

Guia de Estudo LPIC – 102

por Luciano Antonio Siqueira

Lançado sob os termos da
Gnu Free Documentation License

Índice

Introdução.....	5
Porque este documento foi escrito?.....	5
À Quem se Destina.....	5
Versão Atualizada do Guia.....	5
Contribuições.....	5
Informações de Copyright.....	5
Pré-requisitos.....	6
Convenções usadas neste documento.....	6
Tópico 105: Kernel.....	7
Objetivo 1.105.1: Manipular/Investigar o Kernel e seus Módulos em Execução.....	7
Kernel.....	7
Módulos do Kernel.....	7
Objetivo 1.105.2: Reconfigurar, Compilar e Instalar um Kernel e Módulos Personalizados.....	9
Configuração.....	9
Compilação e Instalação.....	11
Tópico 106: Carregamento (Boot), Início e Término do Sistema e Runlevels.....	13
Objetivo 1.106.1: Carregamento (Boot) do Sistema.....	13
Objetivo 1.106.2: Alternando Runlevels, Desligando e Reiniciando o Sistema.....	14
Tópico 107: Impressão.....	17
Objetivo 1.107.2: Instalar Impressoras e Filas de Impressão.....	17
Administração de Impressoras pela Linha de Comando.....	17
Arquivos de Configuração do CUPS.....	19
Filas de Impressão.....	19
/etc/printcap.....	20
Objetivo 1.107.3: Imprimindo Arquivos.....	20
Objetivo 1.107.4: Impressoras Locais e Remotas.....	20
Impressão Remota.....	20
Filtros de Impressão.....	21
Tópico 108: Documentação.....	23
Objetivo 1.108.1: Usar e Administrar o Sistema Local de Documentação.....	23
Páginas Manual.....	23
Imprimindo Páginas de Manual.....	24
Páginas Info.....	24
Outras Documentações.....	24
Objetivo 1.108.2: Encontrar Documentação Linux na Internet.....	24
Objetivo 1.108.5: Informar o Usuário sobre Questões Pertinentes ao Sistema.....	25
Tópico 109: Shells, Script, Programação e Compilação.....	27
Objetivo 1.109.1: Personalizar e Trabalhar no Ambiente Shell.....	27
Variáveis.....	27
Funções.....	27
Arquivos de configuração do bash.....	28
Objetivo 1.109.3: Editar e Escrever Scripts Simples.....	29
Variáveis Especiais.....	29
if then else.....	29
Substituição de Comandos.....	31
Instruções de Laço.....	32
Local, Propriedade e Permissão.....	33
Scripts para Tarefas Administrativas.....	33
Tópico 111: Tarefas Administrativas.....	35
Objetivo 1.111.1: Administrar Contas de Usuário, Grupos e Arquivos de Sistema Relacionados.....	35
Conta de Usuário.....	35
Grupos de Usuários.....	38

Objetivo 1.111.2: Ajustar o Ambiente do Usuário e Variáveis de Ambiente de Sistema.....	40
Objetivo 1.111.3: Configurar e Recorrer a Arquivos de log para Corresponder às Necessidades Administrativas e de Segurança.....	40
Objetivo 1.111.4: Automatizar Tarefas Administrativas de Sistema Agendando Trabalhos para Execução Futura.....	42
at.....	42
cron.....	42
Objetivo 1.111.5: Manter uma Estratégia Eficiente de Backup de Dados.....	43
Objetivo 1.111.6: Manutenção de Data e Hora do Sistema.....	44
Relógios e Fuso Horário.....	44
NTP – Network Time Protocol.....	45
Tópico 112: Fundamentos de Redes.....	47
Objetivo 1.112.1: Fundamentos de TCP/IP.....	47
Endereço IP.....	47
Endereço de rede, máscara de rede e endereço broadcast.....	47
Classes de Redes.....	48
Subredes.....	48
IPv4 e IPv6.....	50
Protocolos de rede.....	50
Portas TCP e UDP.....	51
Rotas de Endereços.....	52
Objetivo 1.112.3: Configuração e Resolução de Problemas de Redes TCP/IP.....	54
Arquivos de configuração.....	54
Configuração da interface.....	55
Configuração de Rotas.....	55
Clientes DHCP.....	56
Comandos de Configuração e Inspeção.....	56
Objetivo 1.112.4: Configurar o Linux como um Cliente PPP.....	58
Tópico 113: Serviços de Rede.....	61
Objetivo 1.113.1: Configurar e Administrar o inetd, xinetd e Serviços Relacionados.....	61
O daemon inetd.....	61
O daemon xinetd.....	62
Configuração de Serviços.....	62
Controle de Pedidos.....	63
Objetivo 1.113.2: Operação e Configuração Fundamental de MTA.....	64
sendmail.....	64
Objetivo 1.113.3: Operação e Configuração Fundamental do Apache.....	65
Objetivo 1.113.4: Administração Apropriada dos Daemons NFS e SAMBA.....	66
NFS.....	66
SAMBA.....	67
Objetivo 1.113.5: Configurar um Serviço Básico de DNS.....	69
DNS.....	69
Servidor DNS.....	69
Cliente DNS.....	70
Registro de Domínios.....	70
Objetivo 1.113.7: Utilização do Shell Seguro (OpenSSH).....	70
Tópico 114: Segurança.....	73
Objetivo 1.114.1: Tarefas Administrativas de Segurança.....	73
TCP wrappers.....	73
SUID/SGID.....	74
Verificação de pacotes.....	75
Senhas.....	75
Atualização de programas.....	77
Filtragem de Pacotes – iptables.....	78

Exemplo de criação de firewall simples.....	79
Verificando portas abertas no sistema.....	80
Objetivo 1.114.2: Segurança do Host.....	81
syslog.....	81
Sistema de senhas shadow.....	82
Desativando serviços de rede.....	82
Objetivo 1.114.3: Segurança a Nível de Usuário.....	82
Apêndice 1.....	85
Objetivos detalhados para o exame 102.....	85
Exam 102: Detailed Objectives.....	85
Topic 105: Kernel.....	85
Topic 106: Topic 106 Boot, Initialization, Shutdown and Runlevels.....	86
Topic 107: Printing.....	87
Topic 108: Documentation.....	88
Topic 109: Shells, Scripting, Programming and Compiling.....	89
Topic 111: Administrative Tasks.....	90
Topic 112: Networking Fundamentals.....	92
Topic 113: Networking Services.....	94
Topic 114: Security.....	96
Apêndice 2.....	99
GNU Free Documentation License.....	99

Introdução

Porque este documento foi escrito?

Este material foi escrito quando da minha própria preparação para os exames da certificação LPI nível 1. Depois de terminado, considerei que poderia ser útil para outras pessoas que buscam a certificação e sentem falta de material específico em português. O exame para obtenção do certificado é dividido em duas provas, 101 e 102. Este volume é específico para a prova 102 e foi escrito tendo como referência os objetivos detalhados para prova 102 fornecidos pelo próprio LPI. A lista dos objetivos detalhados para o exame 102 pode ser conferida no primeiro apêndice do presente volume. Mais informações sobre o LPI e suas certificações em <http://www.lpi.org/>.

Todo material contido neste guia foi basicamente retirado de HOWTOs, páginas de manual de programas e demais documentos disponíveis através do [Linux Documentation Project](http://www.linuxdoc.org/). Outra importante fonte foi o livro *Linux System Administration 2*, lançado pelo LinuxIT, disponível em <http://savannah.nongnu.org/projects/lpi-manuals/>.

À Quem se Destina

O presente material destina-se à todos que desejam obter a certificação Linux LPI nível 1. No entanto, o guia também poderá ser útil a quem não pretende obter a certificação, mas interessa-se em aprofundar seus conhecimentos em administração de sistemas GNU/Linux.

Versão Atualizada do Guia

Versões atualizadas deste guia podem ser obtidas em <http://lcsqr.byethost15.com/>.

Contribuições

Todos leitores são convidados a contribuir para o guia. Sugestões para aprofundar os tópicos e exercícios para cada objetivo serão muito bem vindos.

Caso identifique informações incorretas, erros de ortografia ou outros equívocos, informe o autor:

<lcsqr_em_yahoo.com.br>

Informações de Copyright

```
Copyright (c) Luciano Antonio Siqueira.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

Leia [The GNU Manifesto](#) se você quiser saber porquê essa licença foi escolhida para esse material.

O guia foi escrito com todo esforço para garantir a confiabilidade das informações contidas. No entanto, as informações aqui contidas são oferecidas sem qualquer garantia, expressa ou implícita. O responsável pelo material aqui apresentado não se responsabiliza por possíveis danos causados ou alegação do gênero em relação à este livro. Tampouco a leitura deste guia é garantia de sucesso na obtenção da certificação LPI nível 1.

Os logotipos, marcas registradas e símbolos usados neste livro são de propriedade de seus respectivos proprietários.

Pré-requisitos

Para melhor utilização deste guia, presume-se que o leitor já esteja familiarizado com o sistema GNU/Linux. Portanto, os assuntos são abordados de maneira direta, com objetivo de serem apenas referência rápida para posterior estudo e exercício mais aprofundados. O material foi escrito e testado num computador rodando *Linux Slackware 10.2*.

Convenções usadas neste documento

Comandos, opções de comandos, caminhos para arquivos/diretórios, saídas de programas e informações entradas ou tiradas de telas de terminal em geral são apresentados com fonte de tamanho fixo:

Exemplo de tabela de rotas mostradas com o comando route:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
200.228.60.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	200.228.60.1	0.0.0.0	UG	0	0	0	eth0

Argumentos de comandos são geralmente mostrados em itálico, significando que devem ser substituídos por valor apropriado:

```
shutdown [opção] horário [mensagem]
```

Também são mostrados em itálico nomes e termos específicos ao tema:

“Projetos GNU geralmente incluem documentações como *FAQ*, *Readme*, *ChangeLog* e *Guia de usuário/administrador*. Podem estar no formato *ASCII*, *HTML*, *LateX* ou *postscript*. Estes arquivos podem ser encontrados em **/usr/share/doc**, em diretórios correspondentes aos programas.”

Termos em negrito são usados quando introduzidos ou muito relevantes para o assunto:

“Uma conta de usuário pode ser apagada com o comando **userdel**. A opção **-r** assegura que o diretório pessoal do usuário também seja apagado.”

Tópico 105: Kernel

Objetivo 1.105.1: Manipular/Investigar o Kernel e seus Módulos em Execução

Peso: 4

Kernel

O *Kernel* é o núcleo do sistema, responsável pela comunicação entre os programas e as instruções de baixo nível (basicamente hardware) do sistema. O comando **uname** é utilizado para retornar informações a respeito do kernel do sistema. Quando usado apenas com a opção **-a** retorna todas as informações sobre o kernel, na seguinte ordem:

1. NOME-DO-KERNEL
2. NODENAME (nome do host)
3. RELEASE-DO-KERNEL
4. VERSÃO-DO-KERNEL
5. MÁQUINA
6. PROCESSADOR
7. PLATAFORMA-DE-HARDWARE
8. SISTEMA-OPERACIONAL

Demais opções do uname:

- i → Plataforma de Hardware
- m → Hardware da máquina
- n → Nome do Host (nodename)
- p → Processador
- o → Sistema Operacional
- r → Release do Kernel
- s → Nome do Kernel
- v → Versão do Kernel

Exemplo de saída do comando uname -a:

```
$ uname -a
Linux slack102 2.6.13 #1 SMP Thu Mar 23 01:57:38 BRT 2006 i686 unknown unknown
GNU/Linux
```

Módulos do Kernel

Funções especializadas do kernel podem existir na forma de módulos externos. Módulos são armazenados em **/lib/modules/x.x.xx/**(onde **x.x.xx** é o release do kernel do sistema) e são carregados e descarregados dinamicamente através dos comandos **modprobe** ou **insmod**. Mesmo o

comando sendo `insmod` ainda aceito, o comando padrão para manejo de módulos nas versões mais recentes do kernel é o `modprobe`.

A diferença entre os dois comandos é que `modprobe` carrega o módulo especificado e os módulos dos quais depende, conforme listados no arquivo `/lib/modules/x.x.xx/modules.dep`, e `insmod` carrega o módulo sem verificar as dependências. O arquivo `modules.dep` armazena as dependências de cada módulo e é gerado ou atualizado pelo comando `depmod -a`. Essa tarefa pode ser realizada manualmente, mas se forem detectados novos módulos durante o carregamento do sistema, `depmod -a` será executado automaticamente.

O comando `lsmod` lista os módulos carregados. Módulos marcados com *(autoclean)* podem ser descarregados com o comando `modprobe -ra` ou `rmmod -a`. Módulos marcados com *(unused)* poderão ser removidos se seus respectivos nomes forem fornecidos como argumento para `modprobe -r` ou `rmmod`. Os demais módulos estão em uso e não é recomendado seu descarregamento.

Exemplo de lista de módulos carregados:

```
# lsmod
Module                Size  Used by
snd_pcm_oss           49696  0
snd_mixer_oss         18560  2 snd_pcm_oss
ehci_hcd              30472  0
sis900                20352  0
ohci_hcd              20356  0
snd_cmipci            32800  1
gameport              13448  1 snd_cmipci
snd_pcm               84100  2 snd_pcm_oss,snd_cmipci
snd_page_alloc        9608  1 snd_pcm
snd_opl3_lib          10240  1 snd_cmipci
snd_timer             22788  2 snd_pcm,snd_opl3_lib
snd_hwdep             8480  1 snd_opl3_lib
snd_mpu401_uart       7424  1 snd_cmipci
snd_rawmidi           21792  1 snd_mpu401_uart
snd_seq_device        7948  2 snd_opl3_lib,snd_rawmidi
snd                   48100  10 snd_pcm_oss,snd_mixer_oss,snd_cmipci,snd_pcm,
snd_opl3_lib,snd_timer,snd_hwdep,snd_mpu401_uart,snd_rawmidi,snd_seq_device
```

O comando `modinfo` retorna a descrição, autor, licença e parâmetros para o módulo solicitado. Tendo como argumento apenas o nome do módulo, retorna todas as informações disponíveis:

Informações do módulo gameport

```
# modinfo gameport
filename:      /lib/modules/2.6.13/kernel/drivers/input/gameport/gameport.ko
author:       Vojtech Pavlik <vojtech@ucw.cz>
description:  Generic gameport layer
license:      GPL
vermagic:     2.6.13 SMP preempt PENTIUMIII gcc-3.3
depends:
```

As opções `-l` e `-p` mostram, respectivamente, as informações de licença e parâmetros aceitáveis para o módulo solicitado.

Licença do módulo sis900:

```
# modinfo -l sis900
GPL
```

Parâmetros aceitáveis pelo módulo sis900:

```
# modinfo -p sis900
multicast_filter_limit:SiS 900/7016 maximum number of filtered multicast
addresses
max_interrupt_work:SiS 900/7016 maximum events handled per interrupt
sis900_debug:SiS 900/7016 bitmapped debugging message level
```

Para modificar os valores padrão de parâmetros, basta indicá-los quando carregando os módulos como `modprobe` ou `insmod`:

```
# modprobe sis900 multicast_filter_limit=valor
```

Mas o mais comum é que os módulos sejam carregados automaticamente pelo sistema. Por isso, os parâmetros podem ser armazenados no arquivo `/etc/modules.conf`. O mesmo parâmetro do exemplo poderia estar em `/etc/modules.conf` na forma:

```
options sis900 multicast_filter_limit=valor
```

Também através do arquivo `/etc/modules.conf` é possível criar *alias*es (cognomes) para módulos, de forma que possam ser invocados usando um nome convencional. Exemplo de alias em `/etc/modules.conf`:

```
alias eth0 sis900
alias sound-slot-0 snd_cmipci
```

Alias e opções também podem ser definidos da mesma forma no arquivo `/etc/modprobe.conf`. Através deste arquivo, também é possível definir as ações para eventos de carregamento e descarregamento de módulos, como a execução de um programa.

Exemplo: Quando solicitado a carregar sis900, carregar tun e só depois sis900:

```
install sis900 modprobe tun; modprobe --ignore-install sis900
```

A opção `--ignore-install` é necessária para que o `modprobe` não execute novamente a linha `install` referente à `sis900` em `/etc/modprobe.conf`. É possível usar o mesmo procedimento quando do descarregamento de módulos:

```
remove sis900 modprobe --ignore-remove -r sis900; modprobe -r tun
```

Dessa vez, a opção `--ignore-remove` aparece para impedir a execução cíclica da linha `remove` referente ao módulo `sis900` em `/etc/modprobe.conf`.

Objetivo 1.105.2: Reconfigurar, Compilar e Instalar um Kernel e Módulos Personalizados.

Peso: 3

Configuração

O local padrão de armazenamento do código fonte do kernel é `/usr/src/linux`, que é um link simbólico para `/usr/src/linux-x.x.xx/` (onde `x.x.xx` é o release do kernel). Existe mais de uma interface para configurar o kernel, e todas elas desempenham a mesma função. As interfaces de configuração são invocadas através do comando `make`:

`make config`

Interface em linha de comando. É feita uma sequência de perguntas (bastante extensa) sobre cada aspecto da configuração.

`make menuconfig`

Interface de menus *ncurses*.

`make xconfig`

Interface gráfica X, utilizando a biblioteca *QT*.

`make gconfig`

Interface gráfica X, utilizando a biblioteca *Gtk*.

`make oldconfig`

Usa as configuração anteriores como ponto de partida para uma nova configuração.

Enquanto que alguns recursos precisam ser compilados como estáticos (embutidos no kernel), a maioria pode ser compilada como dinâmico (módulo). O item marcado com asterisco (*) será compilado como estático e o item marcado com a letra “M” será compilado como módulo. Itens deixados em branco não serão compilados. Espaços de escolha “[]” indicam que o item só poderá ser compilado como estático e espaços de escolha “< >” indicam que o item poderá ser compilado tanto como estático quanto módulo.

A configuração do kernel está dividida nos eixos principais:

Code maturity level options

Mostrar ou não recursos considerados instáveis

General setup

Características gerais do kernel. É possível incluir um termo de versão para o kernel personalizado

Loadable module support

Suporte ao sistema de módulos e define algumas características.

Processor type and features

Indica o tipo de processador que o kernel utilizará e recursos como multiprocessamento.

Power management options (ACPI, APM)

Opções relativas ao controle de energia. Indicado especialmente para laptops

Bus options (PCI, PCMCIA, EISA, MCA, ISA)

Suporte para os diferentes tipos de barramentos

Executable file formats

Tipos de arquivos que o sistema será capaz de executar

Networking

Suporte e configuração dos diferentes tipos de plataformas de rede

Device Drivers

Escolha e configuração dos dispositivos de hardware, integrados e periféricos.

File systems

Lista de sistemas de arquivos compatíveis e recursos relacionados

Kernel hacking

Opções de depuração do kernel

Essas são as principais categorias de configuração, e podem diferenciar de uma versão do kernel para outra.

As configurações são salvas no arquivo `/usr/src/linux/.config`, que será usado para guiar a construção do novo kernel e módulos. No arquivo `Makefile` é possível mudar variáveis como `EXTRAVERSION`, que indica ser uma compilação de kernel personalizado.

Compilação e Instalação

Para assegurar-se de que todo kernel e módulos sejam compilados com a nova configuração e não reaproveitem objetos pré-compilados, é usado o comando:

```
# make clean
```

Agora já podem ser compilados kernel e módulos:

make bzImage ou **make zImage**

Compila o novo kernel

make modules

Compila os novos módulos

make all

Compila o novo kernel e os novos módulos

make modules_install

Instala os novos módulos em `/lib/modules/x.x.xx/` (`x.x.xx` é a versão do novo kernel). O comando **depmod -a** será automaticamente executado para criar o arquivo `modules.dep` de dependências dos módulos.

O comando `make` também pode criar pacotes de instalação do kernel específicos, com `make rpm-pkg`, `make binrpm-pkg` e `make deb-pkg`.

Num computador padrão i386 (o mais comum), o novo kernel estará em `/usr/src/linux/arch/i386/boot/bzImage`. Este arquivo deve ser copiado (ou feito um link) para `/boot`, e o arquivo de configuração do carregador de boot (`/etc/lilo.conf` ou `/boot/grub/grub.conf`) deve ser atualizado para apontar para o novo kernel. No caso do *lilo*, é importante reinstalá-lo com o comando `lilo`.

Novas funcionalidades podem exigir configurações suplementares, como passar novas opções para módulos. Para funções compiladas estaticamente, opções podem ser passadas pela instrução `append`, no arquivo `/etc/lilo.conf`.

Com o novo kernel instalado, basta reiniciar o sistema para passar a utilizá-lo.

Tópico 106: Carregamento (Boot), Início e Término do Sistema e Runlevels

Objetivo 1.106.1: Carregamento (Boot) do Sistema

Peso: 3

Antes de carregar o kernel, o *LILO* apresenta um prompt no qual é possível alterar o comportamento padrão de carregamento do sistema. Esse prompt aparece na forma:

```
boot:
```

Onde é possível indicar o kernel a carregar, passar parâmetros de configuração e alterar o *runlevel* inicial.

Parâmetros são passados no formato *item=valor*. Exemplo de itens:

- `acpi` → liga/Desliga suporte a ACPI
- `init` → Programa para executar no lugar de `/sbin/init`
- `mem` → Montante de memória RAM do sistema
- `root` → Especificar o dispositivo raiz

Outras opções não precisam de parâmetros, por exemplo:

- `noinitrd` → Não carrega o *ramdisk* inicial
- `ro` → Monta o dispositivo como somente leitura

Carregar o kernel linux, indicando /dev/hda3 como dispositivo raiz, sem ramdisk inicial e somente leitura

```
boot: linux root=/dev/hda3 noinitrd ro
```

Dessa mesma forma, é possível passar parâmetros para os módulos compilados estaticamente no kernel. Para que os parâmetros sejam automaticamente passados em todo boot, eles podem ser incluídos na instrução `append` no arquivo `/etc/lilo.conf`. O lilo deverá então ser reinstalado. Para os módulos externos, parâmetros são passados diretamente pelo comando `modprobe`/`insmod` ou podem constar nos arquivos `/etc/modules.conf` ou `/etc/modprobe.conf`.

Outra possibilidade de uso do prompt de boot é alterar o *runlevel* inicial do sistema. Os valores aceitos são `s`, `single`, `S`, `1`, `2`, `3`, `4`, `5`. Exemplo:

Iniciar o sistema em modo de usuário único (Single user)

```
boot: linux S
```

Se nenhum parâmetro for passado, o runlevel inicial será o especificado em `/etc/inittab`.

Para inspecionar o processo de inicialização do sistema, é usado o comando `dmesg`. As mensagens do carregamento são armazenadas em `/var/log/dmesg`, mensagens do kernel podem ser checadas através do arquivo `/var/log/messages`.

Objetivo 1.106.2: Alternando Runlevels, Desligando e Reiniciando o Sistema

Peso: 3

Runlevels (níveis de execução do sistema) são numerados de 0 a 6 e suas funções podem variar de uma distribuição para outra. Via de regra o próprio arquivo `/etc/inittab`, que define os runlevels, traz também informações a respeito de cada um.

Formato de entrada em /etc/inittab:

