mais importante. Ele sempre tem um PID de 1 e é um antepassado de todos os processos de usuário e da maioria dos processos de sistema.

Na inicialização, init coloca o sistema no modo de usuário único (single user) ou gera um shell para ler os arquivos de inicialização do sistema. Quando você inicializa o sistema no modo de usuário único, init lê os arquivos de inicialização depois que você termina o shell de usuário único, normalmente digitando exit ou <control-D>.

Depois de processar os arquivos de inicialização, init consulta um arquivo de onfiguração (**/etc/inittab** na maioria dos sistemas, **/etc/ttys** para o **FreeBSD**) para determinar quais portas físicas ele deve esperar que os usuários efetuem login. Ele abre essas portas e gera um processo getty em cada uma. Se uma porta não puder ser aberta, init emite periodicamente mensagens de erro para o console do sistema até que a

porta se torne capaz de abrir ou seja removida da lista de portas ativas.

A maioria das sessões de login é estabelecida na rede por daemons como rlogind, telnetd e sshd.

Além dos seus deveres de gerenciamento de terminais, init também tem a difícil tarefa de exorcizar processos zumbis que, do contrário, ficariam acumulados no sistema.

Você pode desligar o sistema enviando um sinal ao **init** - normalmente **SIGTERM** - que faz com que ele leve o sistema a atuar no modo monousuário. Esse é o último passo na maioria dos scripts de desligamento.

init é tão essencial à operação do sistema que o sistema automaticamente reinicializará se **init** sair de operação alguma vez.

O **init** define vários **níveis de execução** que determinam qual conjunto de recursos do sistema deve ser ativado.

Dependendo do Sistema Operacional, há 6 níveis, ou mais. As características de cada nível de execução são definidas no arquivo **/etc/inittab**.

Normalmente init recebe seu nível de execução inicial como um argumento a partir da rotina de carga do sistema (**boot loader**). Se 's' for especificado (ou **linux single**), init entra no modo monousuário. Caso contrário, ele procura em **/etc/inittab** por entradas que se aplicam ao nível de execução solicitado e executa seus comandos correspondentes.

O comando **telinit** altera o nível de execução do init uma vez que o sistema está ligado. Por exemplo, **telinit 5** força init a ir ao nível de execução 5. O argumento mais útil do **telinit** é **-q**, que faz com que ele releia o **/etc/inittab**.

Muitos sistemas implementam uma camada adicional de abstração acima do mecanismo básico de níveis de execução fornecido por **init**. Esses sistemas mantêm scripts de inicialização no diretório **/etc/init.d**, a partir de onde eles são relacionados a diretórios específicos ao nível de execução denominados **/etc/rcX.d**.

Trazer **init** para um novo nível de execução faz com que os scripts apropriados sejam executados com os argumentos **start** e **stop**. Esse recurso permite que a inicialização e o desligamento sejam tratados de uma maneira metódica.

6.2 Classificação dos daemons

No Linux Debian os scripts de inicialização do sistema estão localizados no diretório **/etc/init.d**. Nos **Linux** baseados em RedHat estes mesmos scripts estão localizados no diretório **/etc/rc.d/init.d**. Cada Unix traz um diretório apropriado para estes arquivos.

Estes serviços (ou **daemons**) são classificados em:

6.2.1 Daemons de Sistema:

Algumas tarefas do sistema, como gerenciar memória virtual e sincronizar a cache de disco, são gerenciadas por daemons, em vez de pelo próprio kernel. Os daemons que realizam estas funções não podem ser manipulados pelo administrador de sistemas e geralmente devem ser deixados sozinhos.

Exemplos de daemons :

kswapd: paginação da memória virtual

Daemons de NFS:

Fazem o compartilhamento de NFS (**Network File Systems**)

nfs-common: serviço de NFS

nfs-user-server: script de startup do NFS

nfsd:executado em servidores de arquivo e trata de solicitações de clientes NFS. Em alguns sistemas, esse daemon é chamado de **rpc.nfsd**

mountd:responde a solicitação de montagem (para NFS)

amd e automount: montagem de sistemas de arquivos sob demanda. São automontadores de NFS, os daemons que esperam até que um processo tente utilizar um sistema de arquivos antes que realmente o monte.

lockd e statd: gerenciam bloqueios de NFS.

6.2.2 Daemons de Internet:

São os daemons que utilizam os protocolos de Internet para manipular solicitações.

talkd: serviço de bato-papo em rede

sendmail: Transporta correio eletrônico, servidor de e-mails.

snmpd: serviço remoto de gerenciamento de rede.

rwhod: mantém lista remota de usuários

ftpd: servidor de transferência de arquivos. **sftpd, proftpd, wu-ftpd**, etc.

popper: servidor básico de caixa de correio. **pop3d, cucipop**, etc

imapd: servidor de luxo de caixa de correio, Internet Mail Access Protocol, alternativa mais requintada do POP.

rlogind: servidor de login remoto

telnetd: servidor de login remoto

sshd: servidor de login remoto seguro

rshd: servidor de execução de comando remoto

rexecd: servidor de execução de comando remoto

routed: mantém tabelas de roteamento. Um roteador baseado em Unix.

named: servidor de DNS.

syslogd: processa mensagens de **log**.

fingerd: pesquisa usuários.

6.2.3 Daemons de sincronização de tempo

São protocolos que servem para sincronizar os servidores, em relação a **clock** e horários.

Clocks sincronizados são essenciais para correlacionar entradas de arquivo em **log** em caso de uma falha de segurança.

timed: sincroniza relógios. Sincroniza os relógios das máquinas servidoras Unix. Baseados em máquinas mestras e máquinas escravo.

xntpd: sincroniza relógios. Versão melhorada do **ntpd**. Utiliza o protocolo **Network Time Protocol** para sincronizar vários relógios.

6.2.4 Daemons de inicialização e de configuração

São daemons que inicializam quando o servidor é ligado.

bootpd: servidor de inicialização. Usado para atribuir um endereço IP para máquinas cliente na rede, geralmente clientes sem disco.

tftpd: servidor de transferência de arquivos trivial. Similar ao ftpd, porém mais simples. Permite que todos leiam seu diretório.

rarpd:Reverse Address Resolution Protocol. Praticamente abandonado. Permite que as máquinas sem disco determinem seu endereços IP na inicialização.

bootparamd: suporte vital avançado a clientes sem disco. Diz aos clientes sem disco onde localizar seus sistemas de arquivos.

dhcpd:Dynamic Host Configuration Protocol (DHCP) fornece endereços IP aos clientes conectados na rede.

6.2.5 Controle dos daemons

Os daemons encontrados no **/etc/init.d** são facilmente controlados através dos argumentos **start, stop, restart, reload e status**.

Por exemplo:

$> cd /etc/init.d

$> ./webmin stop

Isso faz com que o serviço **webmin** seja parado. Para inicializá-lo o comando a ser usado é:

$> ./webmin start

Dependendo do nível de execução que se encontra, consulte o arquivo **/etc/inittab**, alguns **daemons** serão inicializados durante o **boot** do unix.

Por exemplo, no nível 2, os daemons que serão inicializados são os que se localizam no diretório **/etc/rc2.d**.

Neste diretório há apenas links para os arquivos do diretório **/etc/init.d**.

Os links cujos nomes iniciam com a letra **S** são os **daemons** que inicializam no **boot** do sistema. Os que iniciam com a letra **K** são os que não inicializam no **boot** do sistema.

Para você desabilitar um serviço basta trocar o nome do **link**.

Por exemplo:

$> cd /etc/rc2.d

$> mv S99webmin K99webmin

Use o software Webmin para fazer estas mudanças. Selecione os serviços que deverão e os que não deverão inicializar no **boot**.

6.3 Utilização do Inetd

O **inetd** é um **daemon** que gerencia outros **daemons**. Ele ativa seus **daemons** clientes quando há trabalho para eles fazerem e permite que sejam eliminados de maneira suave quando suas tarefas estiverem concluídas. Só funciona com **daemons** que oferecem serviços para rede.

Muitos **daemons** podem ser utilizados de maneira tradicional (em que são iniciados uma vez e continuam a ser executados até o sistema ser desligado) ou com o **inetd**.

6.3.1 Configurando o inetd

O **inetd** consulta um arquivo de configuração (normalmente **/etc/inetd.conf**) para determinar quais portas de rede ele deve ouvir. O formato é o mesmo em todas as plataformas. Eis um exemplo:

serviço	socket	protocolo	flags	user	servidor	argumentos
telnet	stream	tcp	nowait	root	/usr/sbin/telnetd	telnetd
ftp	stream	tcp	nowait	root	/usr/sbin/ftpd	ftpd
finger	stream	tcp	nowait	guest	/usr/sbin/fingerd	fingerd
pop-3	stream	tcp	nowait	root	/usr/sbin/popper	popper

A primeira coluna contém o nome do serviço. inetd mapeia os nomes de serviços para números de portas consultando o arquivo **/etc/services** (para serviços de **TCP** e **UDP**).

A segunda coluna determina o tipo de socket que o serviço utilizará e será comumente **stream** ou **dgram** . Em geral, **stream** é utilizado com serviços TCP (orientado a conexão) e **dgram** é utilizado com **UDP**.

A terceira coluna identifica o protocolo de comunicação utilizado pelo serviço. Os tipos admissíveis estão listado no arquivo **/etc/protocols**. O protocolo é quase sempre **TCP** ou **UDP.**

Se o serviço que está sendo descrito puder processar várias solicitações de uma vez (em vez de processar uma solicitação e sair), a coluna quatro deve ser configurada como wait; essa opção impede que **inetd** ative constantemente novas cópias do **daemon**. Ele é utilizado com serviços que lidam com uma grande quantidade de pequenas solicitações. Se **wait** não for apropriado, coloque **nowait** aqui para fazer **inetd** ativar uma nova cópia do **daemon** toda vez que ele receber uma solicitação.

A quinta coluna fornece o nome de usuário sob o qual o daemon deve ser executado.

Se não confiar em um determinado programa ou souber que ele tem problemas de segurança, você pode executá-lo sob um usuário diferente de root para reduzir sua exposição. Naturalmente, essa técnica só funciona para daemons que não exigem poderes de **root**. No exemplo acima, **fingerd** é executado sob o usuário **guest**.

Os campos restantes fornecem o nome de caminho completo do daemon e seus argumentos de linha de comando. O primeiro argumento sempre deve ser o nome curto do programa. Este requisito não é uma peculiaridade de **inetd**, mas uma convenção tradicional do Unix que normalmente é ocultada pelo **shell**.

6.3.2 O arquivo /etc/services

Depois de adicionar um novo serviço ao inetd.conf, você também pode precisar criar uma entrada para ele no arquivo **/etc/services**. Esse arquivo normalmente está localizado em **/etc**. Ele é utilizado só para serviços **TCP/IP** bem conhecidos; informações semelhantes para os serviços de **RPC** são armazenadas em um arquivo de configuração separado, normalmente em **/etc/rpc**.

Eis algumas linhas selecionadas de um arquivo services (o original contém mais de 400 linhas):

```
tcpmux 1/tcp # TCP port service multiplexer
echo 7/tcp
echo 7/udp
discard 9/tcp sink null
discard 9/udp sink null
systat 11/tcp users
daytime 13/tcp
daytime 13/udp
netstat 15/tcp
qotd 17/tcp quote
...
chargen 19/tcp ttytst source
chargen 19/udp ttytst source
ftp-data 20/tcp
ftp 21/tcp
fsp 21/udp fspd
ssh 22/tcp # SSH Remote Login Protocol
ssh 22/udp # SSH Remote Login Protocol
telnet 23/tcp
# 24 - private
smtp 25/tcp mail
...
```

O formato da linha é :

nome porta/proto aliases #comentário

Os serviços geralmente são listados em ordem numérica, embora essa ordem não seja necessária.

nome é o nome do simbólico do serviço (o nome que você utiliza no arquivo **/etc/inetd.conf**). **porta** é o número da porta em que o serviço normalmente ouve; se o serviço for gerenciado por **inetd**, ele é a porta em que **inetd** ouvirá.

proto indica o protocolo utilizado pelo serviço; na prática, ele é sempre **tcp** ou **udp**.

Se um serviço pode utilizar **UDP** ou **TCP**, uma linha para cada um deles deve ser incluída (assim como o serviço time antes). O campo alias contém nomes adicionais para o serviço (por exemplo, **whois** também pode ser pesquisado como **nickname**).

6.3.3 Reiniciando o inetd

As configurações no inetd serão válidas só quando você reiniciar o processo **inetd**. A reinicialização deve ser feita enviando um sinal de **HANGUP** (**kill -HUP** ou **kill -1**) para o **inetd**. Depois de sinalizar, espere um minuto e, então, verifique se os arquivos em **log** para mensagens de erro relacionadas com suas alterações **(inetd** registra erros em **syslog** pelo recurso **daemon)**. Teste qualquer serviço novo que você tenha adicionado para se certificar de que eles funcionam corretamente.

6.4 Protegendo o inetd

Como **inetd** é responsável por gerenciar muitos serviços comuns baseados em rede, ele desempenha um papel importante na proteção do sistema. É importante verificar se somente os serviços de que você precisa e em que confia foram ativados em **inetd.conf**. Em um novo sistema, você quase que certamente precisará modificar **inetd.conf** para desativar serviços desnecessários ou indesejáveis no seu ambiente.

Uma boa regra prática é ativar somente os serviços de que você realmente precisa e desativar todo o resto.

Mesmo assim, é uma boa idéia suplementar **inetd** com pacote **TCP wrappers,** de Wietse Venema, que registra em **log** todas as tentativas de conexão e restringe acesso a **daemons**, dependendo de quem estiver tentando conectá-los.

Mais referências sobre **daemon, init, inetd e serviços de rede** podem ser encontradas no seguinte URL:

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Avancado/index.html/ch-rede.html#s-rede-servicos)

(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Intermediario/index.html/ch-rede.html#s-rede-servicos-inetd-c)

6.5 xinetd

"**xinetd**" é uma alternativa ao **"superservidor" inetd**.
Extended Internet Daemon.

Apresenta mais recursos que o **inetd. É um inetd** melhorado. Apresenta diversos recursos que não estão disponíveis no **inetd**.

O **xinetd** apresenta os seguintes recursos:

- Controle de acesso embutido, semelhante ao TCP Wrappers (discutido anteriormente) com base no endereço, nome ou domínio do host remoto.
- Controle de acesso baseado em segmentos de tempo.
- Registro em **log** completo das conexões, incluindo sucessos e flhas.
- Prevenção de ataque de DoS, limitando o número de servidores do mesmo tipo que podem executar ao mesmo tempo, o número total de servidores, o tamanho dos arquivos de **log** e o número de conexões que uma única máquina pode iniciar.
- Vínculo de um serviço a uma interface específica, por exemplo, apenas um endereço IP interno.