```
id:runlevels:ação:processo
|           |           |           |
|           |           |           |  --> O comando a ser acionado
|           |           |           |  --> O tipo da ação a ser tomada
|           |           |           |  --> Lista dos runlevels para os quais a
|           |           |           |  ação da entrada deverá ser executada
|           |           |           |
|  --> Nome de até 4 caracteres para identificar
|           |           |           |  a entrada no inittab
```

Os tipos mais comuns para ação são:

`sysinit` → Processo executado durante o boot do sistema

`wait` → O processo será executado e `init` aguardará seu término

`ctrlaltdel` → O processo será executado quando o `init` receber o sinal `SIGINT`, o que significa que as teclas `Ctrl+Alt+Del` foram pressionadas no console.

Trecho exemplo de /etc/inittab:

```
# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S
# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K
# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M
```

Na maioria dos casos, os runlevels são assim definidos:

0 → Desligar o sistema

1 → Usuário único (super usuário)

2 → Multiusuário sem serviços de rede

3 → Multiusuário com serviços de rede

4 → Livre (não definido)

5 → Multiusuário, serviços de rede e login gráfico

6 → Reiniciar o sistema

Os únicos runlevels comuns à toda distribuição Linux são 0, 1 e 6. O runlevel padrão é definido no `/etc/inittab`, na entrada:

```
id:x:initdefault
```

Onde “x” é o número do runlevel padrão ao iniciar o sistema. Esse número jamais pode ser 0 ou 6, por razões óbvias.

O programa que realiza o procedimento de início e troca de um runlevel é `/sbin/init`. O `init` é o primeiro programa disparado após kernel acessar o dispositivo raiz, portanto o *PID* de `init` será sempre 1.

Para alternar de runlevel, usa-se o próprio `init` ou o comando `telinit`, fornecendo como argumento o número do runlevel desejado. O comando `runlevel` mostra dois números, o primeiro mostra o runlevel anterior e o segundo o runlevel atual.

Para desligar ou reiniciar o sistema, o comando `shutdown` oferece mais funcionalidades. Todos os usuários no sistema são notificados e novos *logins* são bloqueados. Todos os processos recebem o sinal `SIGTERM` seguido de `SIGKILL` antes do sistema desligar ou alternar o runlevel. O padrão, caso não sejam usadas as opções `-h` ou `-r` é que o sistema alterne para runlevel 1.

```
shutdown [opção] horário [mensagem]
```

Apenas o argumento horário é obrigatório. Ele indica quando efetuar a ação requisitada e seu formato pode ser:

hh:mm (horário para execução)

+m (minutos restantes)

`now` ou *+0* (imediatamente)

Algumas das opções mais usadas:

`-a` → Usar o arquivo de permissão `/etc/shutdown.allow`

`-r` → Reiniciar a máquina

`-h` → Desligar a máquina

`-t segundos` → Tempo de espera antes do `shutdown` executar a ação solicitada.

O argumento *mensagem* será o aviso enviado a todos usuários no sistema.

Para impedir que qualquer usuário reinicie a máquina apertando `Ctrl+Alt+Del`, a opção `-a` deve constar para o comando `shutdown` referente a ação `ctrlaltdel` em `/etc/inittab` e os usuário liberados para realizar a ação devem constar no arquivo `/etc/shutdown.allow`.

Tópico 107: Impressão

Objetivo 1.107.2: Instalar Impressoras e Filas de Impressão

Peso: 1

O sistema impressão mais comum em Linux é o **CUPS** (*Common Unix Printing System*). O CUPS provê controle sobre impressoras, filas de impressão, impressão remota e compatibilidade com as ferramentas do sistema lpd. A configuração do CUPS pode ser feita através da linha de comando ou através de uma interface WEB, em **http://localhost:631/**. Para configuração e uso do CUPS, é fundamental que o servidor de impressão (daemon **/usr/sbin/cupsd**) esteja ativo. A maioria das distribuições inicia o servidor de impressão no boot do sistema.

Administração de Impressoras pela Linha de Comando

O comando `lpinfo` é usado para obter uma lista dos dispositivos de impressão e modelos de impressoras disponíveis.

Listando dispositivos disponíveis:

```
# lpinfo -v
network socket
network http
network ipp
network lpd
direct canon:/dev/lp0
direct epson:/dev/lp0
direct parallel:/dev/lp0
direct scsi
serial serial:/dev/ttyS0?baud=115200
serial serial:/dev/ttyS1?baud=115200
serial serial:/dev/ttyS2?baud=115200
serial serial:/dev/ttyS3?baud=115200
direct usb:/dev/usb/lp0
direct usb:/dev/usb/lp1
direct usb:/dev/usb/lp2
(...)
```

A primeira palavra da lista identifica o tipo do dispositivo. Para impressoras locais, é importante que o módulo do respectivo dispositivo esteja carregado (porta paralela, usb, etc).

Listando Modelos de Impressoras Disponíveis:

```
# lpinfo -m
(...)
C/pcl-550.ppd.gz HP DeskJet 550C - CUPS+Gimp-Print v4.2.7
C/pcl-560.ppd.gz HP DeskJet 560C - CUPS+Gimp-Print v4.2.7
foomatic-ppds/HP/HP-DeskJet_600-hpijs.ppd.gz HP DeskJet 600 Foomatic/hpijs
(reco
mmended)
C/pcl-601.ppd.gz HP DeskJet 600 series - CUPS+Gimp-Print v4.2.7
C/pcl-600.ppd.gz HP DeskJet 600/600C - CUPS+Gimp-Print v4.2.7
foomatic-ppds/HP/HP-DeskJet_610C-hpijs.ppd.gz HP DeskJet 610C Foomatic/hpijs
(re
commended)
foomatic-ppds/HP/HP-DeskJet_610CL-hpijs.ppd.gz HP DeskJet 610CL Foomatic/hpijs
(
recommended)
foomatic-ppds/HP/HP-DeskJet_612C-hpijs.ppd.gz HP DeskJet 612C Foomatic/hpijs
(re
commended)
(...)
```

A maior parte das tarefas de administração de impressão pode ser realizada com o comando **lpadmin**, que é equivalente ao antigo **lpc**.

Opções comuns de lpadmin:

-c classe

Adiciona a impressora indicada à classe. Se a classe não existir, será criada.

-m modelo

Especifica o driver padrão da impressora, geralmente um arquivo PPD. Os arquivos PPD podem ser encontrados em `/usr/share/cups/model/`. A lista de todos modelos disponíveis é mostrada com o comando `lpinfo -m`.

-r classe

Remove a impressora indicada da classe. A classe será apagada se tornar-se vazia.

-v dispositivo

Indica o endereço do dispositivo de comunicação da impressora.

-D info

Descrição textual para a impressora.

-E

Autoriza a impressora a receber trabalhos.

-L localização

Descrição textual para a localização da impressora

-P arquivo PPD

Especifica um arquivo PPD de driver local para a impressora

Para adicionar uma impressora local, modelo HP DeskJet 600:

```
# lpadmin -p HP_DeskJet_600 -E -v parallel:/dev/lp0 -D "HP DeskJet 600" \
-L "Impressora Local" -m foomatic-ppds/HP/HP-DeskJet_600-hpijs.ppd.gz
```

A impressora foi adicionada com suas opções padrão (tamanho da folha, qualidade de impressão, etc). Para alterar esse valores, usa-se o comando `lpoptions`.

Listar as opções possíveis para a impressora recém instalada:

```
# lpoptions -p HP_DeskJet_600 -l
PageSize/Page Size: Custom Letter *A4 Photo PhotoTearOff 3x5 5x8 A5 A6
A6TearOff B5JIS Env10 EnvC5 EnvC6 EnvDL EnvISOB5 EnvMonarch Executive FLSA
Hagaki Legal Oufuku w558h774 w612h935
PageRegion/PageRegion: Letter A4 Photo PhotoTearOff 3x5 5x8 A5 A6 A6TearOff
B5JIS Env10 EnvC5 EnvC6 EnvDL EnvISOB5 EnvMonarch Executive FLSA Hagaki Legal
Oufuku w558h774 w612h935
PrintoutMode/Printout Mode: Draft Draft.Gray *Normal Normal.Gray High.Gray
Quality/Resolution, Quality, Ink Type, Media Type: *FromPrintoutMode
300ColorCMYK 300DraftColorCMYK 300DraftGrayscaleCMYK 300GrayscaleCMYK
600x300BestGrayscaleCMYK
```

Definir a opção `PrintoutMode` como `Draft`:

```
# lpoptions -p HP_DeskJet_600 -o PrintoutMode=Draft
```

Para remover a impressora:

```
# lpadmin -x HP_DeskJet_600
```

O estado das impressoras e filas pode ser checado com o comando `lpstat -a`. Para ativar a impressora, são utilizados os comandos `/usr/bin/enable` ou `/usr/sbin/accept`.

Arquivos de Configuração do CUPS

Os arquivos de configuração do CUPS encontram-se em `/etc/cups/`:

`classes.conf`

Define as classes para as impressoras locais.

`cupsd.conf`

Configurações do daemon `cupsd`

`mime.convs`

Define os filtros disponíveis para conversão de formatos de arquivos

`mime.types`

Define os tipos de arquivos conhecidos

`printers.conf`

Define as impressoras locais disponíveis

`lpoptions`

Configurações específicas para cada impressora

Filas de Impressão

Fila de impressão é o diretório temporário onde ficam os trabalhos antes de serem impressos. Por

padrão, a fila no sistema `lpd` é `/var/spool/lpd/` e no CUPS é `/var/spool/cups/`.

O comando `lpc` (em desuso) é um utilitário interativo de controle de fila de impressão. Através dele é possível bloquear/liberar filas e impressoras e alterar a seqüência de impressão de trabalhos.

Para apenas listar os trabalhos numa fila de impressão, é usado o comando `lpq`.

/etc/printcap

O arquivo `/etc/printcap` é o principal arquivo de configuração do antigo sistema `lpd` e define as filas de impressão disponíveis no sistema e suas características. Este arquivo é gerado pelo CUPS para manter a compatibilidade com o sistema antecessor. Cada definição de impressora é feita em uma linha e os campos de configuração são delimitados por “:”. O primeiro nome, na primeira coluna do arquivo, indica o nome da impressora. É possível criar *aliases* para a impressora, separados por barra vertical “|”. Os demais campos são parâmetros de controle da impressora.

Exemplo de /etc/printcap:

```
HP_DeskJet_600|HP DeskJet 600:rm=localhost:rp=HP_DeskJet_600:
```

Objetivo 1.107.3: Imprimindo Arquivos

Peso: 1

A maioria dos aplicativos já imprime diretamente no formato compatível para impressão (*PostScript*). Porém, também é possível converter manualmente o arquivo para o formato *postscript*. Um dos programas usados para essa tarefa é o `a2ps`. Dessa forma, é possível ter um controle maior sobre o resultado final da impressão.

O comando `lpr` envia o documento indicado para a fila de impressão. Opções comuns são `-Pxxx`, envia o arquivo para a fila `xxx`, `-#x`, imprime o documentos `x` vezes e `-s`, que não copia o documento para a fila de impressão, mas cria um link simbólico na mesma.

Para listar os trabalhos numa fila de impressão, é usado o comando `lpq`. `lpq -a` mostra os trabalhos em todas as filas do sistema. `lpq -P` mostra os trabalhos no host especificado. A cada trabalho é associado um número. Esse número pode ser usado pelo comando `lprm` para cancelar um trabalho na fila de impressão. O comando `lprm` sem argumentos irá cancelar o último trabalho enviado. Informando o nome de um usuário como argumento cancelará todos os trabalhos de impressão deste usuário na fila. Para cancelar todos os trabalhos, `lprm -a` ou `lprm -`.

Objetivo 1.107.4: Impressoras Locais e Remotas

Peso: 1

Impressão Remota

Impressoras instaladas e configuradas na máquina local poderão estar disponíveis para outras máquinas na rede automaticamente. Para tanto, basta especificar no host da impressora a rede da qual aceitar pedidos e no(s) terminal(is) , o host da impressora.

Supondo ser o host da impressora `slack102` e a rede na qual compartilhar a impressora `192.168.1.0/24`, a configuração pode ser feita da seguinte forma:

No host da impressora, no arquivo `/etc/cups/cupsd.conf`

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1

Allow From 192.168.1.0/24 # Liberar para a rede 192.168.1.0/24

</Location>
```

No(s) terminal(is), no arquivo `/etc/cups/client.conf`

```
ServerName slack102 # host da impressora
```

É necessário reiniciar o daemon `cupsd` para utilizar as novas configurações. A impressora remota pode ser verificada no terminal com o comando `lpstat -a`:

```
# lpstat -a
HP_DeskJet_600 accepting requests since Jan 01 00:00
```

Para que os documentos sejam automaticamente impressos nessa impressora, esta é tornada padrão com o comando:

```
lpoptions -d HP_DeskJet_600
```

Através do sistema SAMBA, é possível imprimir com uma impressora num host MS-Windows®. Para fazê-lo, ao adicionar uma impressora o dispositivo escolhido deverá ser “*Windows Printer via SAMBA*” e a URI `smb://servidor/impressora`, substituindo pelos valores apropriados. Se o servidor exigir autenticação, a URI deverá ser `smb://usuário:senha@grupo/servidor/impressora`.