A sintaxe de configuração é muito diferente do **Inetd**. Cada serviço é salvo em seu próprio arquivo no diretório **/etc/xinetd.d**, ou em qualquer diretório especificado pela diretiva **includedir** no arquivo **xinetd.conf**. O arquivo normalmente tem o nome do próprio serviço, como **telnet, ftp, ou imap**. Por exemplo, o arquivo **/etc/xinetd.d.ftp** pode ser escrito como:

```
service ftp
{
  flags = REUSE NAMEINARGS
  socket-type = stream
  protocol = tcp
  wait = no
  user = root
  server = /usr/sbin/in.ftpd
  server_args = -l -a
```

```
}
```

Além disso, pode-se fornecer padrões para todos os serviços, incluindo uma seção de **defaults** em **/etc/xinetd.conf**. Esses se aplicarão a todos os serviços, a menos que seja modificado pelo arquivo específico do serviço:

```
defaults
{
  instances = 25
  log_type = FILE /var/log/servicelog
  log_on_success = HOST PID
  log_on_failure = HOST RECORD
  per_source = 5
}
```

onde:

instances: é o número máximo de solicitações que um servidor pode tratar ao mesmo tempo
log_type: Log para um arquivo específico (FILE nomearquivo) ou via **syslog**
log_on_success: Registrar em **log** várias informações no caso de uma conexão bem sucedida.
log_on_failure: Registrar em **log** várias informações no caso de uma conexão que falhou.
per_source: Número máximo de conexões que um endereço IP específico pode fazer para determinado serviço.

Exercícios:

1 - Examine os arquivos relacionados à configuração do **init** responda aos seguintes itens:

a - Quais são os níveis de execução do **init** ?

b - Como se utiliza cada um deles ?

c - Como mudar de um nível de execução para outro ?

d - Onde estão os daemons correspondentes a cada nível de execução ?

2 - Explique as diferenças entre os diversos níveis de execução do init.

3 - O que é necessário para controlar e restringir acesso aos diversos terminais disponíveis ?

4 - O que são e como se classificam os daemons ?

5 - Como se controla cada daemon ?

6 - O que é o inetd e para quê serve ?

7 - Crie um serviço de rede no inetd de modo a mostrar os últimos logins de cada usuário. O serviço deve ser acessível pela porta 12345.

8 - Explique porquê somente o usuário root tem controle sobre os daemons, init e inetd, enfim, sobre os serviços de inicialização e de Rede.

9 - Como deve ser o procedimento para a manutenção dos seguintes serviços administrativos:

a - Verificação e instalação de novos discos

b - Atualização dos serviços disponíveis (FTP, SENDMAIL, etc)

c - Troca de placas de rede, de hardware, etc.

d - Adição / Remoção de usuários do sistema

10 - Utilize a ferramenta nmap para fazer a verredura das portas TCP/UDP do servidor Unix.

11 - Crie um serviço (no **inetd.conf**) para monitorar o espaço em disco de seu Linux. Coloque ele para rodar na porta 12345.

11 - Configure o **xinetd**. Configure os serviços telnet e ftp e faça o controles e registros de logs necessários.

12 - Faça uma comparação entre o **inetd** e o **xinetd**.

Capítulo 7

TCP-Wrappers

tcpd, frequentemente chamado de pacote de 'TCP wrappers' ('empacotadores de TCP') permite registrar em log conexões a serviços TCP, como telnetd, ftpd e fingerd. Além disso, ele permite restingir os sistemas que podem se conectar a esses serviços. Esses dois recursos podem ser muito úteis quando você estiver monitorando ou controlando convidados indesejáveis. tcpd foi escrito por Wietse Venema e está disponível em ftp.porcupine.org.

Ele vem de forma-padrão com muitos Unix, tais como **Debian, RedHat e FreeBSD.**

tcpd é de fácil instalação e não requer modificações em programas de rede existentes.

Ele pode trabalhar junto com o **inetd**; você simplesmente modifica o seu arquivo **/etc/inetd.conf** para executar **tcpd** em vez do programa real do servidor de rede. **O tcpd** então realiza o registro em **log** necessário e as verificações de segurança apropriadas antes de por o servidor em execução. Pode-se também modificar os arquivos **/etc/hosts.deny** e **/etc/hosts.allow** para que os registros em outros arquivos de **log** sejam efetuados, de maneira personalizada.

Por exemplo, se seu **/etc/inetd.conf** originalmente contivesse a linha
telnet stream tcp nowait root /usr/sbin/in.telnetd in.telnetd

Você poderia alterar isso para **:**
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
O arquivo de log resultante (configurado em **/etc/syslog.conf**) seria semelhante a algo como:

```
Nov 12 08:52:43 chimchim in.telnetd[25880]: connect from tintin.colorado.edu
Nov 12 09:33:00 chimchim in.telnetd[23456]: connect from teste.usp.br
Nov 12 10:50:23 chimchim in.telnetd[26744]: connect from tik.dominio.com.br
Nov 12 22:02:53 chimchim in.telnetd[56350]: connect from link.suspeito.com.br
```

7.1 Arquivos de configuração

Os arquivos usados para controlar o acesso aos serviços controlados pelo TCP wrapper são encontrados no diretório /etc.

No FreeBSD há apenas o arquivo hosts.allow e todo o controle do tcpd é feito neste arquivo.

No Linux há dois arquivos, hosts.allow e hosts.deny. O controle é feito colocando-se limites e regras nos arquivos.

Veja detalhes no seguinte URL:
(http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Avancado/index.html/ch-rede.html#s-rede-seg-tcpd)

Exemplos de arquivos **/etc/hosts.allow** e **/etc/hosts.deny:**

```
# /etc/hosts.allow
```

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

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

7.2 Verificação da segurança do TCPD

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

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

Pode-se colocar alguns parâmetros extras nos arquivos **hosts.allow** e **hosts.deny**.

Por exemplo, para o usuário root receber um e-mail a cada consulta via serviço finger adicione a seguinte linha no arquivo **/etc/hosts.allow**:

in.fingerd: ALL: (finger @%h | mail -s "finger from %h" root)

O arquivo /etc/hosts.deny também pode ser modificado para gerar arquivos de **log** dedicado ao **TCP wrapper**.

Eis um exemplo:

```
ALL:ALL: twist /var/noaccess %h %d;/bin/echo -e "%h tentou acessar %d > %n" `date` >>
/var/log/logtcpd.log;
ALL:ALL except local
```

O programa **/var/noaccess**, será executado caso a conexão ao serviço **TCP** solicitado esteja bloqueada.

O arquivo de **log** gerado será o **/var/log/logtcpd.log**

Eis o programa fonte **noaccess.c**:

```
/*
NoAccess
Credits go to Piveti , a friend of mine
*/
#include <stdio.h>
int main(int argc, char *argv[]){
int i;
printf("Voce , %s ,nao foi autorizado a usar %s\n\n", argv[1],argv[2]);
printf("Acesso negado ...\n\n");
printf("...mais um para a estatistica...\n\n");
}
```

A compilação e instalação do programa **noaccess.c** deve ser feita da seguinte maneira:

$> gcc -o noaccess noaccess.c
$> cp noaccess /var/noaccess
$> chmod 755 /var/noaccess
$> touch /var/log/logtcpd.log

$> chmod 777 /var/log/logtcpd.log

Faça os testes habilitando apenas alguns serviços (no arquivo **/etc/hosts.allow**) e verificando o conteúdo do arquivo **/var/log/logtcpd.log**.

Proíba o acesso ao serviço **telnetd** a alguns **endereços IP** e tente fazer conexão no serviço **telnetd**.

Exercícios:

1 - Qual a função do **TCP wrapper** ?

2 - Como devem ser verificados os **logs** do **TCP wrapper** ? Onde eles estão localizados ?

3 - Quais os cuidados que devem ser tomados ao se utilizar o **TCP wrapper** ?

4 - Elabore um roteiro para configuração, testes e utilização do **TCP wrapper**.

5 - Instale o **talk**d e configure-o para atender os seguintes itens:

a - permitir que somente 6 endereços IP possam utilizá-lo.

b - fazer os **logs** das conexões em arquivo de **log** específico.

c - somente os IPs da rede local e mais dois IPs de outra rede podem utilizar este serviço, .

6 - Somente os serviços encontrados no **/etc/inetd.conf** podem ser controlados pelo **TCP wrapper** ? Explique.

7 - Se o seu **S.O. Unix** não tem o **TCP wrapper** instalado como você deve proceder para instalá-lo ?

8 - Qual a função do comando **tcpdump** ? É aconselhável sua utilização ?

9 - O que você usaria como referência para saber se há um **TCP wrapper** rodando num S.O. Unix ?

10 - Quais os itens que devem ser observados para que seja possível controlar serviços de rede utilizando o **TCP Wrapper** ?

11 - Faça uma comparação entre os seguintes trechos de um **TCP wrapper** instalado em dois servidores com Sistema Operacional Unix :

a - (trecho do arquivo **/etc/hosts.deny** de um **S.O Linux**):

```
ALL:ALL: /bin/mail -s "%s connection attempt from %c" root
```

b - (trecho do arquivo /etc/hosts.allow de um **S.O. FreeBSD**):

```
fingerd : ALL \ : spawn (echo Finger. | \
/usr/bin/mail -s "tcpd\: %u@%h[%a] fingered me!" root) & \ : deny
```

Capítulo 8

Controle de tarefas agendadas

A necessidade de fazer tarefas com horários pré-definidos é uma tarefa constante de todos os administradores de sistemas. Verificar conexões de rede, efetuar backups e outras tarefas podem ser facilitadas utilizando ferramentas tais como o daemon crond .

Para isso o Administrador Unix precisa fazer com que um script ou comando seja executado sem qualquer intervenção humana.

8.1 cron: comandos de escalonamento

No Unix, a execução periódica fica a cargo do daemon cron. O cron é iniciado na inicialização do sistema.

O cron lê os arquivos de configuração que contém listas de linhas de comando e o momento em que elas devem ser invocadas. As linhas de comando são executadas pelo interpretador shell sh. O que você pode fazer manualmente a partir do shell também pode ser feito via cron.

Um arquivo de configuração cron é chamado de crontab, forma abreviada para **cron table** - tabela cron.

Todos os arquivos crontab são armazenados em um único diretório do sistema, onde cron sabe procurá-los. O comando crontab transfere arquivos crontab para e a partir desse diretório.

Em geral, há (no máximo) um arquivo crontab por usuário: um para o root, um para o usuário **zemane**, e assim por diante.

Os arquivos crontab são nomeados com os nomes de logins dos usuários aos quais eles pertencem, e cron utiliza esses nomes de arquivos para descobrir qual UID utilizar ao executar os comandos que cada arquivo contém.

Os comandos a serem executados estão nos crontabs. Os crontabs são analisados e são executados no momento certo.

O daemon crond é atualizado automaticamente, não é necessário que se envie o comando kill -HUP (crontab PID).

Os arquivos de logs do cron estão localizados no diretório (/var/cron/log, /var/adm/cron/log ou /var/log), dependo do Unix que se está utilizando. Estes arquivos listam os comandos executados e o horário em que foram executados.

Em alguns sistemas, criar o arquivo de log permite o registro em log, e remover os arquivos de log desativa o registro em log. Em outros sistemas, o registro em log é ativado ou desativado por meio de arquivos de configuração.

Uma outra variação é fazer com que cron utilize syslog. O arquivo de log cresce rapidamente e raramente é útil; deixe o registro em log desativado, a menos que você esteja depurando um problema específico.

A maioria dos crons não compensa os comandos que não foram executados enquanto o sistema esteve desligado. Além disso, algumas versões de cron não entendem o horário de verão, fazendo com que os comandos sejam pulados ou executados duas vezes quando acontece a alteração de

horário. Se você utiliza cron para tarefas sensíveis ao horário (como contabilidade), tenha cuidado nessas situações.

É possível previnir problemas potenciais evitando os horários afetados pelas mudanças.

Os URLs abaixo mostra mais detalhes sobre o cron:

http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-manut.html#smanut-tarefas

http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-manut.html#smanut-cron

As tarefas são definidas no arquivo /etc/crontab e por arquivos individuais de usuários em /var/spool/cron/crontabs/[usuário]

(criados através do programa crontab).

Adicionalmente a distribuição Debian utiliza os arquivos no diretório /etc/cron.d como uma extensão para o /etc/crontab.

Para agendar uma nova tarefa, basta editar o arquivo /etc/crontab com qualquer editor de texto (como o **ae** e o **vi**) e definir o mês/dia/hora que a tarefa será executada. Não é necessário reiniciar o daemon do cron porque ele verifica seus arquivos a cada minuto.

8.2 Formato dos arquivos crontab

Todos os arquivos crontab em um sistema compartilham um formato comum. Os comenstários são introduzidos com um caracter # na primeira coluna de uma linha. Cada linha sem comentários contém seis campos e representam um comando:

minuto hora dia mês dia-da-semana comando

Os cinco primeiros campos são separados por um espaço em branco (ou <tab>), mas, dentro do campo comando, o espaço em branco é tomado literalmente.

minuto, hora, dia, mês e dia-da-semana fornecem ao comando as informações sobre os horários em que deve ser executado. Suas interpretações são mostradas na tabela a seguir:

Campo Descrição Intervalo

minuto : o minuto da hora 0 a 59

hora : a hora do dia 0 a 23

dia : o dia do mês 1 a 31

mês : o mês do ano 1 a 12

dia-da-semana : o dia da semana 0 a 6 (0=domingo)

Por exemplo, a especificação de horário:

45 10 * * 1-5

significa **10:45**, de segunda a sexta-feira.

Nunca coloque um asterisco no primeiro campo, a menos que você queira que o comando seja executado a cada minuto.

Nos campos **dia-da-semana** e **dia**, há uma grande ambiguidade com a qual se deve tomar cuidado.

Cada dia é tanto um dia da semana como um dia do mês. Se **dia-da-semana** e dia forem especificados, o dia precisa satisfazer apenas uma das duas condições a fim de ser selecionado. Por exemplo:

0,30 * 13 8 5

significa que a **cada meia hora na sexta-feira** e a **cada meia hora no 13** , 0 dia do mês, não a **cada meia hora da sexta-feira 13**.

comando é a linha de comandos do **Shell** a ser executada. Pode ser qualquer comando válido de **Shell** e não deve ser colocado entre aspas. Considera-se que comando continue até o final da linha e pode conter espaços em branco ou tabulações.

O **crontab** do superusuário pode executar como um usuário qualquer precedendo-os com **/bin/su nomedousuário -c**.

Eis alguns exemplos de comandos válidos para **crontab**:

30 2 * * 1 (cd /home/zemane/tarefas; gzip testes.*)

Essa entrada será ativada às 2:30 da manhã de cada segunda-feira. Ela executará o comando **gzip testes.*** no diretório **/home/zemane/tarefas**.

Normalmente, todas as saídas produzaidas por um comando cron são remetidas ao proprietário do **crontab.**

20 1 * * * find /tmp -atime +3 -exec rm -f {} ';'

Este comando será executado a 1:20 toda manhã. Ele remove todos os arquivos no diretório **/tmp** que não forem acessados em três dias.

55 23 * * 0-3,6 /opt/gambiarra/testa-script

Esta linha executa testa-script às 23:55, todos os dias, exceto quintas e sexta-feiras.

8.3 Gerenciamento de crontab

crontab nomedoarquivo instala **nomedoarquivo** com o o seu **crontab**, substituindo qualquer versão anterior.

crontab -e confere uma cópia de seu **crontab**, invoca o seu editor sobre ele e, então reapresenta-o no diretório de **crontab**.

crontab -l lista o conteúdo do seu crontab para a saída padrão e **crontab -r** o remove, deixando você absolutamente sem nenhum **crontab**.

A maioria dos sistemas permite que **root** forneça um argumento **nomedousuário**, e, desta forma, os argumentos dos **crontabs** de outros usuários podem ser visualisados ou editados.