Filtros de Impressão

Mesmo que arquivos sejam mandados sem terem sido convertidos para impressão, é possível usar filtros para realizar essa tarefa. A identificação do tipo de arquivos é feita consultando o arquivo `/etc/cups/mime.types`. Identificado o tipo do arquivo, a ferramenta de conversão adequada é indicada através do arquivo `/etc/cups/mime.convs`, tornando-o apto para conversão.

Tópico 108: Documentação

Objetivo 1.108.1: Usar e Administrar o Sistema Local de Documentação

Peso: 4

Páginas Manual

Praticamente todos os comandos e arquivos de configuração no Linux têm uma página manual, que demonstra de maneira detalhada as funções e como usar o item em questão. Para ver uma página manual, basta usar o comando **man** tendo o comando/arquivo como argumento.

Em sua maioria, as páginas manual têm a seguinte organização:

Nome

Assunto da página manual seguido por uma descrição breve.

Sinopse

A sintaxe do comando

Descrição

Descrição detalhada

Opções

Revisão de todas as opções e suas funções

Arquivos

Arquivos relacionados ao assunto

Veja também

Outras páginas de manual relacionadas ao tópico

É possível buscar por ocorrências de um termo na seção nome das páginas manual através do comando **apropos**, retornando a descrição breve para cada ocorrência:

```
$ apropos pop3
Net::POP3          (3) - Post Office Protocol 3 Client class (RFC1939)
getmail            (1) - retrieve mail from one or more POP3 accounts
popa3d             (8) - Post Office Protocol (POP3) server
```

O banco de dados do comando **whatis** armazena a seção *nome* das páginas manual do sistema. O banco de dados geralmente é atualizado por um agendamento *cron*. Um recurso fornecido como argumento para **whatis** retorna a descrição breve para o mesmo:

```
$ whatis man
man                (1) - format and display the on-line manual pages
man                (7) - macros to format man pages
man.conf [man]    (5) - configuration data for man
```

Os números entre parênteses referem-se à seção a qual pertence a página manual.

Definição das seções:

Seção 1 → Programas disponíveis ao usuário

Seção 2 → Rotinas de Sistema Unix e C

Seção 3 → Rotinas de bibliotecas C

Seção 4 → Arquivos especiais (dispositivos em /dev)

Seção 5 → Convenções e formatos de arquivos

Seção 6 → Jogos

Seção 7 → Diversos (macros textuais, etc)

Seção 8 → Procedimentos Administrativos (daemons, etc)

Para acessar um item em uma seção específica, o número da seção precede o nome do item.

Acessar o manual do recurso man na seção 7:

```
$ man 7 man
```

Por padrão, as páginas manual geralmente são armazenadas em `/usr/man` e `/usr/share/man`, em subdiretórios correspondentes à seção. Outros locais podem ser especificados através da variável `MANPATH`, definida no arquivo de configuração do `man` `/usr/lib/man.conf` ou `/etc/man.conf`.

Imprimindo Páginas de Manual

Páginas manual podem ser impressas como texto sem formatação direcionando a saída do comando `man` para um arquivo ou comando de impressão.

Para não perder a formatação, a página pode ser convertida usando o comando `groff`:

Gravando o manual de find para um arquivo postscript:

```
$ zcat /usr/man/man1/find.1.gz | groff -man -Tps > find.ps
```

Para imprimir diretamente:

```
$ zcat /usr/man/man1/find.1.gz | groff -man -Tps | lpr
```

Páginas Info

Formato alternativo ao das páginas manual. São acessíveis através do comando `info`. Em geral, informações disponíveis em páginas *info* também estão disponíveis em páginas manual, porém de forma menos detalhada.

Por padrão são armazenadas em `/usr/share/info`.

Outras Documentações

Projetos GNU geralmente incluem documentações como *FAQ*, *Readme*, *ChangeLog* e *Guia de usuário/administrador*. Podem estar no formato *ASCII*, *HTML*, *LateX* ou *postscript*. Estes arquivos podem ser encontrados em `/usr/share/doc`, em diretórios correspondentes aos programas.

Objetivo 1.108.2: Encontrar Documentação Linux na Internet

Peso: 3

O **LDP** – *Linux Documentation Project* - <http://www.tldp.org> – fornece documentos detalhados sobre muitas tarefas. Os documentos estão organizados na forma de HOWTOs (receitas passo-a-passo) e guias, que abordam temas de forma mais abrangente e aprofundada. Há também links para artigos sazonais.

Respostas para perguntas recorrentes podem ser encontradas e novos questionamentos podem ser feitos em sites de fóruns e em newsgroups. Um site com fóruns diversos e voltados para tópicos específicos é <http://www.linuxquestions.org/>. Newsgroups podem ser acessados através do Google Groups <http://groups.google.com/>. Os grupos tradicionais para discussão de Linux são os derivados do **comp.os.linux.***.

Como cada distribuição tem suas particularidades, os respectivos sites fornecem documentação de grande valia, sobre administração, configuração e uso de cada aspecto do sistema. Também os sites dos desenvolvedores de programas costumam fornecer documentação extra. Alguns sites de distribuições populares:

<http://www.slackware.com/>

<http://www.debian.org/>

<http://www.redhat.com/>

Objetivo 1.108.5: Informar o Usuário sobre Questões Pertinentes ao Sistema

Peso: 1

Informações úteis podem ser mostradas durante o login do usuário. Contato do administrador, informes de manutenção, etc.

O programa de controle de terminal `agetty` mostra o conteúdo do arquivo `/etc/issue` no prompt de login. Para sessões `telnet`, o arquivo `/etc/issue.net` desempenha a mesma função.

Para exibir uma mensagem após um login bem sucedido, utiliza-se do arquivo `/etc/motd` (message of the day).

Tópico 109: Shells, Script, Programação e Compilação

Objetivo 1.109.1: Personalizar e Trabalhar no Ambiente Shell

Peso: 5

O shell é ao mesmo tempo uma interface de interação com o computador e um ambiente de programação. Há diferentes opções de shell, como o *bash*, *csh* ou *zsh*. O mais utilizado, devido a suas muitas qualidades, e que será aqui abordado é o *bash*. Há quem prefira o *csh*, que possui sintaxe semelhante a da linguagem C.

Variáveis

Variáveis são utilizadas para fornecer informações úteis e necessárias à programas e usuários. São definidas na forma `nome=valor`. Não deve haver espaços na definição. Variáveis podem ser globais ou locais.

Variáveis globais são aquelas acessíveis à todos os processos. Exemplos de variáveis globais são `PATH`, que define os diretórios de programas; `HOME`, o diretório pessoal do usuário; `SHELL`, o shell padrão para o usuário. Por conveniência, os nomes de variáveis globais são em maiúsculas. As variáveis globais podem ser listadas com o comando `env`. Todas as variáveis são listadas usando `set`.

As variáveis globais são definidas no login, para todo o sistema através do arquivo `/etc/profile` ou por usuário, através do arquivo `~/.bash_profile`.

Variáveis locais são acessíveis apenas à sessão atual do shell. Podem ser definidas em scripts ou na própria linha de comando. Para tornar uma variável acessível aos processos filhos subseqüentes, é usada a instrução `export`:

```
$ export BASH='Bourne Again Shell'
```

Os valor da variável é retornado com uso de “\$”:

```
$ echo $BASH
Bourne Again Shell
```

Para excluir uma variável, é usada a instrução `unset`:

```
$ unset BASH
```

Funções

Para simplificar tarefas corriqueiras, é possível escrever funções que aglutinam comandos. Podem ser escritas diretamente pela linha de comando ou serem definidas em scripts ou nos arquivos de configuração do *bash*.

Exemplo: Mostrar algumas informações a respeito do nome de programa fornecido.

Definir a função no prompt, interativamente:

Tópico 109: Shells, Script, Programação e Compilação

```
$ function pinfo () {
> echo "Localização de $1:"
> which $1
> echo "Processos referentes a $1:"
> ps x | grep $1
> }
```

Em uma só linha:

```
$ function pinfo () { echo "Localização de $1:"; \
which $1; echo "Processos referentes a $1:"; ps x | grep $1; }
```

Através do arquivo ~/.bashrc:

```
function pinfo () {
    echo "Localização de $1:"
    which $1
    echo "Processos referentes a $1:"
    ps x | grep $1
}
```

Assim, a função `pinfo` poderá ser utilizada como um comando:

```
$ pinfo soffice
Localização de soffice:
/usr/bin/soffice
Processos referentes a soffice:
 5163 ?          Ss      0:00 /bin/sh /usr/bin/soffice
 5179 ?          Sl      2:20 /opt/openoffice.org2.0/program/soffice.bin
 5735 pts/0     S+      0:00 grep soffice
```

Arquivos de configuração do bash:

É possível automatizar a criação de variáveis, aliases, funções e outras personalizações do bash, tanto para quando o usuário entra no sistema quanto para cada nova instância do bash. Para cada propósito, seja para todos ou por usuário em particular, existe o arquivo apropriado:

`/etc/profile`

Arquivo de configuração para o sistema, executado durante o login. Contém definições de variáveis globais de ambiente, como `PATH`.

`/etc/bashrc`

Arquivo com funções e aliases comuns, para ser chamado pelos `~/.bashrc` e cada usuário.

`~/.bash_profile`

Se o arquivo existir, será executado após `/etc/profile` após o login.

`~/.bash_login`

Se `~/.bash_profile` não existir, este será procurado.

`~/.profile`

Se nem `~/.bash_profile` ou `~/.bash_login` existirem, este será procurado.

`~/.bashrc`

Arquivo executado após o login e em toda sessão, interativa ou não, do bash.

~/`.bash_logout`

Executado no logout do sistema.

~/`.inputrc`

Define as opções de entrada de teclado. Por padrão, combinações estilo *Emacs* são usadas. Para alterar para o estilo do *vi*, adiciona-se a linha `set editing-mode vi`. Para eliminar o aviso de erro sonoro, adiciona-se `set bell-style none`. O arquivo de configuração global é `/etc/inputrc`.