Por exemplo: **crontab -l zemane** mostra os **crontabs** do usuário **zemane**.

crontab -r zemane vai remover os crontab do usuário **zemane**.

No **Linux e FreeBSD**, o argumento **-u** é usado para especificar o nome do usuário. **crontab -l -u zemane**, **crontab -r -u zemane.**

Veja **crontab -help** e **man crontab** para mais detalhes do **crontab.**

Por default, todos os usuários podem enviar arquivos crontab para **cron**.

Dois arquivos de configuração, normalmente chamados de **cron.allow** e **cron.deny**, permitem que você anule esta diretiva.

No **Linux Debian** esses arquivos estão no diretório **/etc**.

No **FreeBSD**, os arquivos encontram-se em **/var/cron** e são chamados simplesmente de **allow** e **deny**.

O **allow** tem a lista dos usuários que podem enviar **crontab**, um usuário por linha.

Um usuário não listado não pode enviar **crontabs**. Se o arquivo **allow** não existir, o arquivo **deny** é verificado.

Ele também é apenas uma lista de usuários, mas o significado é o inverso: o acesso é permitido a todo mundo, exceto aos usuários listados.

Se nem o arquivo **allow** nem o arquivo **deny** existirem, apenas o **root** pode enviar **crontabs.**

É importante observar que o controle de acesso é implementado por **crontab**, não por **cron**.

Se um usuário for capaz de introduzir sorrateiramente um arquivo **crontab** em um diretório apropriado por outros meios, **cron** executará sem restrições os comandos que este diretório contém.

8.4 Comandos at, atq e atrm

O **at** agenda tarefas de forma semelhante ao cron com uma interface que permite a utilização de linguagem natural nos agendamentos. Sua principal aplicação é no uso de tarefas que sejam disparadas somente uma vez.

Uma característica deste programa é a execução de aplicativos que tenham passado de seu horário de execução, muito útil se o computador é desligado com frequência ou quando ocorre uma interrupção no fornecimento de energia.

Para utilizar o at, instale-o com o comando:

$> apt-get install at

O próximo passo é criar os arquivos /etc/at.allow e /etc/at.deny.

Estes arquivos são organizados no formato de um usuário por linha. Durante o agendamento, é verificado primeiro o arquivo at.allow (lista de quem pode executar comandos) e depois o at.deny (lista de quem NÃO pode executar comandos). Caso eles não existam, o agendamento de comandos é permitido a todos os usuários.

Veja mais detalhes e exemplos de aplicação no seguinte URL:

http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Iniciante.com.Intermediario/index.html/ch-manut.html#smanut-at

Veja também os manuais relacionados a estes comandos:

man at, man atq, man atrm

Todas as tarefas agendadas são armazenadas em arquivos dentro do diretório **/var/spool/cron/atjobs**

O comando atq mostra os **jobs** pendentes de cada usuário.

O formato da saída é: **Job number, date, hour, job class.**

Exemplo:

$> atq

68 2003-07-17 12:00 a root

O comando **atrm** apaga os **jobs**, identificado pelo número.

Exemplo:

$> atrm 68

Exercícios:

1 - Quais as aplicações do crontab ?

2 - O que acontece com as tarefas agendadas se o servidor sofre uma parada total e reinicia duas horas depois ?

3 - Elabore crontabs que executem as seguintes funções:

a - limpar o diretório /tmp todos os dias às 23:00

b - verificar os usuários que usaram os serviços telnet e ssh durante todos os dias da semana e guardar estas informações num arquivo.

c - agendar o download de um certo arquivo para as 18:30 de toda quinta-feira, usando o comando **wget.**

4 - Use o comando at para executar instruções especificadas num determinado arquivo.

5 - Faça uma comparação entre os serviços **cron** e **at**.

6 - Um crontab executa tarefas em todos os 30 dias de cada mês.

Se o número de dias do mês alterar para 31 ou 29 dias, o que acontece com a execução destas tarefas ?

7 - Um certo arquivo de log cresce sem limite, constantemente. Todos os dias ele tem o tamanho aumentado em 5 Kbytes.

Elabore um crontab para atender os seguintes itens:

a - Comprimir o arquivo (com o comando gzip) e trocar o nome do arquivo, a cada dois dias, às 19:30.

b - Manter o arquivo zipado em algum diretório e criar o arquivo novo que vai crescer novamente.

c - Remover os arquivos zipados no final de cada mês.

d - Os arquivos zipados deverão ter o nome do arquivo original e a extensão deve ser a data de quando ele foi zipado.

Capítulo 9

Análise de logs

9.1 Introdução

A monitoração do trabalho de servidores de rede sempre é acompanhada analisando-se arquivos e relatórios gerados por determinados programas.

Estes programas podem ser daemons de serviços ou programas dedicados. Estes relatórios são comumente denominados **arquivos de log** que vão mostrar a situação de cada serviço, mensagens que ajudarão a entender o que se passa com o servidor, mensagens de erros, alertas, etc.

Através destes relatórios, o Administrador de Sistemas vai executar determinadas tarefas de modo a manter o servidor em condições de funcionamento ou tomar decisões corretivas.

As máquinas **UNIX** utilizam um dos sistemas de **logging** mais simples, porém útil.

Os programas possuem duas opções principais quando se trata de gerar arquivos de **log**:

Arquivos de log gerados pelo processo Alguns programas tratam do seu próprio **logging**. Isso significa que seus arquivos de **log** contêm a saída apenas dessa origem. Os arquivos de **log** normalmente são determinados por meio de argumentos da linha de comandos ou arquivos de configuração, ou então são definidos dentro do próprio programa. Por exemplo, o **Apache WEB Server** possui um arquivo de log de acesso contendo os **URLs** atendidos (normalmente chamado de **acces_log**) e um arquivo de **log** de erro relacionando os problemas (páginas faltando, respostas de CGI inválidas, etc) que ele experimenta (normalmente chamado de **error_log**).

Mensagens syslog O modo mais comum de os programas registrarem informações é por meio do daemon **syslogd**. Esse é um programa cuja única finalidade é permitir um método comum de **logging** para programas divergentes. **Syslog** determina o que fazer com os **logs**, dependendo de duas coisas: a **facilidade syslog** e o **nível de logging**.

Todos os softwares tem sua própria maneira de gravar os arquivos de **log.**

9.2 O que são logs ?

Registro histórico das atividades de um dado programa/sistema.

Em geral, programas não interativos (servidores executando em **background**, **kernel** dos Sistemas Operacionais): Um dos benefícios do **UNIX** (e seus derivados) é fornecer mecanismos padronizados de **logging** (mensagens de execução de programas e processos) das atividades dos numerosos **daemons** e programas que estão sendo executados no sistema.

Estes **logs** podem ser usados para verificar erros (**debug**) do sistema, monitorar a sua utilização, cobrindo tudo, desde possíveis falhas de segurança até mensagens de alerta (**warning**) de possíveis problemas de **hardware**.

Às vezes é a única maneira de saber o que eles estão fazendo.

Logs gerados voluntariamente:

- A geração de **logs** normalmente pode ser desligada.
- Gerar **logs** úteis é característica de um bom programa servidor.

Logs são normalmente armazenados e apresentados em modo texto e

Permite fácil consulta usando qualquer editor de textos.

Se apresentam frequentemente com jargões e abreviaturas específicas que tornam a leitura incompreensível para um não-iniciado.

Às vezes, porém, requerem visualizadores específicos (exemplo: o **Event Viewer** do **Windows**).

9.3 O que é "logado" ?

É importante verificar a diferença entre os tipos de **logs** encontrados num **S.O. UNIX**

Basicamente há dois tipos de logs: **logs do sistema** e **logs dos aplicativos**.

Todos os Sistemas Operacionais apresentam os **logs** de sistema.

Nos **S.O. Windows** há aplicativos dedicados para a geração e interpretação dos logs do sistema, enquanto que nos **S.Os. Unixes** os arquivos de **log** (na sua maioria) são armazenados em formato de texto simples.

Os **logs** de aplicativos são dependentes das aplicações que estão sendo executadas e como estas aplicações estão configuradas para gerar os **logs**.

Nos **logs de sistema** serão encontradas mensagens e alertas (**warnings**) do **Kernel** que incluem informações dos módulos carregados, dados do **Sendmail**, que permite visualizar o caminho das mensagens que são processadas no sistema e mensagens sobre tentativas de logins efetuados com sucesso ou conexões que falharam.

Os **logs** de sistema são gerados pelo **Daemon "syslogd"**, que é carregado na inicialização do **UNIX**. O **syslogd** acessa mensagens em oito níveis de cada processo do sistema, tais como:

Kernel, sistema de E-Mail, programas de usuários configurados a usar o syslogd e programas de autenticação (o login).

Os níveis de mensagens são (em ordem crescente de prioridade):

debug
info
notice
warnig
err
crit
alert
emerg

Estes níveis são usados no arquivo **/etc/syslog.conf**, que faz com que o **syslogd** crie **logs** para diferentes tipos de informação.

O arquivo **/etc/syslog.conf** contém várias entradas, uma por linha, cada uma contendo dois campos separados por um ou mais espaços:

o nível de log e a **localização do arquivo de log.**

A lista dos níveis de **log** é formada por níveis de log separados por ponto-e-vírgula.

Estes níveis de logs são indicados por nomes, tais como **"mail", "kern" (para o kernel), "user" (para os usuários) e "auth" (para os programas de autenticação**).

Os pares **níveis de log** incluem:

mail.err : mensagens de erros geradas pelos servidores de **E-Mail**

***.info** : todas mensagens de informações

kern.emerg : mensagens de emergência do **Kernel**

9.4 Para que servem os logs ?

Finalidades principais:

Depuração de problemas (**debugging/troubleshooting**):

Informais, assume-se que sejam confiáveis e que não tenham sido adulterados.

Cômputo de estatísticas de uso e/ou performance:

Exemplos: **logs de Webservers, de Servidores de E-Mail, de FTP, etc.**

Fornecer trilhas de auditoria:

Tendem a ser mais formais, preferivelmente com detecção de adulteração, visando análise forense. Por exemplo: **Auditoria de um sistema comprometido.**

Billing/cobrança:

Serviços cobrados por volume/quantidade por período de tempo.

9.5 Sincronização dos relógios de rede

Muitos serviços de rede se utilizam deste recurso para sincronizar seus relógios.

O serviço **NTP (Network Time Protocol)** executa esta função.

Sistemas de **log** sincronizados com relógios padrões ajudam na verificação precisa dos **logs** de um servidor.

Recomenda-se a utilização de mais de um servidor **NTP**.

O configurador deste recurso é o comando ntpdate. A vantagem é a precisão (da ordem de segundos). Rápido e fácil de instalar, mas requer um servidor **NTP** que não saia do ar.

NTPd instalado localmente provê uma precisão de milissegundos.

Mais trabalhoso de instalar, mas pode ser tornado robusto com o uso de vários servidores.

9.6 Protocolo Syslog

É o protocolo e respectivo servidor do Unix clássico para receber mensagens de **log** via rede.

Suporta especificar os vários subsistemas (**facilities**) que originam a mensagem e um nível de prioridade.

Despacha a mensagem para um arquivo, para um determinado usuário logado na máquina, ou para outro servidor **syslog** em outra máquina.

Este protocolo usa a porta **514/UDP** e, geralmente faz parte da instalação default dos **S.O. UNIX**.

Se for desativado, ou não instalado, não haverá registro das mensagens de **log**.

O número de **logs** gerado é grande e pode ser configurado. Diversas configurações de **syslog** podem ser efetuadas. É bastante comum encontrar **clientes syslog**, **servidor syslog**, **servidores syslog em rede, etc.**

Numa rede estes elementos podem estar presentes, conectados através de **cabos seriais, crossover, switches ou Hubs**.

Também pode ser implementado um sistema de criptografia para o tráfego de **syslog**.

Utilizando o **syslog-ng (Next generation logging daemon)** é possível criar um **tunnel SSL** entre o **host** emissor e o receptor.

O **syslog** padrão pode ser substituído pelo **syslog-ng**, que será descrito mais adiante.

9.7 Facilidades Syslog

Todas as mensagens **syslog** são marcadas com uma facilidade e nível específico. O arquivo **/etc/syslog.conf** permite especificar onde as mensagens entram. A facilidade **syslog** é simplesmente uma maneira de fazer com que um programa descreva em qual grupo de **logging** ele se encaixa.

As facilidades disponíveis são:

- **kern**: núcleo (**kernel**) do sistema operacional
- **user**: aplicação ou processo do usuários (**default**)
- **mail/news/UUCP**: subsistemas de correio eletrônico e notícias
- **cron**: executor de tarefas agendadas por horário
- **daemon**: servidores (**daemons**) do sistema
- **auth**: autorização, autenticação e controle de acesso
- **lpr**: subsistema de impressão
- **mark**: marcações de data/hora regulares
- **local0-local7**: para aplicações personalizadas
- **syslog**: mensagens internas geradas pelo próprio **syslog**
- **authpriv**: mensagens de autorização que não sejam do sistema em si
- **user**: mensagens genéricas no nível de usuário
- "*": todas as acima exceto **mark**

9.8 Nível de logging de syslog

Os programas apanham cada entrada de **log** com um nível de **logging**, de modo que o daemon **syslog** possa informá-lo ou ignorá-lo, dependendo da configuração. Os níveis de mensagens são:

- **emerg**: situações de emergência/pânico
- **alert**: situações urgentes
- **crit**: situações críticas
- **warning**: advertências
- **notice**: situações incomuns que inspiram investigação

- **info**: informativos do estado normal do sistema
- **debug**: dados detalhados/prolixos para depuração
- **err:** situações de erros

9.9 Ações do Syslog

O registro das mensagens de **log** são estabelecidos no arquivo **/etc/syslog.conf**. O formato de cada linha é:

facilidade.nivel_do_log destino_do_log

Os campos são separados por tabulações.
Exemplo: daemon.notice /var/log/daemon.log
grava todos os **logs** para programas que estão usando a facilidade **daemon** e sejam de prioridade **notice** ou maior no arquivo **/var/log/daemon.log**. Pode-se especificar um asterisco (*****) para uma facilidade ou nível de **log** combinar com qualquer facilidade ou nível de **log**, respectivamente.

- **nomedearquivo**: acrescenta a mensagem para o arquivo especificado na máquina local
- **@nomeouip**: Repassa as mensagens para o servidor syslog na máquina especificada
- **user1,user2,&**: escreve a mensagem no console dos usuários especificados que estiverem conectados (se o usuário não estiver conectado, nada é escrito)
- *****: escreve a mensagem para todos os usuários conectados.

Exemplo de syslogd.conf:

Registra todas as mensgens do kernel no console

kern.* /dev/console

Registra tudo (exceto correio) de nível info ou maior

Não registra mensagens de autenticação privativas

***.info;mail.none;authpriv.none;cron.none /var/log/messages**

O arquivo authpriv tem acesso restrito

authpriv.* /var/log/secure

Registra todas as mensagens de correio em um só lugar

mail.* /var/log/maillog

Registre as atitudes do **cron**

cron.* /var/log/cron

```
# Todos recebem as mensagens de emergência; além disso, copie-as para outra máquina
*.emerg                                          *,@172.31.11.50

# Erros de correio e serviço de notícias em um arquivo especial
uucp,news.crit                                   /var/log/spooler

# Salve as mensagens de inicialização em um arquivo
local7.*                                         /var/log/boot.log

# Envia mensagens de nível crítico para o centalizador, mantendo
# uma cópia no arquivo de mensagens e avisando ao root
*.crit                                           /var/log/messages,@172.31.11.50,root
```

9.9.1 Armazenamento dos Dados

Os arquivos de log (nos **S.Os. UNIX**) são textos simples, crescem muito e a busca é sequencial .

Sem uma política de rotacionamento/descarte, os arquivos tendem a crescer sem limite.

O rotacionamento de **logs** será visto mais adiante.

Os arquivos de **log** podem ser compactados e guardados e guardados em discos, **CD-ROMs, Fitas DAT, etc.**

Também podem ser utilizados Bancos de dados para se guardar os **logs**, mas estes costumam crescer mais rápido ainda.

Cuidados: Tamanho máximo fixo, conhecido desde o início: não cresce infinitamente

9.10 Arquitetura de coleta de Logs

Logging local em cada máquina Quando os **logs** são armazenados na própria máquina, sempre há o risco de usuários maliciosos que podem adulterar os **logs**.

Ferramentas para este fim são facilmente encontradas e são de fácil utilização.

Simples de utilizar e bastante perigosas. Adulteram os dados e comprometem a segurança.

Logging local + centralizado

- Se o **log** local for adulterado, ainda há a cópia no servidor de **log**.
- Espaço, tráfego, processamento, **backup** e resistência a ataques do servidor tornam-se questões mais complexas.

Para evitar estes tipos de problemas, aconselha-se que sejam utilizados **loghosts** bem dimensionados e bem protegidos.

Caracterísitcas de um **loghost**:

- servidor de **syslog**
- amplo espaço em disco
- política de rotacionamento

- **firewalling** próprio
- análise, sumarização e publicação

Desta forma, os arquivos de **log** podem ficar mais seguros, porém tenha sempre em mente que **nada é totalmente seguro**.

9.11 Centralizando logs

É bastante aconselhável a criação de um sistema de **loghost** que executa serviços muito limitados, apenas arquivando e processando dados de auditoria. A conexão com este sistema deve ser através de **SSH** ou outro protocolo com criptografia forte para acesso administrativo.

Configurações recomendadas:

- Dificulte o acesso à configuração do **syslog**
- **Colete logs via linhas seriais**
- Mantenha **logs, kernel e aplicações** em sistemas de arquivos distintos.
- Monitore o espaço em disco.
- Armazene dados em **CDs-worm**
- Documente os processos para gerência dos dados de auditoria.

Configure o sistema de logging no cliente para envio de syslog para o loghost.

Elabore uma política e configure S.Os., elementos ativos e aplicações para reportar os eventos nos quais esteja interessado.

9.12 O que deve ser "logado" ?

Dos diversos servidores que podem formar uma rede, todos devem ter um grau de importância. Classificação do que pode ser **logado**:

Servidores mais vulneráveis

- Web servers (públicos, intranet, extranet)
- Servidores de correio visíveis externamente.
- Estações dos admins, DBAs...

Dispositivos geralmente privilegiados

Qualquer sistema que abrigue dados corporativos

- Servidores de bancos de dados
- Repositório de código

9.13 Conteúdo do logging

Pode-se classificar os dados em:

- Eventos normais do dia-a-dia
- Assinaturas de ataques ou falhas dos sistemas
- Mensagens que voce não consegue identificar

Entre os eventos normais estão:

- Atividade autorizada
- Testes autorizados de segurança
- Erros ou problemas conhecidos
- Falsos positivos

Entre os eventos não críticos estão:

- Port scans
- Testes de vulnerabilidades não autorizados
- Tentativas mal sucedidas de comprometimento dos sistemas

Entre os eventos críticos estão:

- Tentativas bem sucedidas de comprometimento
- Ataques para redes de parceiros
- Queda de serviços devido a falhas de hardware ou de software
- Negação de serviço bem sucedida
- Mensagens desconhecidas

9.14 Falhas de Hardware

Muitas falhas de **Hardware** podem ser detectadas simplemente analisando os **logs** dos servidores.

No **UNIX**, geralmente em dispositivos **SCSI**, as mensagens encontradas podem indicar o mau funcionamento de um disco, de uma controladora **SCSI**, placas de rede, Drivers de **CD-ROM**, etc. Em sistemas **UNIX**, a verificação de muitos detalhes de **Hardware** é comum. Discos **IDE ATA IV** com cabos impróprios, falta de terminadores em cabos SCSI, placas de rede, placas de vídeo, memória, drivers de CD-ROM, etc, podem ser sinalizados nos arquivos de **log**.

Falhas de hardware frequentemente incluem palavras como:

- **error**
- **traceback**
- **panic**
- **dumping**
- **booting ...**
- além da popular: **file system full**

9.15 Logs de autenticação

Autenticação de serviços (FTP, TELNET, SSH, POP, etc)

- Sep 12 10:17:11 kuspy PAM_pwdb[17529]:authentication failure;(uid=0) -> tbird for ssh service
- Sep 12 10:17:12 kuspy sshd[17529]: log: Password authentication for tbird accepted.

Logs de autenticação frequentemente incluem palavras como:

- **authentication**
- **failure**
- **success**
- **login**
- **accepted**

Conexões permitidas/negadas em Roteador CISCO: Logs de autenticação frequentemente incluem palavras como:

- **Allowed**
- **Denied**
- **Access**
- **Refused**
- **TCP**
- **UDP**
- **ICMP** (e outros protocolos usados na rede)

9.16 Acesso à Web pages

- 172.20.1.54 - -[11/Feb/2000:20:39:10 -0800]"GET /img/cislogo.gif HTTP/1.0" 200 7607

Pode ser interessante buscar por:

Longas cadeias de dados sem sentido (pode indicar tentativa de buffer overflow)

Tentativas de executar scripts CGI não existentes

Caracteres especiais submetidos a formulários HTML

Tentativas de acesso a arquivos de:

- Senhas
- Configurações de **webserver**
- **ACLs** (access control files)

9.17 Coisas interessantes de procurar

WinNT/Win2K

- Usuário desconhecido ou senha rejeitada
- Reinício do sistema (**System Restart**)
- Descarte de eventos auditados
- Criação de contas
- Designação de direitos/permissões
- Novas relações/domínios de confiança

9.18 Removendo evidências

São ferramentas para adulterar arquivos de **log**. Largamente utilizadas por Hackers e Crackers.

As aplicações mais imediatas são remoção de vestígios de invasão, limpeza de pistas, violação dos arquivos de logs, modificação de conteúdo, etc.

Wipe

http://www.digitaloffense.net/worms/adore/lib_hack_hellno/wi pe-1.00/

Clássico do Unix, multiplataforma.

Seletivamente apaga vários logs (wtmp{,x}, utmp{,x}, lastlog, pacct, etc)

WinZapper, ClearLogs

http://ntsecurity.nu/toolbox/

Apagam seletivamente ou totalmente os logs do Windows

9.19 Analisadores para syslogs

logsurfer, logchecker Ferramentas que comparam certas expressões com textos encontrados nos arquivos de **log**.

As ferramentas para análise de **logs** serão descritas no **Capítulo 10.**

- **http://www.cert.dfn.de/eng/logsurf/**

Win32 <-> Syslog

Backlog

- **http://www.intersectalliance.com/projects/BackLogNT/index.h tml**

NTsyslog

- **http://ntsyslog.sourceforge.net/**

Capítulo 10

Arquivos de log

Os arquivos de **log** (padrão **UNIX**) geralmente se encontram no diretórios **/var/log** ou **/var/adm**. Depende do Sistema Operacional que está sendo utilizado.

Os programas que geram **logs** podem ter seus arquivos de **logs** localizados nestes diretórios ou em outros locais, que devem ser especificados nos arquivos de configuração.

Há muitos softwares que, quando instalados a partir dos arquivos fontes, tem como argumento para compilação e instalação a localização dos arquivos de **log**.

No **UNIX**, os arquivos de **log** são especificados no **/etc/syslog.conf**, como será mostrado nese capítulo.

10.1 Descrição

Dependendo do **UNIX** utilizado, os arquivos trazem informações de muitos serviços.

Muitos serviços podem trazer seus próprios arquivos de **log**. Você pode criar seu próprio Software e utilizar o recurso de **logs**.

Dê uma olhada nos manuais relacionados a **syslog, syslogd, syslog.conf e logger .**

Para diferentes **UNIX**, as configurações são quase parecidas.

Muitos cuidados devem ser tomados em relação aos arquivos de **log**. Permissões e proprietários (usuário e grupo) devem ser bem especificadas.

Como o usuário **root** tem acesso a todos os arquivos do **UNIX**, não é uma boa prática fazer com que os **logs** sejam lidos apenas pelo **root.** Recomenda-se que seja criado usuários específicos para a análise dos logs de um **S.O. UNIX**.

O **root** só deve ser usado em casos de extrema necessidade.

As permissões devem ser bem seguras. Os softwares que geram arquivos de **log** fazem as recomendações mínimas necessárias. Não deixe de seguir estas regras. Todo cuidado é pouco.

Permissões fracas permitem que usuários indesejados alterem os arquivos de log e removam indícios e vestígios de ataques.

Exemplo de um arquivo de **log** de um sistema **UNIX** (**/var/log/messages**) mostrando informações num período de 28 minutos.

```
Jul 9 14:41:31 afthouse su: zemane to root on /dev/ttyp0
Jul 9 14:42:46 afthouse pppd[968]: pppd 2.3.5 started by zemane, uid 1001
Jul 9 14:42:47 afthouse pppd[968]: Connect: ppp0 <–> /dev/cuaa0
Jul 9 14:42:48 afthouse pppd[968]: local IP address 143.107.200.216
Jul 9 14:42:48 afthouse pppd[968]: remote IP address 143.107.200.1
Jul 9 14:43:23 afthouse su: zemane to root on /dev/ttyp1
Jul 9 14:43:36 afthouse su: zemane to root on /dev/ttyp1
```

```
Jul 9 14:44:38 afthouse su: zemane to root on /dev/ttyp1
Jul 9 14:46:02 afthouse kernel: sio0: 92 more interrupt-level buffer overflows (total 92)
Jul 9 14:46:42 afthouse kernel: sio0: 67 more interrupt-level buffer overflows (total 159)
Jul 9 14:59:20 afthouse pppd[968]: Connection terminated, connected for 16 minutes
Jul 9 15:04:37 afthouse su: zemane to root on /dev/ttyp2
Jul 9 15:09:03 afthouse pppd[1405]: pppd 2.3.5 started by zemane, uid 1001
Jul 9 15:09:03 afthouse pppd[1405]: Connect: ppp0 <-> /dev/cuaa0
Jul 9 15:09:04 afthouse pppd[1405]: local IP address 143.107.200.203
Jul 9 15:09:04 afthouse pppd[1405]: remote IP address 143.107.200.1
Jul 9 15:09:15 afthouse su: zemane to root on /dev/ttyp1
```

O trecho de texto acima mostra que o usuário **zemane** se conectou como root no terminal **/dev/ttyp0**, ou seja, o sistema informa que o usuário root está conectado (fez o **login**), uma mensagem sobre usuários conectados.

Em seguida o serviço **pppd** foi inicializado pelo usuário **zemane**.

O **kernel** envia mensagens sobre a utilização da **porta serial sio0** (**MODEM** na porta Serial). E assim por diante...

10.2 Arquivo de configuração syslogd.conf

Como descrito anteriormente, arquivo **/etc/syslogd.conf** estabelece os arquivos e diretórios onde serão armazenados os arquivos de **log** do **UNIX**.

Para se ter uma idéia de como o **syslog** trabalha, eis um exemplo de um arquivo **/etc/syslog.conf** :

"arquivo.um"

```
# $FreeBSD: src/etc/syslog.conf,v 1.26 2003/04/23 13:08:31 des Exp $
# # Spaces ARE valid field separators in this file. However,
# other *nix-like systems still insist on using tabs as field
# separators. If you are sharing this file between systems, you
# may want to use only tabs as field separators here.
# Consult the syslog.conf(5) manpage.
*.err;kern.debug;auth.notice;mail.crit                          /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err   /var/log/messages
security.*                                                      /var/log/security
auth.info;authpriv.info                                         /var/log/auth.log
mail.info                                                       /var/log/maillog
lpr.info                                                        /var/log/lpd-errs
ftp.info                                                        /var/log/xferlog
cron.*                                                          /var/log/cron
*.=debug                                                        /var/log/debug.log
*.emerg                                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*.*                                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.* @loghost
# uncomment these if you're running inn
# news.crit                                                     /var/log/news/news.crit
# news.err                                                      /var/log/news/news.err
```

O exemplo acima é o arquivo (**/etc/syslog.conf**) de um Sistema Operacional **FreeBSD**.

Para os **S.Os. Linux**, o arquivo (**/etc/syslog.conf**) é bem parecido.

Eis um exemplo:

"arquivo.dois"

```
# /etc/syslog.conf Configuration file for syslogd.
# For more information see syslog.conf(5) manpage.
# First some standard logfiles. Log by facility.
#
auth,authpriv.*                                          /var/log/auth.log
*.*;auth,authpriv.none                                   /var/log/syslog
#cron.*                                                  /var/log/cron.log
daemon.*                                                 /var/log/daemon.log
kern.*                                                   /var/log/kern.log
lpr.*                                                    /var/log/lpr.log
mail.*                                                   /var/log/mail/mail.log
user.*                                                   /var/log/user.log
uucp.*                                                   /var/log/uucp.log
#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                                                /var/log/mail/mail.info
mail.warn                                                /var/log/mail/mail.warn
mail.err                                                 /var/log/mail/mail.err
# Logging for INN news system
# news.crit                                              /var/log/news/news.crit
news.err                                                 /var/log/news/news.err
news.notice                                              /var/log/news/news.notice
#
# Some 'catch-all' logfiles.
#
*.=debug;\
   auth,authpriv.none;\
   news.none;mail.none -/var/log/debug
*.=info;*.=notice;*.=warn;\
   auth,authpriv.none;\
   cron,daemon.none;\
   mail,news.none -/var/log/messages
#
# Emergencies are sent to everybody logged in.
#
*.emerg *
#
# I like to have messages displayed on the console, but only on a virtual # console I usually leave idle.
#
#daemon,mail.*;\
# news.=crit;news.=err;news.=notice;\
# *.=debug;*.=info;\
# *.=notice;*.=warn /dev/tty8
# The named pipe /dev/xconsole is for the 'xconsole' utility. To use it,
# you must invoke 'xconsole' with the '-file' option:
#
# $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably busy site..
#
daemon.*;mail.*;\
   news.crit;news.err;news.notice;\
   *.=debug;*.=info;\
   *.=notice;*.=warn
```

|/dev/xconsole

Um trecho importante a se observar:

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

Estas linha mostram que os **logs** relativos a **info, notice e warn** dos serviços de autenticação **(auth), cron e mai**l serão colocados no arquivo **/var/log/messages**.

Não serão colocados no arquivo **/var/log/messages** os **logs** relativos a **"info", "notice" e "warn"** dos serviços **authpriv, daemon e news**.