Objetivo 1.109.3: Editar e Escrever Scripts Simples

Peso: 3

Scripts são arquivos que passam instruções a um interpretador. Diferente de programas compilados, scripts são arquivos de texto que podem ser editados em qualquer editor.

A primeira linha de um script deve especificar o interpretador para o script, que é indicado pelos caracteres `#!` (*she-bang*). Para um script com instruções para o `bash`, a primeira linha deverá ser `#!/bin/bash`. Assim, o interpretador padrão será o shell *bash*.

O script deverá ter permissão de execução para rodar diretamente, ou ser invocado como argumento do comando `bash` ou `sh`.

Variáveis Especiais

Os argumentos passados para um script e outras informações úteis são retornados através da variável especial `$x`, onde `x` determina que valor retornar:

`$*` → Todos os valores passados como argumentos

`$#` → O número de argumentos

`$0` → O nome do script

`$n` → O valor do argumento na posição `n`

`$!` → PID do último programa executado

`$$` → PID do shell atual

`$?` → Código de saída do último comando

Para requisitar valores do usuário durante a execução do script, é usada a instrução `read`:

```
echo "Entrar valor solicitado:"
read RESPOSTA
```

O valor retornado será armazenado na variável `RESPOSTA`. Caso uma variável de retorno não seja especificada, o nome padrão da variável de retorno, `REPLY`, será utilizado.

if then else

A estrutura lógica `if` executa um comando ou uma lista de comandos se uma afirmação for verdadeira. A instrução `test` avalia a afirmação e retorna se verdadeira ou falsa. Seu uso é geralmente associado à instrução condicional `if`:

Retornar “ok” se o arquivo `/bin/bash` for executável

```
if test -x /bin/bash ; then
    echo "ok"
fi
```

Maneira mais prática para testar:

```
if [ -x /bin/bash ] ; then
    echo "ok"
fi
```

A instrução `else` é um apêndice à estrutura `if`, e determina o bloco de instruções a executar caso a afirmação avaliada seja falsa. Exemplo:

```
if [ -x /bin/bash ] ; then
    echo "ok"
else
    echo "não ok"
fi
```

O final da estrutura `if` deve ser sempre sinalizado com `fi`.

Uma variação da instrução `if` é **case**. A instrução `case` executará a instrução se um item indicado for encontrado em uma lista de itens divididos pelo caracter “|”. Exemplo:

```
case 3 in (1|2|3|4|5)
    echo "Número 3 encontrado na lista,";
    echo "portanto case finalizou e";
    echo "executou esses comandos";
esac
```

O final da estrutura `case` deve ser sempre sinalizado com `esac`.

Opções de avaliação de `test` para arquivos e diretórios:

-d *caminho*

Verdadeiro se o *caminho* existir e for um diretório

-c *caminho*

Verdadeiro se o *caminho* existir

-f *caminho*

Verdadeiro se o *caminho* existir e for um arquivo comum

-L *caminho*

Verdadeiro se o *caminho* existir e for um link simbólico

-r *caminho*

Verdadeiro se o *caminho* existir e puder ser lido (acessado)

-s *caminho*

Verdadeiro se o *caminho* existir e seu tamanho for maior que zero

-w *caminho*

Verdadeiro se o *caminho* existir e puder ser escrito

-x *caminho*

Verdadeiro se o *caminho* existir e for executável

caminho1 -ot caminho2

Verdadeiro se *caminho1* for outro que não *caminho2*

Opções de avaliação de `test` para texto:

-n *escrito*

Verdadeiro se o tamanho de *escrito* for diferente de zero

-z *escrito*

Verdadeiro se o tamanho de *escrito* for zero

escrito1 == escrito2

Verdadeiro se *escrito1* for igual a *escrito2*

escrito1 != escrito2

Verdadeiro se *escrito1* for diferente de *escrito2*

Opções de avaliação de `test` para números:

num1 -lt num2

Verdadeiro se *num1* for menor que *num2*

num1 -gt num2

Verdadeiro se *num1* for maior que *num2*

num1 -le num2

Verdadeiro se *num1* for menor ou igual a *num2*

num1 -ge num2

Verdadeiro se *num1* for maior ou igual a *num2*

num1 -eq num2

Verdadeiro se *num1* for igual a *num2*

num1 -ne num2

Verdadeiro se *num1* for diferente de *num2*

Substituição de Comandos

Um dos principais propósitos de um script é trabalhar com os dados produzidos por outros comandos. Para retornar a saída de um comando, o mesmo é colocado entre aspas simples invertidas ` ` ou entre \$ (). Exemplo:

```
TRESLINHAS=`cat -n3 /etc/inputrc`  
echo "As três primeiras linhas de /etc/inputrc:"  
echo $TRESLINHAS
```

Resultado idêntico será produzido na forma:

```
TRESLINHAS=$(cat -n3 /etc/inputrc)
echo "As três primeiras linhas de /etc/inputrc:"
echo $TRESLINHAS
```

Operações matemáticas com números inteiros são feitas através da instrução `expr`:

```
SOMA=`expr $VALOR1 + $VALOR2`
```

Produz resultado idêntico:

```
SOMA=$(( $VALOR1 + $VALOR2 ))
```

Instruções de Laço

for

A instrução `for` executa ação para cada elemento de uma lista. Neste caso, para cada número gerado pelo comando `seq`:

```
for i in $(seq -w 5); do
    echo "Baixando foto_${i}.jpg";
    echo wget http://www.fotos.com/foto_${i}.jpg;
done
```

Produzirá a saída:

```
Baixando foto_1
wget http://www.fotos.com/foto_1.jpg
Baixando foto_2
wget http://www.fotos.com/foto_2.jpg
Baixando foto_3
wget http://www.fotos.com/foto_3.jpg
Baixando foto_4
wget http://www.fotos.com/foto_4.jpg
Baixando foto_5
wget http://www.fotos.com/foto_5.jpg
```

until

A instrução `until` Executa um ação até que uma afirmação seja verdadeira. Por exemplo, adicionar uma linha ao arquivo `texto_simples` até que alcance 10 linhas:

```
LENTEXTO=$(wc -l texto_simples)

until [ ${LENTEXTO%% *} -eq 10 ]; do
    echo "Mais uma linha" >> texto_simples
    LENTEXTO=$(wc -l texto_simples)
done
```

while

A instrução `while` é semelhante a instrução `until`, mas executa uma ação enquanto a afirmação `for` verdadeira. Por exemplo, adicionar uma linha ao arquivo `texto_simples` até que alcance 20 linhas (em outras palavras, enquanto `for` inferior a vinte linhas):

```
LENTEXTO=$(wc -l texto_simples)

while [ ${LENTEXTO%% *} -lt 20 ]; do
    echo "E dá-lhe linha" >> texto_simples
    LENTEXTO=$(wc -l texto_simples)
done
```

Local, Propriedade e Permissão

Para que um script possa ser usado por todos os usuários, é importante que ele seja executável e que esteja num diretório incluído na variável `PATH`. Direito de escrita deve ser retirado para todos exceto o dono (root). Por ser um arquivo mais vulnerável, o sistema não aceitará o bit SUID para arquivos script.

Scripts para Tarefas Administrativas

Scripts são comumente utilizados para realizar tarefas morosas de administração de sistemas, como backup de dados e atualizações. Para facilitar o acompanhamento dessas tarefas, o próprio script pode enviar um email ao administrador, geralmente informando sobre falhas. Se a saída do último comando (`$!`) foi diferente de 0, é porque houve uma falha.

Enviar email para o administrador informando sobre um erro:

```
mail -s "Erro em $HOSTNAME" admin@casadepraia.com < backup.log
```


Tópico 111: Tarefas Administrativas

Objetivo 1.111.1: Administrar Contas de Usuário, Grupos e Arquivos de Sistema Relacionados

Peso: 4

Conta de Usuário

O comando `/usr/sbin/useradd` cria uma nova conta no sistema. `/usr/sbin/adduser` é um link para `useradd`. Valores padrão são usados quando nenhum argumento é fornecido ou através de `useradd -D`. Em algumas distribuições essas informações podem ser verificadas e alteradas em `/etc/default/useradd`.

Opções comuns de `useradd`:

-c *comentário*

Comentário (geralmente o nome completo do usuário).

-d *diretório*

Caminho para o diretório pessoal do usuário.

-g *grupo*

Grupo inicial (GID). Precisa existir no sistema.

-G *grupo1,grupo2*

Grupos adicionais, separados por vírgula.

-u *UID*

UID (user ID) do usuário.

-s *shell*

Shell padrão para o usuário.

-p *senha*

Senha (entre aspas).

-e *data*

Data de validade da conta.

-k

Copia o diretório modelo `/etc/skel`.

-m

Cria o diretório pessoal, se não existir.

Para que o usuário possa acessar sua conta, o administrador precisará definir uma senha para ele, o que pode ser feito através do comando `passwd usuário`. Usado sem argumentos, `passwd` altera a senha para o usuário atual. O campo de descrição pode ser alterado com o comando `chfn` e o shell inicial com `chsh`. Usuários comuns podem usar estes comandos para alterar apenas suas próprias

contas.

Uma conta de usuário pode ser apagada com o comando **userdel**. A opção **-r** assegura que o diretório pessoal do usuário também seja apagado.

As informações de conta dos usuários do sistema são armazenadas no arquivo **/etc/passwd**.

Exemplo de usuários em /etc/passwd:

```
root:x:0:0::/root:/bin/bash
luciano:x:1000:100:Luciano Antonio Siqueira:/home/luciano:/bin/bash
```

Cada usuário é definido em uma linha, em campos separados por “:” representando respectivamente:

1. Nome de Login
2. Senha (“x” se usando o arquivo `/etc/shadow`)
3. Número de identificação do usuário (UID)
4. Número do grupo efetivo do usuário (GID)
5. Descrição breve do usuário (opcional)
6. Diretório pessoal para o usuário
7. Shell inicial do usuário (se vazio, `/bin/sh` será usado)

Para editar diretamente o arquivo `/etc/passwd`, é altamente indicado usar o comando **vipw**, que bloqueia o arquivo `/etc/passwd` contra possíveis alterações externas, evitando corrupção do arquivo. O editor utilizado será o determinado pela variável de ambiente `VISUAL` ou `EDITOR`. Se ambas não existirem, o editor utilizado será o `vi`. Usado com a opção `-s`, **vipw** abrirá para edição o `/etc/shadow`.

O arquivo `/etc/passwd` é legível por qualquer usuário (`-rw-r--r--`), o que pode tornar as senhas criptografadas passíveis de serem decodificadas. Para evitar essa possibilidade, é usado um segundo arquivo, acessível apenas ao super-usuário, `/etc/shadow` (`-rw-r-----`). Para converter as senhas de arquivos `/etc/passwd` antigos para `/etc/shadow`, utiliza-se o comando `pwconv`. Para retornar as senhas para o formato `/etc/passwd` antigo, utiliza-se `pwunconv`.

Como no arquivo `/etc/passwd`, os campos no arquivo `/etc/shadow` são separados por “:”, correspondendo respectivamente:

1. Nome de usuário, que deve corresponder a um nome válido em `/etc/passwd`
2. A senha, criptografada numa sequência de 13 caracteres. Em branco permite login sem senha. Com um asterisco “*” indica que a conta está bloqueada.
3. O número de dias (desde 01/01/1970) desde que a senha foi alterada.
4. Número mínimo de dias até que uma senha possa ser novamente alterada. 0 permite alterar a senha sem esperar.
5. Número de dias depois do qual a senha deverá ser alterada. Por padrão 99999, ou 274 anos.
6. Número de dias para advertir o usuário sobre a senha a expirar.
7. Número de dias depois da senha expirar após o qual a conta será bloqueada.
8. O número de dias, a partir de 01/01/1970, desde que a conta foi bloqueada.
9. Campo reservado.