Este trecho é indicado por : **authpriv.none, daemon.none e news.none**

Observando o **arquivo.dois**, todas as mensagens de **log** referentes a **"mail"** serão colocadas no arquivo **/var/log/maillog**

mail.* -/var/log/mail/mail.log

Há também outras mensagens de **logs** relacionadas ao serviço de **E-Mail:**

```
#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                                        /var/log/mail/mail.info
mail.warn                                        /var/log/mail/mail.warn
mail.err                                         /var/log/mail/mail.err
```

O arquivo de configuração de **logs** (**/etc/syslog.conf**) encontrado no **FreeBSD** (**arquivo.um**) é menor que o arquivo (**/etc/syslog.conf**) encontrado no **Linux**.

Faça uma comparação entre eles e veja as semelhanças, diferenças, etc.

Como mostrado e detalhado anteriormente, as mensagens de **log** são separadas em diferentes arquivos.

O propósito é manter maleável o tamanho de cada arquivo de **log** e facilitar a busca de mensagens nos arquivos, rastrear os arquivos e procurar por determinados padrões de textos tais como: **"permission denied"**, **"syntax error on line 123"**, **"user unkown"**, etc.

Se todas as mensagens de **log**, dos diferentes níveis, fossem colocadas em apenas em um arquivo, com certeza a dificuldade de rastrear e examinar este grande arquivo seria comprometida.

A maneira de armazenar as mensagens de **log** pode variar entre os diversas distribuições **Linux, FreeBSD** e outros **Unixes**.

As modificações no arquivo **/etc/syslog.conf** podem ser efetuadas somente pelo administrador do sistema (**usuário root**).

Após as modificações o **daemon syslog** deverá ser reinicializado.

kill -HUP 'cat /var/run/syslogd.pid'

10.3 Alguns arquivos importantes

- /var/log/syslog
- /var/log/messages
- /var/log/sulog

- /var/log/xferlog
- /var/log/wtmp, /var/run/utmp
- /var/log/auth.log
- outros (no diretório /var/log)

Interpretar alguns dos arquivos acima consiste em apenas visualizar seus conteúdos, pois são arquivos texto simples. Alguns são do tipo **binário** e exigem comandos especiais para fazer a sua interpretação.

Com os comandos **cat, more ou less** o conteúdo dos arquivos texto pode facilmente ser examinados. Com os comandos **zcat, zmore ou zless**, os arquivos **zipados (*.zip, *.gz)** podem ser visualizados.

O mesmo pode ser feito com editores de textos tais como o **vi, pico, nano, ae, vin, ex, etc.**

Deve-se tomar cuidado com o final de cada linha do arquivo texto. É fácil confundir e não perceber os finais de linha e considerar algumas linhas como sendo uma única linha. Para evitar isso é sempre recomendável que se use os editores de textos. Pode-se usar também o comando **view**.

O **editor view** não permite alteração do arquivo texto. É bom para editar e não modificar o conteúdo do arquivo.

Para acompanhar o crescimento de um arquivo de **log** (texto simples) use os comandos **head** e **tail**.

O **head** mostra as linhas iniciais de um arquivo. O **tail** mostra as últimas linhas de um arquivo texto.

O **tail -f arquivo** permite que se veja o contante crescimento de um arquivo texto simples.

Para interpretar arquivos de **log** que estão no formato **binário**, deve-se usar ferramentas apropriadas.

Como exemplo, vamos fazer uma breve interpretação do comando **last**, que mostra os usuários conectados, os que se conectaram ao servidor, datas, duração da conexão, etc.

Este comando ordena as entradas no arquivo, relacionando os tempos de login e logout. Se invocado sem argumentos, o comando **last** exibe toda a informação contida no arquivo.

O arquivo **wtmp** deve ser examinado manualmente através do comando **last**. Este arquivo **não é um texto puro**, é um arquivo tipo **data**, ou seja, **binário.**

O arquivo **/var/log/utmp** (veja **man utmp**) também é um arquivo **binário**.

O comando **last** é usado para identifiar os usuários que já se conectaram ou estão conectados.

O arquivo **wtmp** mostra os usuários que se conectaram. O arquivo **utmp** mostra os usuários que estão conectados.

Pode-se usar o comando **last** com a opção **-f** e especificando o **wtmp** ou o **utmp** como parâmetro.

Exemplos :

$> last -f /var/run/utmp —> vai mostrar o usuários conectados

$> last -f /var/log/wtmp —> vai mostrar os usuários conectados e os que se conectaram.

Para maiores detalhes **: man last (ou info last)**

O **syslog** é um mecanismo que permite que qualquer comando registre mensagens de erro e informativas na console do sistema e/ou em um arquivo.

Normalmente mensagens de erro são gravadas no arquivo **/var/log/messages** juntamente com a data e hora em que foram gravadas. Os arquivos de **log** encontrados no diretório **/var/log** possuem permissões de leitura e escrita somente para o **root** e permissão de leitura para os demais usuários. Muitos podem ser consultados apenas pelo **root**.

Muitos **UNIX** apresentam subdiretórios dentro do **/var/log**. Alguns subdiretórios guardam os **logs** para serviços bem específicos, tais como o Servidores de E-Mail (Sendmail, Postfix, QMail, etc).

Cada distribuição **UNIX** apresenta suas particularidades. A localização dos arquivos de **log** de cada programa depende das configurações destes.

Arquivos padrões de **logs** são armazenados no diretório **/var/log**.

Pode-se modificar os nomes dos arquivos e sua localização. O trecho abaixo faz o servidor de **DNS** armazenar os **logs** no arquivo **/var/log/quartzo.named.log.**

Os níveis de **log** são mais detalhados.

```
logging {
channel my_default {
file "/var/log/quartzo.named.log";
severity info;
print-time yes;
print-category yes;
print-severity yes; };
category default { my_default; default_debug; };
category panic { my_default; default_stderr; };
category packet { default_debug; };
category eventlib { default_debug; };
category lame-servers { null; }; }
```

10.4 Rotacionando os logs

O objetivo é fazer com que os arquivos de **log** sejam preservados, fazer um **backup** e criar novos arquivos frequentemente.

Com o rotacionamento os arquivos de **log** são **zipados**, renomeados e novos arquivos são criados.

As informações mais velhas ficam nos arquivos compactados.

Se não se faz o rotacionamento, o tamanho do arquivo tende a crescer muito e a sua manipulação torna-se praticamente impossível.

A maneira mais simples é remover os arquivos de **log** e reinicializar o **daemon syslogd**.

```
$> rm /var/log/messages
$> kill -HUP `cat /var/run/syslogd.pid`
```

Este método é bem aplicado num servidor **Linux** que não esteja em rede, ou num simples **Desktop com Linux**.

Em servidores Unix os arquivos de **log** precisam ser preservados para uma análise futura, uma auditoria, rastreamento de brechas de segurança, etc.

Neste caso a estratégia é diferente. **Os arquivos de log deverão ser mantidos.**

Deve-se renomear os arquivos antigos e criar arquivos novos.

Os arquivos antigos podem ser compactados e guardados em sistemas de **backup**.

```
$> mv /var/log/messages /var/log/messages.1
$> kill -HUP `cat /var/run/syslogd.pid`
```

Se se pretende criar duas gerações de histórico de **logs**, será necessário mover a primeira geração de arquivos para a segunda geração e, então, mover a geração atual para a primeira geração.

```
$> mv /var/log/messages.1 /var/log/messages.2
$> mv /var/log/messages /var/log/messages.1
$> kill -HUP `cat /var/run/syslogd.pid`
```

Aconselha-se a empacotar os arquivos de **log** antigos:

$> gzip /var/log/messages.1
$> gzip /var/log/messages.2

Em muitos **UNIX** este procedimento é automatizado.
No **Linux Debian**, e em outras distribuições, os scripts estão nos diretórios **/etc/cron.daily, /etc/cron.weekly** e **/etc/cron.monthly**.
Estes **scripts** devem ter permissão de execução.
Verifique estes arquivos e examine os **shell scripts** que realizam o rotacionamento dos **logs**.

10.5 Exemplos de arquivos de log

A seguir são mostrados alguns exemplos de arqivos de **log:**

- **$> cat /var/log/vsftpd.log | grep Debian**

Arquivo de log do servidor VSFTP, instalado num servidor **Linux Debian**

Mon Mar 15 12:32:22 2004 6 143.107.200.102 317706 /pub1/Cursos.CIRP/Curso.Adm.Linux.Debian.CIRP.pdf b _ o a mozilla@example.com ftp 0 * c

Mon Mar 15 18:05:33 2004 4 143.107.200.102 281448 /pub1/Cursos.CIRP/Curso.Adm.Linux.Debian.CIRP.quase.final.22.Julho.2003.pdf b _ o a mozilla@example.com ftp 0 * c

Mon Mar 15 18:11:47 2004 5 143.107.200.102 281448 /pub1/Cursos.CIRP/Curso.Adm.Linux.Debian.CIRP.quase.final.22.Julho.2003.pdf b _ o a -wget@ ftp 0 * c

Mon Mar 15 18:17:11 2004 5 143.107.200.102 317706 /pub1/Cursos.CIRP/Curso.Adm.Linux.Debian.CIRP.pdf b _ o a mozilla@example.com ftp 0 * c

- **$> cat /var/log/vsftpd.log.* | grep BSD**

Fri Mar 12 21:09:29 2004 1 200.187.167.146 9037 /FreeBSD/Dicas/experiencia.com.fortran b _ o a mozilla@example.com ftp 0 * c

Mon Mar 1 17:44:46 2004 1 200.222.74.74 9037 /FreeBSD/Dicas/experiencia.com.fortran b _ o a Squid@ ftp 0 * c

Tue Mar 2 11:46:52 2004 1 200.225.73.100 173 /pub1/FreeBSD/releases/i386/supfile b _ o a libramar.png@libramar.com.br ftp 0 * c

Thu Mar 4 15:05:53 2004 53 217.148.68.113 1480512 /pub1/OpenBSD3.2/OpenBSD32-i386-base.iso b _ o a Squid@ ftp 0 * i

Thu Mar 4 15:45:13 2004 3 200.181.17.112 451 /pub1/FreeBSD/ISO-IMAGES-i386/4.7/pegar b _ o a mozilla@example.com ftp 0 * c

Thu Mar 4 15:45:30 2004 1 200.181.17.112 451 /pub1/FreeBSD/ISO-IMAGES-i386/4.7/pegar b _ o a mozilla@example.com ftp 0 * c

Thu Feb 26 17:29:33 2004 1 143.107.70.187 183 /pub1/OpenBSD3.2/pegar b _ o a mozilla@example.com ftp 0 * c

Thu Feb 26 17:29:46 2004 1 143.107.70.187 75 /pub1/OpenBSD3.2/pegar2 b _ o a mozilla@example.com ftp 0 * c

Thu Feb 26 17:29:52 2004 1 143.107.70.187 65 /pub1/OpenBSD3.2/MD5SUM b _ o a mozilla@example.com ftp 0 * c

Fri Feb 20 10:31:11 2004 31 161.24.64.254 1136012 /pub1/OpenBSD3.2/OpenBSD32-i386-base.iso b _ o a IE40user@ ftp 0 * i

Fri Feb 20 10:31:34 2004 22 161.24.64.254 1004940 /pub1/OpenBSD3.2/OpenBSD32-i386-base.iso b _ o a IE40user@ ftp 0 * i

Fri Feb 20 10:32:09 2004 34 161.24.64.254 1519172 /pub1/OpenBSD3.2/OpenBSD32-i386-base.iso b _ o a IE40user@ ftp 0 * i

Fri Feb 20 13:29:49 2004 14698 200.161.147.235 645169152 /FreeBSD/5.1-RELEASE/5.1-RELEASE-i386-disc1.iso b _ o a anon@ ftp 0 * c

- **$> cat /var/log/vsftpd.log.* | grep Slackware**

Sat Mar 13 23:20:22 2004 1 201.4.162.96 184 /Slackware9.0/slackware/l/libxml2-2.5.4-i386-1.txt b _ o a mozilla@example.com ftp 0 * c

Sat Mar 13 23:28:46 2004 491 201.4.162.96 1434976 /Slackware9.0/slackware/l/libxml2-2.5.4-i386-1.tgz b _ o a mozilla@example.com ftp 0 * c

Sat Mar 13 23:50:56 2004 156 201.4.162.96 629704 /Slackware9.0/slackware/l/libtiff-3.5.7-i386-3.tgz b _ o a mozilla@example.com ftp 0 * c

- **$> cat /var/log/vsftpd.log.* | grep Mandrake**

Sun Mar 7 05:53:05 2004 1 200.171.10.214 7912 /Mandrake9.1/INSTALL.txt a _ o a anon@ ftp 0 * c

Sun Feb 22 10:28:36 2004 2 200.158.225.44 143310 /Mandrake9.1/pkg-9.1-Bamboo-i586.idx b _ o a Squid@ ftp 0 * c

Wed Feb 18 15:51:30 2004 201 200.201.164.11 9698952 /Mandrake9.1/ISO/Mandrake91-cd1-inst.i586.iso b _ o a proxyuser@proxy.caixa ftp 0 * i

- **$> tail -5 /var/log/dns.log**

20-Mar-2004 17:38:26.414 maintenance: info: Cleaned cache of 2163 RRsets

20-Mar-2004 17:38:26.483 statistics: info: USAGE 1079804306 1079534305 CPU=31.9493u/21.5371s CHILDCPU=0u/0s
20-Mar-2004 17:38:26.483 statistics: info: NSTATS 1079804306 1079534305 TYPE0=431 A=125083 NS=67 CNAME=8 SOA=1870 PTR=167113 MX=19477 AAAA=23608 SRV=1981 A6=2201 ANY=27473

20-Mar-2004 17:38:26.483 statistics: info: XSTATS 1079804306 1079534305 RR=213515 RNXD=74194 RFwdR=59395 RDupR=593 RFail=4017 RFErr=8849 RErr=1851 RAXFR=0 RLame=7432 ROpts=0 SSysQ=89629 SAns=398279 SFwdQ=82779 SDupQ=127745 SErr=5 RQ=370758 RIQ=0 RFwdQ=82779 RDupQ=12856 RTCP=1548 SFwdR=59395 SFail=265 SFErr=0 SNaAns=122215 SNXD=170870 RUQ=0 RURQ=0 RUXFR=0 RUUpd=484

20-Mar-2004 17:54:06.805 default: info: Response from unexpected source ([149.174.211.3].9052) for query "dns-02.ns.aol.com IN AAAA" 20-Mar-2004 17:56:40.521 update-security: notice: denied update from [143.107.200.190].2365 for "cirp.usp.br" IN

20-Mar-2004 17:56:40.550 update-security: notice: denied update from [143.107.200.190].2370 for "200.107.143.in-addr.arpa" IN 20-Mar-2004 17:57:01.205 default: info: ns_forw: query(3.186.53.195.in-addr.arpa) No possible A RRs

- **$> zcat /var/log/maillog.7.gz | grep virus**

Mar 12 21:50:13 quartzo sendmail[90697]: i2CLoDnB090697: Authentication-Warning: quartzo.cirp.usp.br: vscan set sender to virusalert@cirp.usp.br using -f

Mar 12 21:50:13 quartzo sendmail[90697]: i2CLoDnB090697: from=virusalert@cirp.usp.br, size=1269, class=0, nrcpts=1, msgid=<VAjPxmbCYF@quartzo.cirp.usp.br>, relay=vscan@localhost

Mar 12 21:50:13 quartzo sendmail[90697]: i2CLoDnB090697: to=virusalert@cirp.usp.br, delay=00:00:00, mailer=esmtp, pri=31269, dsn=4.4.3, stat=queued Mar 12 21:50:13 quartzo sendmail[90698]: i2CLoDS5090698: Authentication-Warning: quartzo.cirp.usp.br: vscan set sender to virusalert@cirp.usp.br using -f

Mar 12 21:50:13 quartzo sendmail[90698]: i2CLoDS5090698: from=virusalert@cirp.usp.br, size=666, class=0, nrcpts=1, msgid=<VRjPxmbCYF@quartzo.cirp.usp.br>, relay=vscan@localhost

Mar 12 21:50:13 quartzo amavis[90640]: (jPxmbCYF) INFECTED (Worm.SomeFool.Gen-2), <alexmgarcia2002@hotmail.com> -> <usuario@cirp.usp.br>, quarantine virus-20040312-215013-jPxmbCYF, Message-ID: , Hits:

Estas linhas mostram os e-mails recebidos com vírus. O usuário **virusalert@cirp.usp.br** foi notificado.

- **$> zcat /var/log/messages.8.gz**

Ese trecho mostra problemas de **Hardware**. Os **S.O. UNIX** fazem identificação detalhada dos problemas de **Hardware** e reportam em arquivos de **log.** O trecho inicial mostra o problema de **reset** constante na controladora **SCSI** e o trecho final não mostra estes problemas.

Aug 13 19:02:18 linorg kernel: SCSI bus is being reset for host 0 channel 0.

Aug 13 19:02:18 linorg kernel: (scsi0:0:0:0) Synchronous at 20.0 Mbyte/sec, offset 15.

Aug 13 19:02:19 linorg kernel: (scsi0:0:1:0) Synchronous at 20.0 Mbyte/sec, offset 15.

Aug 13 19:02:20 linorg kernel: scsi0 channel 0 : resetting for second half of retries.

Aug 13 19:02:20 linorg kernel: SCSI bus is being reset for host 0 channel 0.

Aug 13 19:02:20 linorg kernel: (scsi0:0:1:0) Synchronous at 20.0 Mbyte/sec, offset 15.

Aug 13 19:02:23 linorg kernel: scsi0 channel 0 : resetting for second half of retries.

Aug 13 19:02:23 linorg kernel: SCSI bus is being reset for host 0 channel 0.

Aug 13 19:02:23 linorg kernel: (scsi0:0:1:0) Synchronous at 20.0 Mbyte/sec, offset 15.

Aug 13 19:02:47 linorg kernel: (scsi0:0:0:0) Synchronous at 20.0 Mbyte/sec, offset 15.

Aug 13 19:13:25 linorg – MARK –

Aug 13 19:33:25 linorg – MARK –

Aug 13 19:53:25 linorg – MARK –

Aug 13 20:13:25 linorg – MARK –

Aug 13 20:33:25 linorg – MARK –

- **$> cat /var/log/apache/access.log | grep "cmd.exe"**

Trechos do arquivo de access.log do Apache Web Server.

61.115.118.197 - - [14/Mar/2004:07:47:35 -0300] "GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 302 229

61.115.118.197 - - [14/Mar/2004:07:47:35 -0300] "GET /NULL.printer HTTP/1.0" 302 229

A primeira linha mosta atentativa de se executar o comando **dir**, do DOS.
A segunda linha mostra atentativa de se enviar algo para a impressora.

- **$> cat /var/log/apache/error.log | tail -5**

[Sun Oct 5 06:25:19 2003] [error] [client 64.68.80.44] File does not exist:
/raid/Conectiva7/Disk2/doc/howto/Online-Troubleshooting-HOWTO

[Sun Oct 5 06:25:39 2003] [error] [client 66.77.73.72] File does not exist:
/raid/Conectiva7/Disk2/doc/guias/guia_de_instalacao/node215.html

[Sun Oct 5 06:25:52 2003] [error] [client 66.77.73.72] File does not exist: /raid/Slackware8.0//bootdsks.144/optics.i

[Sun Oct 5 06:26:43 2003] [error] [client 64.68.80.41] File does not exist:
/raid/Conectiva7/Disk1//conectiva/cncimage/etc/X11/xkb/symbols/ua

[Sun Oct 5 06:27:20 2003] [notice] SIGUSR1 received. Doing graceful restart

- **$> tail -9 /var/log/auth.log**

Mar 22 10:15:52 quartzo sshd[47934]: Accepted keyboard-interactive/pam for rubens from 143.107.200.46 port 1315 ssh2
Mar 22 11:09:46 quartzo sshd[48811]: Accepted keyboard-interactive/pam for aftaha from 143.107.207.36 port 1029 ssh2
Mar 22 11:14:57 quartzo su: aftaha to toor on /dev/ttyp0
Mar 22 11:45:09 quartzo sshd[49502]: Accepted keyboard-interactive/pam for aftaha from 143.107.207.36 port 1031 ssh2
Mar 22 11:47:56 quartzo login: login on ttyv0 as rubens Mar 22 14:30:49 quartzo sshd[51096]: Accepted keyboard-interactive/pam for aftaha from 143.107.200.102 port 1171 ssh2
Mar 22 14:30:54 quartzo su: aftaha to toor on /dev/ttyp0
Mar 22 15:29:45 quartzo sshd[87]: Server listening on 0.0.0.0 port 22.
Mar 22 15:29:54 quartzo webmin[227]: Webmin starting
Mar 22 15:45:08 quartzo sshd[87]: Server listening on 0.0.0.0 port 22.

Capítulo 11

Ferramentas para análise de logs

As mais comuns são:

- **LogSentry** (Analisa expressões regulares nos arquivos de log do UNIX)
- **Logcheck** (sumariza os logs do UNIX e envia resultados por E-Mail)
- **Logmuncher** (variação do Logcheck)
- **Logsurf** (logs do UNIX, capaz de criar regras dinâmicas)
- **Swatch** (Simple Watch Dog, para arquivos de log do UNIX)
- **Sec** (correlacionador de eventos simples)
- **Lire** (logs do UNIX, de servidores WEB, etc)
- **Webalizer** (logs de Servidores WEB)
- **WebLog** (logs de Servidores WEB e Proxy)
- **Analog** (logs de Servidores WEB)
- **Calamaris** (logs do Servidor Proxy Squid)
- **Logguard (**sumariza os logs do UNIX e envia resultados por E-Mail)

Mais ferramentas podem ser encontadas nos seguintes **URLs**:
http://fmg-www.cs.ucla.edu/fmg-members/geoff/logmuncher.html,
http://www.hotscripts.com/Tools_and_Utilities/Log_Analyzers/

Uma das tarefas do administrador de sistemas é a monitoração da segurança.

Esta tarefa envolve o exame de arquivos de **log** para detectar acessos não autorizados, bem como a monitoração de falhas de segurança.

As contas devem ser monitoradas periodicamente de modo a verificar dois eventos: usuários que se conectam quando não devem (por exemplo, tarde da noite ou quando estão de férias) e usuários executando comandos que normalmente não deveriam usar.

O arquivo **/var/log/lastlog** (no **Linux**) registra o **login** mais recente de cada usuário do sistema.

A mensagem impressa no terminal a cada vez que um usuário se conecta utiliza a data armazenada no arquivo **lastlog**

Last login: Sat Mar 10 10:50:48 from **hostname.servidor.br**.

A data do último **login** relatada pelo comando **finger** (em desuso por questões relacionadas a vulnerabilidades) também usa estes dados.

Os usuários devem ser alertados a inspecionar esta data para certificarem-se de que não foi efetuado nenhum acesso não autorizado às suas contas e, caso positivo, alertar o Administrador de Sistemas para o ocorrido.

Algumas ferramentas disponíveis facilitam a interpretação dos arquivos de **log**.

Estas ferramentas possuem em seus arquivos de configuração instruções para procurar certas palavras, **strings** ou textos nos arquivos de **log** e enviar algum tipo de sinal para o Administrador de Sistemas. Por exemplo, se encontrar expressões como **access denied** em certos arquivos de **log**, um **e-mail** deverá ser enviado ao Administrador. Ele vai receber este **e-mail** e tomar alguma providência.

Basicamente esta é a função das ferramentas para **Análise de logs**.

A seguir serão descritas algumas ferramentas disponíveis para os **S.O. UNIX.**

11.1 LogSentry

É um verificador de **log** no estilo **cron**. Usa vários arquivos contendo expressões regulares **egrep** simples e os combina com as linhas no arquivo de **log** para determinar se um relatório deve ser feito. Os relatórios são remetidos ao **root** ou a outro usuário. Vários arquivos contêm as expressões regulares usada pelo **LogSentry**, como mostrado abaixo:

logcheck.hacking:

Expressões que definitivamente indicam atividade de invasão. Quaisquer mensagens que combinam são remetidas com um cabeçalho antipático, para chamar a atenção imediatamente.

logcheck.violations:

Expressões que indicam atividades impróprias, mas não tão sérias como aquelas em **logcheck.hacking**

logcheck.violations.ignore:

Expressões que são realmente benignas**.** Se uma linha combinar com uma regra em **logcheck.violations**, mas também combinar com uma regra em **logcheck.violations.ignore**, ela não será informada. Por exemplo, esse arquivo lhe permite apanhar mensagens contendo **refused** (como **TCP connection refused)** sem informar mensagens inocentes, como a possibilidade de Sendmail se conectar a um servidor de correio (que cria uma mensagem com **stat=refused**). Também usado para eliminar falsos positivos.

logcheck.ignore:

Se nenhuma combinação tiver sido feita até aqui, a linha será informada, a menos que haja uma combinação no arquivo **logcheck.ignore.**

LogSentry vem com padrões **default** embutidos de **logs** dos ataques de **Internet Security Scanner (ISS)**, mensagens **FWTK (o FireWall ToolKit**, **http://www/fwtk.org**), **wrappers TCP** e mensagens específicas do **Linux**, de modo que já é adequado para uma instalação **Linux** padrão.

LogSentry é escrito em Bourne Shell e C. Ele inclui um utilitário chamado **Logtail** que trata automaticamente da leitura apenas da nova parte dos arquivos de **log**, registrando os números de linha analisados. O sistema é baseado no script **frequentcheck.sh**, escrito por Marcus Ranum e Fred Avolio para o **Firewall Gauntlet**, embora nenhum código seja compartilhado entre eles.

11.2 Ferramenta Logsurfer

Escrito por Wolfgang Ley e Uwe Ellerman no DFN-CERT da Alemanha.

(http://www.cert.dfn.de/eng/logsurf).

É capaz de criar regras dinâmicas e agrupar linhas de **log** em contextos. Enquanto muitas ferramentas operam e geram apenas mensagens de **log** de única linha, o **logsurfer** permite quebrar as mensagens em contextos separados permitindo a análise detalhada.

Se, por exemplo, você viu que alguém conseguiu gravar arquivos em um servidor **FTP** que não deveria ter diretórios com permissão de escrita, provavelmente desejaria determinar quem foi o usuário que gravou arquivos neste diretório.

Como a maior parte do software de verificação de **log**, você teria de ir até o arquivo de **log** original e combinar a linha relatada (a gravação FTP) com o login do usuário, que provavelmente foi ignorada no relatório, pois presume-se que muitas destas linhas estariam presentes.

A configuração do **logsurfer** é um pouco complexa. Ele usa expressões regulares (**regexes** padrão, e nao expressões estendidas em Perl) para

determinar quando uma linha combina com a expressão que voce especifica.

Formato das linhas de configuração:
match-exp not-match-exp stop-exp not-stop-exp timeout action

match-exp : Expressão regular que indica uma combinção e que essa linha deverá ser processada.

not-match-exp : Se o **match-expressão** combinar, mas o **not-match-exp** também combinar, nao a considere como uma combinção (permite a lógica se/mas-não)

stop-exp : Apaga essa regra se a linha combinar com **stop-exp**.

not-stop-exp : Semelhante a not-match-exp, isso significa "apague a regra se **stop-exp** combinar, a menos que **not-stop-exp** também combine"

timeout : Número de segundos em que essa regra devera ser ativa (0 significa tempo sem limite)

action : Uma ação da próxima lista. As acões podem ser seguidas por argumentos opcionais.

Regras permitidas para o campo **'action'** :

ignore : Ignore esta regra

exec : Executa o programa expecificado

pipe: Executa o porgrama especificado e lhe envia a linha de **log** como entrada padrão

open: Inicia um contexto

delete: Apaga um contexto

report: Abre um programa e lhe envia todas as definições de contexto especificadas

rule: Cria uma regra dinâmica

Esta ferramenta oferece mais controle sobre exatamente o que é registrado em **log**, mas é complicado para a configuração e pode consumir muito espaço de memória e CPU do sistema. Por exemplo, embora o padrão para a maioria dos verificadores de **log** seja gerar saída, você precisa chamar explicitamente **/bin/echo** com a opção de **pipe** para realizar qualquer saída do **logsurfer**.

Logsurfer é mais usado para análise de **log** muito específica em conjunto com **LogSentry ou Swatch**, para a verificação de **log** mais profunda.

11.3 Ferramenta Sec

(http://kodu.neti.ee/~risto/sec).

O coordenador de eventos simples, analisa um arquivo, **named pipe** ou entrada padrão. Usando expressões regulares, ele reconhece eventos e pode executar comandos do sistema no caso de uma combinção bem sucedida. Ele pode analisar arquivos de **log**, mas também pode ser integrado a serviços de rede arbitrários, procurando sinais de explorações e realizando comandos quando for apropriado.

Esta ferramenta pode analisar linhas de texto isoladas, várias linhas de texto ou pares de linhas (uma seguida por outra) e pode procurar um limite de linhas que combinam em determinado período de tempo, ignorar certas linhas de texto e realizar ações em horários determinados.

11.4 Ferramenta Lire

(http://logreport.org/lire)

É uma sofisticada ferramenta de análise de **logs** que pode monitorar e criar relatórios de resumo a partir de diversos arquivos de **log** diferentes. Pode ser instalado em seu servidor ou então submeter os **logs** ao **"engine"** de relatório de **Lire** pela internet, recebendo os relatórios de volta por **e-mail**.

O programa **lr_anonymize**, que vem com o pacote **Lire**, torna seus **logs** anônimos, envia para serem processados e depois retira o anonimato dos resultados, quando chegarem por **e-mail.**

Lire possui uma grande lista de formatos de arquivos de log aceitos, incluindo os seguintes:

- Sendmail, Postfix, qmail, exim e nms
- Formato de log comum e combinado do Apache, mod_gzip do Apache
- DNS Bind versoes 8 e 9
- Firewalls: Cisco, ipfilter, ipchains e iptables
- FTP xferlog
- Logs de impressora CUPS e LPRng
- Servidores Proxy Squid e WELF
- Banco de Dados MySQL

Lire converte esses formatos para o **Distilled Log Format (DLF)**, que depois é processado.

Os relatórios são muito úteis tanto para detectar as anomalias como também para ajudá-lo a sintonizar melhor seu sistema e entender suas necessidades específicas.

11.5 Protegendo seus arquivos de log

Se os arquivos de **log** estiverem com as permissões corretas, eles ainda podem ser adulterados.

Se o **Hacker** ou **Cracker** obtiver acesso **roo**t, os arquivos de **log** estão comprometidos.

As permissões normais não impedirão que os arquivos de **log** sejam editados.

Usando permissões específicas de **filesystem**, pode-se impedir que até mesmo o usuário **root** mexa nos arquivos de **log**.