Essas informações referentes à validade da senha podem ser modificadas através do programa **chage**, com as seguintes opções:

-m dias

Mínimo de dias até que o usuário possa trocar uma senha modificada.

-M dias

Número máximo de dias que a senha permanecerá válida.

-d dias

Número de dias decorridos em relação a 01/01/1970 em que a senha foi mudada. Também pode ser expresso no formato de data local (dia/mês/ano).

-E dias

Número de dias decorridos em relação a 01/01/1970 a partir do qual a conta não estará mais disponível. Também pode ser expresso no formato de data local (dia/mês/ano).

-I dias

Inatividade, tolerância de dias após a senha ter expirado até que a conta seja bloqueada.

-W dias

Dias anteriores ao fim da validade da senha nos quais será emitido um aviso a respeito.

Para usuários comuns, **chage** só pode ser usado com a opção **-l usuário**, que mostra as restrições referentes ao *usuário* em questão.

O comando **usermod** agrega as funções de alteração de conta de usuário, através das opções:

-c descrição

Descrição do usuário

-d diretório

Altera diretório do usuário. Com o argumento **-m** move o conteúdo do diretório atual para o novo

-e valor

Prazo de validade da conta, especificada no formato dd/mm/aaaa

-f valor

Número de dias após a senha ter expirado até que a conta seja bloqueada. Um valor **-1** cancela essa função.

-g grupo

Grupo efetivo do usuário

-G grupo1,grupo2

Grupos adicionais para o usuário

-l nome

Nome de login do usuário

-p senha

Senha

-u *UID*

Número de identificação (UID) do usuário

-s *shell*

Shell padrão do usuário

-L

Bloqueia a conta do usuário, colocando um sinal “!” na frente da senha criptografada. Uma alternativa é substituir o shell padrão do usuário por um script ou programa que informe as razões do bloqueio.

-U

Desbloqueia a conta do usuário, retirando o sinal “!” na frente da senha criptografada

Grupos de Usuários

Para criar um grupo, é usado o comando `groupadd`:

```
# groupadd estudo_c
```

O número de identificação do grupo (GID) pode ser especificado através da opção `-g`. Para apagar um grupo, o comando `groupdel`:

```
# groupdel estudo_c
```

Um usuário poderá ser incluído/excluído de um grupo através do comando **gpasswd**, utilizando-se o argumento adequado:

`gpasswd grupo`

Cria uma senha para *grupo*

`gpasswd -r grupo`

Apaga a senha para *grupo*

`gpasswd -a usuário grupo`

Associa *usuário* a *grupo*

`gpasswd -d usuário grupo`

Exclui *usuário* de *grupo*

`gpasswd -A usuário grupo`

Torna *usuário* administrador de *grupo*.

Um usuário pode pertencer a mais de um grupo, mas apenas um grupo deve ser o grupo principal. Para mostrar os grupos aos quais um usuário pertence, é usado o comando `groups`:

```
# groups root
root : root bin daemon sys adm disk wheel floppy video
```

`groups` sem argumentos mostra os grupos para o usuário atual. O comando `id` mostra os grupos para o usuário, mostrando também o número de identificação do usuário e dos grupos:

```
# id root
uid=0(root) gid=0(root) grupos=0(root),1(bin),2(daemon),3(sys),
4(adm),6(disk),10(wheel),11(floppy),18(video)
```

O comando `newgrp` é usado para alterar o grupo efetivo do usuário para o grupo solicitado em uma nova sessão de login:

```
$ groups
users disk lp floppy audio video cdrom
$ newgrp video
$ groups
video disk lp floppy audio cdrom users
$ exit
$ groups
users disk lp floppy audio video cdrom
```

Caso o usuário não pertença ao grupo em questão, será associado.

As informações sobre os grupos existentes no sistema são armazenadas em `/etc/group`. Exemplo de `/etc/group`:

```
ftp::50:
pop::90:pop
scanner::93:
nobody::98:nobody
nogroup::99:
users::100:
console::101:
messagebus:x:102:
estudo_c:x:104:luciano
```

Cada grupo é definido em uma linha, em campos separados por “:” representando respectivamente:

1. Nome do grupo
2. Senha para o grupo (“x” se utilizar `/etc/gshadow`)
3. Número de identificação do grupo (GID)
4. Lista de membros do grupo, separados por vírgula

Para editar diretamente o arquivo `/etc/group`, é altamente indicado usar o comando **vigr**, que bloqueia o arquivo `/etc/group` contra possíveis alterações externas, evitando corrupção do arquivo. O editor utilizado será o determinado pela variável de ambiente `VISUAL` ou `EDITOR`. Se ambas não existirem, o editor utilizado será o `vi`. Usado com a opção `-s`, `vigr` abrirá para edição o arquivo `/etc/gshadow`.

Como o caso do arquivo `/etc/passwd`, é possível usar um segundo arquivo para armazenar informações referentes à senha dos grupos, chamado `/etc/gshadow`. O comando **grpconv** converte as senhas no formato antigo `/etc/group` para `/etc/gshadow` e **grpunconv** realiza o procedimento inverso.

O comando `groupmod` agrega algumas funções de alteração de grupos, pelas opções:

-g *GID*

Altera o número (*GID*) do grupo

-n *nome*

Altera o nome do grupo

Objetivo 1.111.2: Ajustar o Ambiente do Usuário e Variáveis de Ambiente de Sistema

Peso: 3

Novos diretórios pessoais podem ser criados a partir de um modelo situado em `/etc/skel`. Este diretório pode conter arquivos modelos de arquivos como `.bashrc`, `.bash_profile` ou quaisquer outro que o administrador julgue interessante existir no diretório pessoal do usuário. Para que o diretório modelo `/etc/skel` seja utilizado, é necessário passar a opção `-k` para `useradd`.

As variáveis de sistema comuns a todos os usuários podem ser definidas em `/etc/profile`. A cada novo login, o shell usará este arquivo para definir certas características do ambiente do usuário, como as variáveis `PATH` e `VISUAL`, e a máscara padrão de criação de arquivos `umask`.

Exemplo de /etc/profile:

```
(...)  
export LESSOPEN="|lesspipe.sh %s"  
export LESS="-M"  
export VISUAL="vim"  
  
# If the user doesn't have a .inputrc, use the one in /etc.  
if [ ! -r "$HOME/.inputrc" ]; then  
    export INPUTRC=/etc/inputrc  
fi  
(...)
```

Através do comando `env` pode-se executar um programa - indicado como argumento - ignorando as configurações de ambiente atual, através da opção `-i` ou simplesmente `-`. Sem nenhum programa como argumento, lista todas as configurações do ambiente resultante. Uma única variável pode ser ignorada se especificada com a opção `-u`. Para modificar o valor de uma variável para execução do programa, basta fornecê-la como argumento no formato `VARIÁVEL=VALOR`.

O arquivo `/etc/login.defs` permite alterar o comportamento de funções relacionadas ao uso de `/etc/shadow`, para todos usuários. É possível definir opções padrão para criação de novos usuários, como prazo de validade das senhas e criação automática de diretório pessoal. O comportamento do login do sistema também pode ser alterado, como intervalo entre tentativas frustradas de login e checagem automática de email. A maioria das opções é bastante documentada no próprio arquivo.

Objetivo 1.111.3: Configurar e Recorrer a Arquivos de log para Corresponder às Necessidades Administrativas e de Segurança

Peso: 3

A maioria dos arquivos de log são armazenados no diretório `/var/log/`. Enquanto alguns programas geram os próprios arquivos de log, como o Xorg e o samba, a maioria dos logs do sistema são controlados pelo daemon `syslogd`.

Logs de sistema comuns:

`/var/log/cron`

Rastreia as mensagens de execução do *cron*.

`/var/log/mail`

Mensagens relacionadas ao envio de mensagens de email.

`/var/log/messages`

Todas as mensagens do kernel, exceto *authpriv*, *cron*, *mail* e *news*.

`/var/log/secure`

Logins fracassados, inclusão/remoção de usuários ou grupos, etc.

O daemon *syslogd* é configurado através do arquivo `/etc/syslog.conf`. Cada regra de configuração é separada em dois campos, seletor e ação, separados por espaço(s) ou tabulação(ões).

O campo seletor é dividido em duas partes: *facilidade* e *prioridade*, separadas por um ponto. *Facilidade* indica o sub-sistema originário da mensagem, e pode ser um dos seguintes termos: *auth*, *authpriv*, *cron*, *daemon*, *ftp*, *kern*, *lpr*, *mail*, *news*, *syslog*, *user*, *uucp* e *local0* até *local7*.

Prioridade define a gravidade da mensagem e pode ser um dos termos, em ordem crescente de gravidade: *debug*, *info*, *notice*, *warning*, *err*, *crit*, *alert*, *emerg*. O termo *none* indica que não há prioridade para a facilidade em questão.

O caracter asterisco “*” indica que a regra vale para qualquer facilidade ou prioridade, dependendo de que lado do ponto está. Mais de uma facilidade pode ser especificada para a prioridade na mesma regra, separando-as por vírgula.

O sinal “=” confere exclusividade à facilidade/prioridade que precede. Em contrapartida, o sinal “!” faz ignorar a facilidade/prioridade que precede.

O sinal “;” pode ser usado para separar mais de um seletor para a mesma ação.

O campo ação determina o destino dado à mensagem em questão. Geralmente as mensagens são enviadas para arquivos em `/var/log/`, mas podem ser direcionadas *pipes*, consoles, máquina remota, usuário(s) específico(s), e para todos os usuários no sistema.

Exemplo de /etc/syslog.conf:

```
# Todas mensagens do kernel irão para o arquivo kernel.
# Mensagens críticas e maiores irão para o host finlandia
# e para o console. Mensagens info e maiores, a exceção de
# mensagens de erro e maiores, irão para o arquivo kernel-info.
#
kern.*                /var/adm/kernel
kern.crit             @finlandia
kern.crit             /dev/console
kern.info;kern.!err  /var/adm/kernel-info
```

Após alterar o arquivo `/etc/syslog.conf`, é necessário reiniciar o daemon *syslogd* para que utilize as novas configurações.

Através do comando *logger* é possível criar mensagens de log manualmente, indicadas como argumento do comando. A opção `-p` permite de terminar a *facilidade.prioridade* para a mensagem.

Para acompanhar continuamente a atualização de arquivos de log, é interessante o uso de **tail -f** “*arquivo de log*”, que mostrará novas mensagens à medida que forem acrescentadas.

Como os arquivos de log são continuamente ampliados, é bastante indicado que as mensagens mais

antigas sejam movidas, para evitar que o arquivo de log inche demais. Essa tarefa é realizada através do programa **logrotate**. Normalmente, **logrotate** é agendado para execução diária.

Seu arquivo de configuração é **/etc/logrotate.conf**, onde regras de corte, compressão, envio por email e outras podem ser especificadas para cada arquivo de log.

Objetivo 1.111.4: Automatizar Tarefas Administrativas de Sistema Agendando Trabalhos para Execução Futura

Peso: 4

Existem dois sistemas principais de agendamento de tarefas no Linux, o **at** e o **cron**. O **at** é mais utilizado para agendamentos simples de comandos, enquanto o **cron** é mais utilizado para agendar procedimentos recorrentes do sistema.

at

Essas tarefas são agendadas via linha de comando, através do comando **at**, no formato:

at quando comando

Onde quando pode ser, por exemplo:

now

3am + 2 days

midnight

10:15 april 12

teatime

Outras opções de datas e formatos podem ser consultados em **/usr/share/doc/at-xxx/timespec**.