Para os **filesystem ext2 e ext3** o comando que protege os arquivos é o **"chattr"** : **chattr +a /var/log/messages**, coloca o arquivo messages no modo de apenas acréscimo (**append**), o que significa que o **cracker** não pode mais apagar ou excluir o arquivo, somente acrescentar algo a ele. Isso permite que o processo **syslog** continue enviando novos **logs** ao arquivo, mas nenhum processo poderá mexer nos **logs** antigos.

O **chattr** só impede os usuários mais novatos, os **"script kiddies"**.

Os mais experientes podem usar o comando **chattr** com o atributo **-a**, que remove a permissão de modificações.

No **Sistema Operacional FreeBSD** há o comando **chflags**, que permite trabalhar com os **flags** dos arquivos, tornando-os imutáveis, **append-only e undeletable**. Permite também que estruturas inteiras de diretórios, ou partições, sejam consideradas imutáveis, ou seja, o diretório

/sbin, por exemplo, pode ter todos os seus arquivos apenas para execução, não se consegue adicionar nenhum arquivo neste diretório.

Há outros niveis de segurança mais específicos, chegando até a fazer com que nada seja alterado no Sistema Operacional.

O Sistema Operacional fica totalmente **Read Only**.

O assunto e' bastante extenso e deve ser melhor analisado em cursos relacionados a **Segurança em UNIX.**

11.6 Daemon Syslog-ng

Syslog-ng (**Next generation logging daemon**) (http://www.balabit.com/products/syslog-ng) é um daemon de **logging** do sistema melhor do que **syslogd**, embora normalmente não esteja instalado como o **default**. O arquivo de configuração para **syslog-ng**, chamado **syslog-ng.conf**, é radicalmente diferente de um arquivo **syslog.conf** normal. Assim como **syslog**, você pode especificar vários destinos (arquivos locais, servidores remotos e assim por diante).

Contudo, você também pode definir as origens das mensagens e atuar de modo diferente para gerar eventos localmente **versus** mensagens **syslog** remotas, por exemplo.

Ainda mais poderosa é a capacidade de filtrar mensagens com base em **expressões regulares**, em vez de simplesmente jogar todas as mensagens do **daemon** para um único destino, para analisá-la manualmente.

Syslog-ng pode enviar e receber mensagens **TCP**, além de **UDP**, o que significa que você pode ativar um **syslogging** confiável (**TCP** garante remessa de pacote, enquanto **UDP** não). Apenas por esse motivo, **syslog-ng** pode ser mais útil em ambientes em que você precisa ter certeza de que nenhum **log** será perdido ao enviar seus **logs** para um **host** de **logging** dedicado.

Os manuais de **syslog-ng.conf e syslog-ng** mostram como deve ser os formatos dos arquivos de **log** gerados.

11.7 Cuidados com os arquivos de log

Os arquivos de **log** devem ser protegidos contra alterações. As permissões devem ser de leitura para apenas um usuário e um grupo de usuários.

- Criar um grupo e um usuário para trabalhar com os arquivos de **log** é uma boa medida.
- O diretório **/var/log** não pode ser gravado por nenhum outro usuário além do **root** ou outro criado especialmente para manipular este diretório.
- É recomendável que se faça um **backup** do diretório **/var/log** constantemente. Use o **cron** para estas tarefas.
- Impeça a alteração e a remoção dos arquivos de **log.**
- É possível inserir entradas de **log** falsas nos arquivos de **log**. O comando **logger** permite que isso seja feito.
  $> logger -p facility.level "message"
- Se o **cracker** cirar entradas de **log** semelhantes a essa mensagem de erro, poderá enganar o Administrador de Sistemas para que pense que outro usuário está tentando obter acesso **root**.
  Ele poderia tentar o seguinte:
  $> logger -p kern.alert "authentication failure; logname=public uid=509 guid=0 tty= ruser= rhost= user=root"
  No **Linux Debian,** a linha acima altera o arquivo **/var/log/syslog.** No **FreeBSD** o arquivo adulterado é o **/var/log/messages.**

- Leia os **logs** cuidadosamente. Uma boa análise permite identificar as atividades do invasor, dos **hackers, crackers e outros.**
- Analise o conteúdo atentamente. No exemplo anterior, usando o comando **logger**, parece uma tentativa de **login root** sem sucesso.
- O comando **logger pode ser executado por qualquer usuário no UNIX. Cuidado com falsos alertas.**
- Confie nos seus **logs** assim como pode confiar nos seus usuários. Não chegue a conclusão fácil até que tenha comparado com outra evidência de intrusão.

Exercícios:

1. Instalação e configuração das ferramentas (escolher uma): **Logcheck, Logsurf, Swatch, Sec, Lire, Webalizer, WebLog, Analog.**
2. Personalizar os arquivos de configuração.
3. Configurar a ferramenta escolhida de forma a receber notificações via e-mail.
4. Configurar o **Linux** de modo a exibir os **logs** num terminal **TTY.**
5. Estabelecer um **loghost** na rede que está sendo utilizada.
6. Instalação e configuração do **syslog-ng**.
7. Instalação e utilização da ferramente **sysstat**. Examinar os recursos relacionados a monitoração de dispositivos de I/O, Processadores, Memória, etc.
8. Elaborar um sistema de **backup** dos arquivos de **log**. Utilize o comando **tar** e coloque o serviço **cron** para efetuar o **backup** periodicamente.
9. Usar o comando **logger** para alterar os arquivos de **log**. Explique os problemas que podem ocorrer na utilização deste recurso.
10. Elabore uma rede com servidores e **loghosts** distribuídos. Explique as configurações que podem ser utilizadas.

Exemplos de logs em páginas WEB, utilizando os Software WEBALIZER e ANALOG :

- http://www.linorg.cirp.usp.br/webalizer2/index.html
- http://www.linorg.cirp.usp.br/Analog/stats.html

Capítulo 12

Ferramenta administrativa WEBMIN

Administrar um S.O. Unix exige o conhecimento de muitos serviços de rede, técnicas de implementação, configuração, etc.

Conhecer os arquivos de configuração, a localização destes arquivos, a sintaxe que deve ser empregada, dependência de outros arquivos, formato do arquivo, permissões, dentre outros detalhes. Esses detalhes muitas vezes escapam do controle, eda memória, do administrador.

Ferramentas gráficas são largamente utilizadas para administração de sistemas Unix. Há ferramentas proprietárias, de alto custo, que são fundamentais para a configuração e manutenção de diversos serviços de um S.O. Unix.

Estas ferramentas exigem que o administrador esteja diante do servidor. Para administração remota do S.O. Unix a Internet se torna fundamental.

O Software Webmin é uma dessas ferramentas que pode ser gerenciada pela WEB. É de livre distribuição e pode ser usado em muitos Unixes.

12.1 Instalação e configuração

O Webmin é um sistema baseado em administração de sistemas Unix via Web Browsers.

Permite que sejam administrados os diversos recursos dos servidores, usuários, gerenciamento de pacotes, Sistemas de Arquivos, Quotas, etc.

Permite o controle de diversos servidores, tais como: FTP, SMTP, WWW, DHCP, DNS, etc.

É composto por módulos para administrar Interfaces de rede, partições, impressoras, e o próprio Webmin.Ainstalaçãoé bastantesimples:

1 - Obtenha o WEBMIN do site http://www.webmin.com

2 - Faça o download da última versão, que estão em formato RPM e TAR.GZ

3 - Coloque o arquivo (extensão .tar.gz) no diretório /usr/local

4 - Desempacote o arquivo com o comando : tar -zxvf webmin.versao.tar.gz

5 - Entre no diretório /usr/local/webmin.versao

6 - Execute o shell script setup.sh para efetuar a instalação

7 - Responda às perguntas e estabeleça o diretório de configuração (geralmente /etc/webmin), estabeleça o login do administrador e sua senha, escolha a porta de conexão (o valor inicial é 10000. Não coloque o número da porta menor ou igual a 1024)

8 - No final da instalação você deve optar por **iniciar o Webmin na reinicialização da máquina ou não**

9 - A instalação termina e o Webmin está ativo.

10 - Entre no ambiente gráfico preferido e utilize um Browser apontando para o seguinte URL: http://localhost:(porta do webmin), por exemplo : http://localhost:10000

11 - Se você instalou o SSL no seu Linux, que é uma implementação do Suporte para Transações Seguras (SSL) da Netscape, e na instalação do Webmin escolheu utilizar o SSL, o URL que você deve apontar no Browser é diferente: aponte para https://localhost:10000

12 - No diretório /etc/webmin tem todos os scripts de configuração do Webmin.

12.2 Utilização

Faça uma visita a todos os itens do Webmin e procure ver os recursos disponíveis.

Entre no módulo WEBMIN e configure as preferências: Temas, controle de acesso, Módulos, etc.

Visite os itens : System, Servers, Networking, Hardware, Cluster e Others.

Escolha os módulos que deverão ser configurados e veja com detalhes seus recursos e parâmetros exigidos.

12.3 Gerenciamento de pacotes via WEBMIN

Configurar um serviço de rede no Unix exige conhecimento prévio, experiência e muita atenção.

Antes de utilizar o Webmin procure saber os detalhes do serviço de rede que vai ser configurado, suas características, detalhes, exigências e, principalmente, tenha sempre um bom manual à mão.

Sabendo trabalhar com o serviço de rede específico, vá para o Webmin e aprenda a fazer a configuração deste serviço.

Não é aconselhável iniciar imediatamente com o Webmin. As coisas podem se complicar e todas as configurações podem ser feitas de maneira errada.

12.4 Configuração dos módulos

Todo módulo tem um item de configuração, onde devem ser estabelecidos alguns PATHs para arquivos,

argumentos para alguns comandos e outros detalhes. Cada módulo tem também um manual de utilização.

Muitos trazem os parâmetros default, mas você pode alterar estes parâmetros especificando-os corretamente de acordo com o seu Unix.

Se o Webmin está instalado num Unix FreeBSD, por exemplo, as configurações dos módulos serão diferentes do Webmin instalado num Unix Solaris ou Linux Slackware.

Há módulos que permitirão que arquivos de configuração sejam editados e seus dados alterados. Tome cuidado com o que está sendo feito, pois você está alterando arquivos de configuração de serviços do Unix.

Isso é delicado e se houver erros as coisas se complicam ainda mais.

12.5 Gerenciamento dos módulos

Na instalação do Webmin muitos módulos padrões são instalados. Outros podem ser adicionados. Visite o site do Webmin e conheça os módulos extras que podem ser adicionados. Alguns são comerciais, ou seja, tem que ser comprados, e outros são de livre distribuição.

Veja a descrição do módulo extra, entenda seu funcionamento e faça instalação. É bem simples.

Podem ser instalados via rede ou através de um arquivo gravado no disco.

Aconselha-se remover os módulos que não serão utilizados.

Você pode também gerenciar os módulos de modo a permitir que alguns usuários Webmin tenham acesso às configurações de certos módulos.

Cuidado, pois este recurso permite que um usuário de sua confiança administre um serviço de rede de seu servidor Unix.

12.6 Limites e controle de acesso

Controle o acesso ao seu Webmin, limite os endereços de rede que podem utilizar o seu Webmin.

Habilite o menor número possível de máquinas cliente que podem acessar o seu Webmin.

Controle o número de logins permitidos, tempos de conexão, número de falhas de logins, tipos de autenticação, tempo de logout, estabeleça senha para todos usuários que utilizarão o Webmin.

Usuários administrando módulos do Webmin ajudam muito, mas se você é o administrador principal do sistema sua equipe deve estar sintonizada com os procedimentos que estão sendo adotados.

Consulte sempre os logs da utilização do Webmin. Procure por anomalias e ocorrências suspeitas.

Procure manter o Webmin atualizado e, se encontrar falhas, entre em contato com os autores do Webmin, envie suas questões, opiniões, etc.

Geralmente cada módulo tem um responsável. Entre em contato com o autor se precisar de detalhes, esclarescer dúvidas, etc.

12.7 Configuração dos módulos

Pratique começando pelos serviços mais simples, faça as configurações e veja os resultados.

Todo módulo tem opções para confirmar a configuração, controlar serviços de rede (parar, iniciar, recaregar, reinicializar, etc).

Os módulos são associados aos serviços do servidor Unix. Se não há um certo serviço instalado, o módulo correspondente não tem função nenhuma e não vai ser usado. Alguns módulos dependem de que tenha mais de um serviço de rede instalado. As mensagens sobre essa dependência é mostrada quando se vai trabalhar com estes módulos.

Na falta destes serviços de rede, o Webmin sempre avisa que falta o serviço ou as configurações do módulo estão incorretas.

Os módulos que podem ser configurados são muitos, aumenta a cada mudança de versão do Webmin.

A seguir, uma breve descrição de cada módulo componente do Webmin

12.7.1 Configuração do Sistema Operacional

Aqui são configurados diversos recursos do Sistema Operacional Unix BootUp and Shutdown

Especifica os serviços a serem executados ou não na inicialização do servidor. Pode-se editar os scripts de cada serviço de forma a alterar suas características, parâmetros de configuração, etc. É apresentada uma breve descrição de cada serviço e há a possibilidade de se criar novos serviços.

Disk Quotas

Administra o espaço em disco para cada usuário e grupos do Sistema.

Disk e NFS

Permite que se manipule os diversos tipos de filesystems (montar, desmontar, adicionar e configurar) mostrando seus detalhes e opções avançadas.

Páginas de Manual

Sistema de busca de manuais dos comandos do Unix.

NFS Export

Permite configurar o Unix para trabalhar com compartilhamento de arquivos em NFS (Network File System). Apresenta itens relativos a detalhes de configuração e segurança.

Running Process

Mostra os processos em execução. Podem ser visualizados como lista classificada de PID, User, Memory e CPU. Apresenta também um sistema de busca relacionando o proprietário, filesystem, arquivo em uso, etc. Permite também a execução de comandos em Background e Foreground.

Scheduled Cron Jobs

Permite a administração e configuração de tarefas pré-programadas. Apresenta os detalhes de cada Job, comandos necessários e os intervalos de tempo e periodicidade para execução. Também permite que se configure os usuários com permissão a criar e executar tarefas pré-programadas (cron jobs).

Software Packages

Administração dos pacotes instalados (RPM, TGZ, etc), arquivos de cada pacote, localização, atributos, instalação de novos pacotes, etc. Apresenta um sistema de busca, identificação de arquivos e instalação de pacotes a partir de diversas fontes (arquivos, FTP e HTTP).

Configuração do Init System V

Configuração dos arquivos inittab, que especificam os processos que serão inicializados durante o boot e durante a operação normal. Em geral estão localizados nos diretórios /etc/init.d/boot, /etc/init.d/rc, /etc/rc.d/init.d, gettys, etc.

Logs do Sistema

Permite que sejam configurados os conteúdos dos "arquivos de log". Cada arquivo de log terá um destino. Estes arquivos de log podem conter vários tipos de mensagens (cron, daemon, kernel, lpr, mail, etc). Novos arquivos de log podem ser criados.

Usuários e Grupos

Configuração de usuários e grupos de usuários. Gerenciamento dos dados de cada usuário e grupo do sistema.

12.7.2 Configuração dos Servidores

Aqui são configurados os servidores do Sistema Operacional Unix.

Servidor Apache

Configura um dos mais populares servidores de WEB.