Os agendamentos ficam armazenados em **/var/spool/at/***. Para conferir os agendamentos do usuário atual, usa-se **at -l** ou **atq**. Um agendamento pode ser apagado através de seu número específico, fornecido para o comando **atrm**.

Usuários comuns poderão usar **at** se constarem no arquivo **/etc/at.allow**. Se **/etc/at.allow** não existir, o arquivo **/etc/at.deny** será consultado e serão bloqueados ao uso do **at** os usuários que nele constarem. Se nenhum dos arquivos existir, apenas o usuário **root** poderá usar o **at**.

cron

A cada minuto, o daemon **crond** lê as tabelas de agendamento (**crontabs**) contendo tarefas a serem executadas em data e hora específicas. Crontabs de usuários ficam em **/var/spool/cron/***. O crontab do sistema é **/etc/crontab**. Esses arquivos não devem ser editados diretamente, mas através do comando **crontab**.

Opções de crontab:

crontab -l usuário

Mostra as tarefas agendadas por *usuário*

```
crontab -e usuário
```

Edita o crontab de *usuário* no editor padrão do sistema

```
crontab -d usuário
```

Apaga o crontab de *usuário*

Se não fornecido, será assumido como usuário o usuário atual.

Cada linha no arquivo crontab representa uma tarefa, no formato:

```
0-59 0-23 0-31 1-12 0-6 comando
|      |      |      |      |
|      |      |      |      | `--> Dia da Semana
|      |      |      |      |
|      |      |      |      | `--> Mês
|      |      |      |      |
|      |      |      |      | `--> Dia
|      |      |      |      |
|      |      |      |      | `--> Hora
|      |      |      |      |
|      |      |      |      | `--> Minuto
```

O traço “-” delimita um período para execução. O caractere asterisco “*” em um campo determina a execução do comando sempre que o agendamento corresponder a qualquer marcação para o campo em questão. O caracter barra “/” estabelece um passo para a execução.

Executar script_backup a cada quatro horas, de segunda à sexta, nos meses de maio e junho:

```
* */4 * 5,6 1-5 /usr/local/bin/script_backup
```

É importante especificar o caminho completo para o comando, pois a tarefa terá apenas as variáveis de ambiente USER, HOME e SHELL.

Se a tarefa produzir alguma saída, esta será enviada por email para o usuário. Para evitar esse comportamento, basta redirecionar a saída da tarefa para `/dev/null` ou para um arquivo.

É possível controlar o uso do crontab através dos arquivos `/etc/cron.allow` e `/etc/cron.deny`. Se `/etc/cron.allow` existir, apenas os usuários que nele constarem poderão agendar tarefas. Se `/etc/cron.deny` existir, os usuários nele existentes serão proibidos de agendar tarefas. Se nenhum dos arquivos existirem, todos usuário poderão agendar tarefas.

Objetivo 1.111.5: Manter uma Estratégia Eficiente de Backup de Dados

Peso: 3

As ferramentas de backup mais tradicionais são **tar**, **cpio** e **dump**. **tar** e **cpio** são semelhantes, aglutinam arquivos em arquivos ou fitas e podem extrair e atualizar estes arquivos. **dump** age de forma diferente, lidando diretamente com o sistema de arquivos monolítico.

Em linhas gerais, **tar** e **cpio** são utilizados para arquivar grupos de arquivos específicos, dentro de um ou mais sistemas de arquivos.

Um esquema simples de backup é arquivar todos os dados na primeira vez e depois arquivar apenas os dados modificados. O primeiro passo chama-se backup completo e o segundo backup incremental.

O backup completo leva mais tempo devido a necessidade de copiar todos os dados. No entanto, apesar do backup incremental ser mais rápido, o procedimento de recuperar os arquivos de um backup incremental é mais moroso, uma vez que será necessário mais de um procedimento de restauração.

Fazer backup de /etc em múltiplos disquetes usando tar:

```
# tar cMf /dev/fd0 /etc
```

Fazer backup de /etc em um drive de fita usando tar:

```
# tar cf /dev/ftape /etc
```

Verificar se há diferença entre os dados copiados e os originais:

```
# tar dvf /dev/ftape
```

O backup incremental é realizado informando a data base com a opção `--newer (-N)`. Apenas os arquivos criados a partir da data fornecida serão arquivados.

```
# tar cf /dev/ftape --newer '12 apr 2006' /etc
```

Verificando se há erros no backup:

```
# tar tvf /dev/ftape
```

Arquivos apagados dos dados atuais não serão apagados no backup incremental através do `tar`.

Dessa forma, poderão haver diversos níveis de backup, a partir de um nível 0 (backup completo).

Enquanto que com o `tar` e `cpio` a criação e recuperação de níveis de backup é feita manualmente, com o comando `dump` ela é nativa, a recuperação de um nível automaticamente recuperará os níveis anteriores. Utilizando-se do `dump`, as informações referentes ao backup são armazenadas `/etc/dumpdates`. O comando `restore` é utilizado para restaurar o backup. Arquivos de backup `dump` podem ser testados com o comando `restore -t arquivo`.

Outra possibilidade é criar uma imagem de um dispositivo, utilizando o comando `dd`:

```
# dd if=/dev/hda of=backup-hda.img
```

A imagem preservará o sistema de arquivos e até a MBR (se houver) do dispositivo `/dev/hda`. Para restaurar o backup, basta inverter os argumentos:

```
# dd if=backup-hda.img of=/dev/hda
```

O dispositivo em questão deverá estar desmontado ou montado como somente-leitura.

Objetivo 1.111.6: Manutenção de Data e Hora do Sistema

Peso: 4

Relógios e Fuso Horário

O kernel do Linux mantém um relógio separado do relógio do hardware (BIOS). Durante o boot, o relógio do kernel lê o relógio do hardware e a partir daí roda distintamente. Esse procedimento se justifica pois ler o relógio do hardware é lento e complicado. O relógio do kernel guarda a hora universal, de modo que fusos horários são calculados por cada processo através das ferramentas `timezone`.

O relógio do hardware pode estar em hora local ou em hora universal. É preferível que esteja em hora

universal, assim não será necessário modificá-lo no período de horário de verão.

O fuso horário do sistema é determinado pelo arquivo `/etc/localtime`, que geralmente é um link simbólico apontando para o arquivo real em `/usr/share/zoneinfo/` ou em `/usr/lib/zoneinfo/`. Um usuário poderá alterar o fuso horário para si definindo a variável de ambiente `TZ`. O formato da variável `TZ` está elucidado no manual do comando `tzset`.

O comando `date` é usado para mostrar a hora e data do sistema:

```
$ date
Qua Abr 12 18:38:05 BRT 2006
```

`date -u` mostra a hora universal:

```
$ date -u
Qua Abr 12 21:38:09 UTC 2006
```

O próprio comando `date` é usado para alterar o relógio de software (do kernel).

```
# date MMDDhhmmCCYY.ss
| | | | | | | |
| | | | | | | | `--> segundos (opcional)
| | | | | | | |
| | | | | | | | `--> Ano, porção da década (opcional)
| | | | | | | |
| | | | | | | | `--> Ano, porção do século (opcional)
| | | | | | | |
| | | | | | | | `--> Minutos
| | | | | | | |
| | | | | | | | `--> Horas
| | | | | | | |
| | | | | | | | `--> Dia
| | | | | | | |
| | | | | | | | `--> Mês
```

A opção `-u` especifica que a data informada refere-se à hora universal.

Para mostrar ou alterar o relógio do hardware, é usado o comando `hwclock`. Com o argumento `-w` atualiza o relógio de hardware pelo relógio de software. Com o argumento `-s` atualiza o relógio de software com o relógio de hardware. Como no comando `date`, o argumento `-u` indica hora universal.

NTP – Network Time Protocol

Um computador em rede pode manter seu relógio atualizado comparando a hora com um outro computador na rede que tenha um relógio preciso. Isso é possível através do protocolo NTP.

Para um sistema usar o NTP, o arquivo `/etc/ntp.conf` deve estar configurado adequadamente e o daemon `ntpd` deve estar ativo. O `ntpd` utiliza o protocolo UDP, através da porta de comunicação 123.

Exemplo básico de /etc/ntp.conf:

```
server br.pool.ntp.org
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org

driftfile /etc/ntp.drift
```

Neste arquivo `/etc/ntp.conf` mínimo, foram definidos apenas os servidores NTP e o arquivo

drift. Servidores NTP públicos podem ser encontrados em <http://www.pool.ntp.org/>. Outras opções, como restrição de acesso ao servidor NTP local, podem ser consultadas na documentação encontrada em `/usr/doc/ntp-x.x.x/`.

A indicação do arquivo `/etc/ntp.drift` é importante pois é nele que o `ntpd` armazenará as estatísticas de erro, projetando o intervalo de erro do relógio do sistema e atualizando-o ao passo dessa projeção.

Se já estiver rodando, o daemon `ntpd` deverá ser reiniciado para utilizar as novas configurações. Em execução, o `ntpd` poderá funcionar como servidor NTP para outras máquinas na rede.

Para conferir o andamento do `ntpd`, pode ser usado o comando `ntpq`:

#	ntpq -np									
	remote	refid	st	t	when	poll	reach	delay	offset	jitter
=====										
+200.218.160.160	200.20.186.75		2	u	80	128	177	203.427	76.433	68.727
+72.21.46.202	67.52.51.34		3	u	1	256	377	187.361	29.718	79.019
+24.123.66.139	192.168.1.231		2	u	1	256	337	210.402	39.727	120.724
*64.136.200.96	.WWVB.		1	u	200	128	36	235.761	43.737	75.380

Caso os valores locais de hora difiram do servidor, o `ntpd` irá aproximar lentamente a hora até que se corresponda, para evitar mudanças bruscas que podem causar confusão no sistema.

Para forçar o ajuste imediato do relógio, é utilizado o comando `ntpdate servidor`.

Tópico 112: Fundamentos de Redes

Objetivo 1.112.1: Fundamentos de TCP/IP

Peso: 4

Endereço IP

Endereços IP no formato `xxx.xxx.xxx.xxx` (*dotted quad*) são a expressão em números decimais de um endereço de rede binário. Cada um dos quatro campos separados por pontos corresponde a um byte, geralmente chamados octetos. Por exemplo, o número ip:

192.168.1.1

Corresponde à forma binária:

11000000.10101000.00000001.00000001

Tabela simples para conversão de valores binários para decimal:

00000001 → 2^0 → 1

00000010 → 2^1 → 2

00000100 → 2^2 → 4

00001000 → 2^3 → 8

00010000 → 2^4 → 16

00100000 → 2^5 → 32

01000000 → 2^6 → 64

10000000 → 2^7 → 128

Cada interface de rede numa mesma rede deverá ter um endereço IP único.

Endereço de rede, máscara de rede e endereço broadcast

Além do endereço da interface, um número IP contém o endereço de sua rede, que é determinado através da máscara de rede ou máscara de sub-rede. O cálculo é feito a partir da forma binário dos números IP.

Máscara de 16bit:

11111111.11111111.00000000.00000000 → 255.255.0.0

Máscara de 17bit:

11111111.11111111.10000000.00000000 → 255.255.128.0

Na primeira máscara (16bit), pertencerão a mesma rede os IPs cujos dois primeiros octetos do endereço não difiram. Na segunda máscara (17bit), pertencerão a mesma rede os IPs cujos dois primeiros octetos e o primeiro bit do terceiro octeto do endereço não difiram.

Dessa forma, dois endereços de interface 172.16.33.8 e 172.16.170.3 estariam na mesma rede se a máscara for de 16bit, mas não se a máscara for de 17bit. As máscaras de rede variam dependendo do contexto da rede.