Na configuração Global é permitido se configurar os módulos: Processes and Limites, Networking and Addresses, Apache Modules,MIME Types, Miscellaneous, CGI Programs, Per-Directory Options File, Re-Configure Known Modules, Edit Defined Parameters.

Na configuração Virtual Server é permitido se criar Domínios Virtuais. As opções são: Processes and Limits, Networking and Addresses, Log Files, Document Options, MIME Types, Error Handling, User and Group, Aliases and Redirects, CGI Programs, Directory Indexing, Proxying, Show Directives.

As Opções por Diretório são: Directory / , Directory /home/httpd/html, Directory /home/httpd/cgi-bin.

Bind 8 DNS Server

Configura o Servidor DNS. Este é um dos serviços essenciais para um servidor funcionar adequadamente na Internet. Toda configuração de DNS envolve muitos detalhes relacionados a nomes, domínios e números IPs.

Neste serviço tem-se as opções: Global Server Options e Zona já existente. Na opção Global, as opções são: Other DNS Servers, Logging and Errors, Access Control Lists, Files and Directories, Forwarding and Transfers, Addresses and Topology, Miscellaneous Options, Zone Defaults.

Para a manipulação de Zonas, as opções são: Create a new master zone, Create a new slave zone, Create a new stub zone, Create a new forward zone.

Outra função importante é a edição das Zonas Masters. Podem ser editados os diversos parâmetros de cada zona: Address, Name Server, Name Alias, Mail Server, Host Information, Text, Well Known Service, Responsible Person, Edit Records File, Edit Zone Parameters, Edit Zone Options.

Servidor DHCP - Dynamic Host Configuration Protocol

Configura o Serviço de alocação dinâmica de IP.

Pode-se criar:

Novas redes - Faixa de IPs, Netmask, Servidor de Boot, etc.

Redes compartilhadas - Nome da rede, Servidor de Boot, etc.

Hosts - Hostname, Hardware Address, IP fixo, etc.

Grupos de Hosts - Hosts do grupo, Servidor de Boot, etc.

Permite também a edição das opções globais, aplicadas a todas as redes, redes compartilhadas, hosts e

grupos de hosts.

Possibilita a visualização dos IPs ativos

FTP Server

Permite a configuração do Servidor FTP instalado. Controla os usuários, mensagens, limites e controle de acesso, networking, logs de conexão, aliases e paths, FTP Anônimo, Permissões e Miscelâneas.

Protocolos e Serviços de Internet

Permite a configuração e criação de serviços TCP e UDP.

São configurados os detalhes de cada serviço e do programa servidor: porta usada, Protocolos (TCP, UDP, EGP, etc.) programas (interno ao inetd) ou não, internos ao TCP WRAPPER, seus argumentos, modo de execução, etc.

Pode-se estabelecer novos serviços Internet, configurando todos os parâmetros necessários e personalizando o sistema Servidor.

Por exemplo, redefinir a porta de conexão TCP do serviço FTP, TELNET, etc.

Há também a possibilidade de se trabalhar com o RPC (Remote Procedure Call). É permitido criar novos serviços, editando os programas e configurando seus parâmetros.

Lista Mojordomo

Configuração de Listas Majordomo

Banco de Dados MySQL

Permite se trabalhar com o banco de dados MySQL.

Neste módulo é permitido: Criar e editar banco de dados, configurar as opções de Permissão de usuários, Permissões de banco de dados, Permissões de Hosts, Permissões de Tabelas e de Campos.

Na edição de banco de dados são permitidos a criação de tabelas, configuração dos campos, consultas SQL, alteração dos parâmetros e conteúdos.

Usuários e senhas PPP

Permite a configuração de serviço PPP (contas Dial-in and Dial-out). São permitidos a criação de usuários PPP, especificando: Username, password, servidor e Endereços IP válidos.

O controle de usuários PPP pode ser dependente do controle de usuários do Sistema Unix.

Configuração PostFix, QMail e Sendmail

Aplicativo amplamente difundido na Internet para processamento de correio eletrônico. Estima-se que seja responsável por mais de 75% do tráfego e email na Internet;

Apresenta interação com o serviço de DNS através do registro MX. A configuração apresenta muitos detalhes.

Configuração do Servidor de E-Mail PostFix e QMail. Estes servidores são alternativas ao Sendmail.

Muitas distribuições Linux já trazem o PostFix como servidor de E-Mail padrão da instalação.

Configuração do servidor SAMBA

Permite o compartilhamento de arquivos e Impressoras entre sistemas MS Windows e Unix.

As configurações locais apresentam os compartilhamentos existentes e permitem que sejam feitos novos compartilhamentos e acompanhamento das conexões ao servidor.

Nos compartilhamentos de discos locais, muitos parâmetros podem ser configurados: Nome do compartilhamento, Diretório inicial, disponibilidade e comentários.

Também há a possibilidade de se configurar os itens relativos a Segurança e controle de acesso, Permissões dos arquivos, Nomes de arquivos e outras opções.

Nos compartilhamentos de Impressoras, os parâmetros podem ser configurados:

Nome do compartilhamento, nome da Impressora Unix, Diretório Spool, disponibilidade e comentários.

Há também as opções de Segurança e Controle de Acesso e Opções de Impressora. As Configurações Globais envolvem:

Unix Networking, Windows Networking, Authentications, Windows to Unix Printing, Miscellaneous

Options, File Share Defaults, Printer Share Defaults e SWAT.

Também é permitido trabalhar com as senhas nos seguintes modos: Editar os arquivos de senhas e usuários SAMBA. Converter usuários Unix para usuários SAMBA. Configura a sincronização automática de usuários Unix e SAMBA.

Toda a configuração efetuada pelo SWAT (específico para o SAMBA), Linuxconf e outros softwares podem ser replicadas no WEBMIN.

Proxy SQUID

Servidor de Proxy para armzenamento Cache de WEB. Permite armazenar cópias de páginas WEB, permitindo o acesso mais rápido pelos usuários da rede local, consumindo menor largura de banda. Evita que as páginas sejam carregadas novamente da fonte original.

Outro benefício é o controle de acesso de: Serviços, Sites, Usuários, Redes, Tipos de conteúdo de dados, etc.

Pode operar em modo transparente, modo de Proxy tradicional e Proxy reverso (acelerador de HTTPD).

12.7.3 Configuração do Hardware

Aqui são configurados os elementos de Hardware do Sistema Unix

Linux Bootup Configuration

Configurações das opções de Boot Kernel:

Nome do Kernel para o boot, Opções de Kernel, dispositivo Root, Arquivo inicia RAMDISK, modo de montagem do root, etc.

Permite também a criação de novos Boots de Kernel e novas partições de boot.

Linux RAID

Configuração do hardware para recursos RAID (Redundant Array of Inexpensive Disks).

É um método onde as informações estão espalhadas pelos Hard-Disks. Diversas técnicas são empregadas: Concatenated, Striped, Mirror, Parity e Redundant.

É usado para a recuperação rápida de Hard Disks. É um sistema **mirror/espelhamento** de Hard Disks.

Network Configuration

Permite as configurações relacionadas a Interfaces de Rede e endereçamento.

Network Interfaces : As placas de rede são configuradas de forma controlada :ativadas no boot ou não, interface virtual e Loopback.

Routing and Gateways : Roteador default, Placa com conexão ao Roteador, Atuar como Roteador, Rotas Estáticas e Rotas Locais.

DNS Client : Especifica a ordem de como a resolução de nomes de domínios será pesquisada. Através de HOSTS, DNS, NIS, NIS+ e DB.

São especificados também os servidores de DNS e os domínios para procura na resolução de nomes.

Host Addresses : Especifica o nome do host e seu endereço IP. Esses dados são guardados no arquivos /etc/hosts.

Partitions on Local Disks

É um Partition Manager que permite a manipulação das partições existentes nos discos locais e a criação de novas partições (Primárias e Extendidas).

Permite também que se manipule as partições dos discos, tais como Montagem no Boot, Checagem do Filesystem no boot, modos de checagem, permissões de usuários, etc.

Aqui há um link para Process Manager (da seção System - Running Processes, do Webmin), que disponibiliza um sistema de busca para a pesquisa dos processos que estão sendo executados. São mostrados: PID, Dono do processo, uso da CPU e os comandos envolvidos e todas as informações de cada processo.

Printer Administration

Permite que sejam configuradas impressoras locais e remotas. É possível configurar impressoras Paralelas e Seriais.

No caso de haver a necessidade de Drivers de Impressora, será apresentado um menú com os drivers suportados pelo Kernel de seu Sistema Operacional instalado.

O **Linux Red Hat** e outras distribuições de Linux têm essas facilidades.

System Time

Ferramenta para estabelecer o Sistema de Tempo e Tempo do Hardware da máquina. O UNIX tem total dependência dessas informações.

Pode-se configurar também um Servidor de Tempo para Sincronizar o relógio de seu Unix. O mesmo que estabelecer o servidor NTP.

12.7.4 Outros

Dependendo da versão do Webmin, diferentes módulos são adicionados. Nesta sessão os módulos são para uso geral e não são feitos para configurar servidores.

São módulos para executar tarefas secundárias, tais como consultas, File Manager, Download e Upload, monitorar servidores, instalação de novos módulos do Perl, etc.

Comandos Customizados

Permite que sejam obtidos resultados da execução de comandos comumente executados.

Os parâmetros podem ser substituidos por variáveis.

Pode-se fazer scripts executáveis a partir de seu Browser. A página principal mostra um botão para cada comando definido, com parâmetros opcionais.

Os camandos podem ser editados e modificados.

File Manager

Um sistema de File Manager feito em Java. Permite que sejam manipulados arquivos e diretórios do seu Sistema UNIX.

São permitidos os seguintes recursos: Apagar, Editar, Copiar e Criar arquivos e Diretórios. Recursos de Edição: Copy, Cut e Paste.

Upload de arquivos, manipulação de compartilhamentos (NFS e Windows) e Renomear Folders.

Telnet / SSH Login

Applet Java desenvolvido por Matthias L. Jugel & Marcus Meißner. Está sob a licença GPL (GNU Public License).

Permite que se faça uma conexão Telnet ou SSH com o servidor. O módulo de configuração define os tipos de conexão e os hosts servidores.

Upload / Download

Permite que seja feito Upload de arquivos para o servidor e Download de arquivos do servidor.

12.7.5 Clusters

Permite que se configure diversos servidores para trabalhar em conjunto, atuando como um Cluster.

Cluster Software Packages

Permite manipular os softwares instalados em múltiplos servidores do cluster através de uma interface de gerenciamento.

Cluster Users and Groups

Usuários e grupos que vão utilizar o Cluster.

Cluster Webmin Servers

Configuração dos Webmin dos servidores

Configuration Engine

Ferramenta que permite definir uma série de ações do sistema administrativo para serem usadas quando necessário de forma a permitir tarefas como criar links, diretórios, ajustar permissões e manipular os processos.

Heartbeat Monitor

Permite a monitoração do Cluster Linux.

Capítulo 13

Exercícios

1 - Servidor NFS:
http://br.tldp.org/documentos/comofazer/html/nfs.howto/nfs.howto.pt_BR.html#toc634
2 - Servidor Proftpd:
http://www.lbs.com.br/artigos/proftpd/
3 - Servidor SAMBA como PDC (Primary Domain Controler):
http://ncg.unisinos.br/robotica/samba/
4 - Integração com MS-Windows via SAMBA:
http://ncg.unisinos.br/robotica/samba/
5 - Nmap e chkrootkit:
http://obelix.umh.es/pub/mirrors/LinuxFocus/Portugues/July2001/article170.shtml http://www.chkrootkit.org
6 - Servidor de E-Mail Sendmail:
http://www.sendmail.org
7 - Cliente de E-Mail POP3/IMAP e Webmail:
http://www.squirrelmail.org
8 - Lista de discussão com Mailman:
http://www.conectiva.com/doc/livros/online/6.0/guia_pratico/c1624.html
9 - Ferramentas para DNS (host, dig, nslookup e traceroute):
http://www.laespiral.org/documentacion/articulos/bind/bind18052000.txt
10 - Ferramenta SATAN para scan de rede:
http://www.fish.com/satan/ http://www.cert-rs.tche.br/docs_html/satan.html
11 - Ferramenta Snort para scan de rede:
http://www.conectiva.com/doc/livros/online/7.0/servidor/instalacao-snort.html
12 - Integração de Procmail com Sendmail:
http://www.ppgia.pucpr.br/~maziero/seguranca/blockmail/
13 - Apache com modulo SSL:
http://gul.ime.usp.br/Docs/docs/comofazer/html/Apache+SSL+PHP+fp/Apache+SSL+PHP+fp.pt_BR.html
14 - Utilização do OpenSSH:
http://www.conectiva.com/doc/livros/online/7.0/servidor/implementa-ssh.html
15 - Servidor Proxy Squid:
http://www.revistadolinux.com.br/ed/008/squid.php3 http://squid-docs.sourceforge.net/latest/html/book1.html

A página que mostra estes exercícios está no seguinte URL:

ftp://rubi.cirp.usp.br/pub/Cursos/index.html

Capítulo 14

Referências Bibliográficas

1. Segurança contra Hackers Linux - Brian Hatch, James Lee e George Kurtz
   Editora Futura - 2003
2. Manual de Administração do Sistema UNIX - Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein - Editora Bookman - 3.ed. - 2002
3. Red Hat Linux Security and Optimization - Mohammed J. Kabir - Wiley Publishing - 2002
4. Linux System Security - The Administrator's Guide to Open Source Security Tools - Scott Mann and Ellen L. Mitchell - Prentice Hall PTR - 2000
5. Segurança Máxima - O guia de um hacker para proteger seu site na Internet e sua rede - Autor anônimo - Editora Campus - 2000
6. Anais dos SSI 2001, SSI 2002 e SSI 2003 - CTA - São José dos Campos - SP
7. FreeBSD 4.8 - Manual do Usuário - SAMABSD
8. Aprenda Unix em 24 Horas - Dave Taylor, James C. Armstrong , Jr. Editora Campus

Documentação do Linux Debian e FreeBSD:

- Referências Debian:
  http://www.linorg.cirp.usp.br/Debian.refs/
  http://www.linorg.cirp.usp.br/Debian.refs/Foca.Linux/Avancado/index.html/ch-log.html
- Documentação Debian:
  http://www.debian.org/doc/
- Manuais online:
  http://www.linorg.cirp.usp.br/dwww/
- Unix FreeBSD:
  http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/
  http://www.linorg.cirp.usp.br/Manual.FreeBSD/

Revistas:

- Evidência Digital:
  http://www.guiatecnico.com.br/EvidenciaDigital
- Gazette Linux:
  http://www.gazettelinux.com

- Linux Journal:
  http://www.linuxjournal.com
- Linux Documentation Project:
  http://www.tldp.org http://www.linorg.usp.br/LDP/
- Livros diversos: http://www.linorg.cirp.usp.br/livros.html
- Linux Debian (Manuais de Instalação):
  http://www.debian.org/releases/stable/installmanual
- Instalação do Debian:
  http://www.debian.org/releases/stable/i386/install.pt.html
- Guias de referência e manuais:
  http://www.debian.org/doc/ http://www.debian.org/doc/user-manuals.pt.html
- Guias de usuários:
  http://www.debian.org/doc/manuals/users-guide/users-guide.en.html
- Debian GNU/Linux System Administrator's Manual:
  http://www.debian.org/doc/manuals/systemadministrator/