Conseqüentemente, o endereço da rede corresponde à parte do número IP determinado pelos bits marcados da máscara de rede. Para um host 172.16.33.8 com máscara de rede 255.255.0.0, o endereço da rede será 172.16.0.0.

O endereço *broadcast* é o número IP que designa todas as interfaces numa rede. Para um endereço de rede 172.16.0.0, o endereço broadcast será 172.16.255.255.

Através de equações com operadores matemáticos lógicos (AND, OR e NOT) os endereços binários são calculados bit a bit:

IP de rede = IP de interface AND máscara de rede

Broadcast = IP de rede OR NOT máscara de rede

Classes de Redes

Para redes privadas (LANs) há uma certa gama específicas de IPs que podem ser usados e que não devem ser aplicados a interfaces ligadas à internet.

Endereços reservados a redes privadas:

Classe A

1.0.0.0 até 127.0.0.0. Endereços de rede de 8bit e endereços de interfaces de 24bit. O primeiro octeto do número IP representa o endereço da rede. A máscara padrão será 255.0.0.0. Permite aproximadamente 1.6 milhão de IPs de interface por rede.

Classe B

128.0.0.0 até 191.255.0.0. Endereços de rede de 16bit e endereços de interfaces de 16bit. Os dois primeiros octetos representam o endereço da rede. A máscara padrão é 255.255.0.0. Permite 16.320 redes com 65.024 IPs de interface para cada uma.

Classe C

192.0.0.0 até 223.255.255.0. Endereços de rede de 24bit e endereços de interfaces de 8bit. Os três primeiros octetos representam o endereço da rede. A máscara padrão é 255.255.255.0. Permite aproximadamente 2 milhões de redes com 254 IPs de interface cada.

Subredes

Subredes podem ser definidas através da máscara de rede, avançando sobre os bits referentes à interface. Dessa forma, uma rede pode ser dividida em redes menores, sem classe, chamadas CIDR – Classless InterDomain Rounting.

Por exemplo, uma rede classe C 192.168.1.0. Ativando o primeiro bit do quarto octeto na máscara de rede, os primeiros 25bit do IP seriam referentes ao endereço da rede:

Máscara de rede:

11111111.11111111.11111111.10000000
255.255.255.128

Conseqüentemente, a rede classe C foi subdividida em duas subredes sem classe, de 25bit:

Primeira SubRede: 192.168.1.0

Os caracteres “x” correspondem aos bits de interfaces nessa subrede:

11000000.10101000.00000001.0xxxxxxx

192.168.1.1 até 192.168.1.126

Endereço de broadcast:

11000000.10101000.00000001.01111111

192.168.1.127

Segunda SubRede: 192.168.1.128

Os caracteres “x” correspondem aos bits de interfaces nessa subrede:

11000000.10101000.00000001.1xxxxxxx

192.168.1.129 até 192.168.1.254

Endereço de broadcast:

11000000.10101000.00000001.11111111

192.168.1.255

Uma rede de 26bit resultaria no seguinte cenário:

Máscara de rede:

11111111.11111111.11111111.11000000

255.255.255.192

Subredes resultantes:

Primeira SubRede: 192.168.1.0

Interfaces:

11000000.10101000.00000001.00xxxxxx

192.168.1.1 até 192.168.1.62

Broadcast:

11000000.10101000.00000001.00111111

192.168.1.63

Segunda SubRede: 192.168.1.64

Interfaces:

11000000.10101000.00000001.01xxxxxx

192.168.1.65 até 192.168.1.126

Broadcast:

11000000.10101000.00000001.01111111

192.168.1.127

Terceira SubRede: 192.168.1.128

Interfaces:

11000000.10101000.00000001.10xxxxxx

192.168.1.129 até 192.168.1.190

Broadcast:

11000000.10101000.00000001.10111111

192.168.1.191

Quarta SubRede: 192.168.1.192

Interfaces:

11000000.10101000.00000001.11xxxxxx

192.168.1.193 até 192.168.1.254

Broadcast:

11000000.10101000.00000001.11111111

192.168.1.255

É importante lembrar que, como cada subrede ocupa dois IPs para seus respectivos endereços de rede e broadcast, o total de IPs para as interfaces será proporcionalmente reduzido. O número possível de hosts para cada subrede pode ser calculado com a fórmula $2^{(32 - masc)} - 2$, onde *masc* é o número de bits usados para a máscara de rede.

Um endereço IP pode demonstrar a informação de endereço da rede, máscara de rede e broadcast numa forma abreviada, exemplo:

192.168.1.129/25

O número 25 após a barra indica a quantidade de bits reservados para o endereço de rede. Conclui-se que é uma rede CIDR com máscara de subrede 255.255.255.128, de endereço 192.168.1.128 e broadcast 192.168.1.255.

IPv4 e IPv6

O padrão tradicional de 32bit (quatro octetos de bits) dos números IP é conhecido como IPv4. Há outro padrão mais recente, conhecido como IPv6, que consiste numa sequência de 128bit. A vantagem óbvia do IPv6 sobre o IPv4 é a possibilidade de um número muito maior de números IP. Enquanto que o IPv4 é capaz de gerar 4.3x10⁹ (4.3 bilhões) de endereços, o IPv6 é capaz de gerar 3.4x10³⁸ (50 octilhões de endereços).

Um endereço IPv6 é normalmente escrito na forma de oito grupos de quatro números hexadecimais. Exemplo de um endereço IPv6:

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

O IPv4 ainda é largamente mais utilizado e é possível a intercomunicação entre os dois padrões, mas a medida em que cada vez mais dispositivos demandarem o uso de um endereço IP, o padrão IPv6 tornar-se-á o vigente.

Protocolos de rede

Vários protocolos são necessários para transmissão de dados numa rede. Protocolos são a “linguagem” usada na comunicação entre dois hosts, permitindo a transmissão de dados. Os principais protocolos:

IP

Internet Protocol – Protocolo do qual se utilizam os protocolos TCP, UDP e ICMP para endereçar e localizar a transmissão de dados.

TCP

Transfer Control Protocol – Protocolo de controle da formatação e integridade dos dados transmitidos. O processo não sofre interferência da esfera de atuação do aplicativo transmissor do dados.

UDP

User Datagram Protocol – Exerce a mesma função do TCP, porém as os procedimentos são realizados na esfera de atuação do aplicativo transmissor dos dados.

ICMP

Internet Control Message Protocol – Permite a comunicação entre roteadores e hosts para que identifiquem e relatem o estado de funcionamento da rede.

PPP

Point to Point Protocol – Permite a conexão entre dois computadores, geralmente através de uma interface serial e modem, possibilitando seu uso como interface de rede.

Portas TCP e UDP

Os protocolos de rede tornam possível a comunicação dos serviços de rede (FTP, HTTP, SMTP, etc) assinalando uma porta específica para cada um deles. É muito importante que todos os computadores interligados respeitem os números de porta corretos para cada serviço. A lista oficial de portas e serviços associados é controlado pela IANA – Internet Assigned Numbers Authority – <http://www.iana.org/assignments/port-numbers>.

No Linux, a lista de serviços conhecidos e suas portas consta em **/etc/services**. As portas são campos de 16bit, existindo portanto um máximo de 65535 portas.

Principais portas e serviços, extraído de /etc/services:

ftp-data	20/tcp	#File Transfer [Default Data]
ftp-data	20/udp	#File Transfer [Default Data]
ftp	21/tcp	#File Transfer [Control]
ftp	21/udp	#File Transfer [Control]
ssh	22/tcp	#Secure Shell Login
ssh	22/udp	#Secure Shell Login
telnet	23/tcp	
telnet	23/udp	
#	24/tcp	any private mail system
#	24/udp	any private mail system
smtp	25/tcp	mail #Simple Mail Transfer
smtp	25/udp	mail #Simple Mail Transfer
domain	53/tcp	#Domain Name Server
domain	53/udp	#Domain Name Server
http	80/tcp	www www-http #World Wide Web HTTP
http	80/udp	www www-http #World Wide Web HTTP
pop3	110/tcp	#Post Office Protocol - Version 3
pop3	110/udp	#Post Office Protocol - Version 3
nntp	119/tcp	usenet #Network News Transfer Protocol
nntp	119/udp	usenet #Network News Transfer Protocol
ntp	123/tcp	#Network Time Protocol
ntp	123/udp	#Network Time Protocol
netbios-ssn	139/tcp	#NETBIOS Session Service
netbios-ssn	139/udp	#NETBIOS Session Service
imap	143/tcp	imap2 imap4 #Interim Mail Access Protocol v2
imap	143/udp	imap2 imap4 #Interim Mail Access Protocol v2
snmp	161/tcp	
snmp	161/udp	

Rotas de Endereços

Para que os dados possam chegar ao seu destino, é necessário que haja uma tabela de rotas no host de origem dos dados. A tabela de rotas determina o que fazer com cada pacote de dados que seja encaminhado para fora através de uma interface de rede.

Estruturalmente, os hosts numa rede podem se comunicar apenas com outros hosts na mesma rede. Se o host de destino pertencer a alguma rede ligada ao host de origem, o pacote será colocado nessa rede. Se o host de destino não pertencer a alguma rede ligada ao host de origem, o pacote deverá ser direcionado para a rota padrão, a qual são encaminhados todos os destinos desconhecidos ao host local. A máquina ou dispositivo que recebe esse pacotes é chamada roteador ou gateway, que se encarregará de encaminhar os pacotes para as redes apropriadas.

Há quatro configurações comuns de tabelas de rotas:

- Mínima → Para redes isoladas, geralmente feita quando a interface é iniciada.
- Estática → Para redes com um ou mais gateways. Geralmente é criada através de scripts automáticos ou manualmente através do comando route. Se a rede muda, a tabela precisa ser manualmente atualizada.
- Dinâmica → Em redes maiores as informações de rotas e gateways são dinamicamente fornecidas através de protocolos de roteamento. A desvantagem é que a criação dinâmica de tabelas causa maior tráfego na rede.
- Estática/Dinâmica → Geralmente as tabelas de rota contém informações estáticas para encaminhamento de pacotes dentro da rede local e uma rota padrão para demais pacotes que aponta para um gateway que trabalha com roteamento dinâmico.

A alocação dinâmica da tabela de rotas é feita através do daemon **gated**.

Exemplo de tabela de rotas:

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
200.228.60.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	200.228.60.1	0.0.0.0	UG	0	0	0	eth0

Ao host 200.228.60.1 serão encaminhados todos os pacotes de dados que não se destinem à hosts nas redes 200.228.60.0 ou 127.0.0.0 (rede loopback local). Portanto, 200.228.60.1 é o gateway padrão, que se encarregará de encaminhar corretamente os demais pacotes aos seus destinos.

Alguns comandos são úteis para verificar o correto funcionamento de uma rede:

ping

Envia pacotes ICMP ECHO_REQUEST para o host especificado e aguarda retorno de ECHO_REPLY.

host

Converte nomes para IPs e vice-versa, através de pesquisa no servidor DNS.

Exemplo:

```
# host www.gnu.org
www.gnu.org is an alias for gnu.org.
gnu.org has address 199.232.41.10
;; reply from unexpected source: 200.212.223.100#53, expected 200.230.210.6#53
;; Warning: ID mismatch: expected ID 1737, got 56531
www.gnu.org is an alias for gnu.org.
www.gnu.org is an alias for gnu.org.
gnu.org mail is handled by 30 mx30.gnu.org.
gnu.org mail is handled by 10 mx10.gnu.org.
gnu.org mail is handled by 20 mx20.gnu.org.
```

dig

Domain Information Groper - Retorna informações úteis para diagnóstico de problemas em servidores DNS. Se nenhum argumento for utilizado, dig realizará o teste padrão no(s) servidor(es) encontrados em /etc/resolv.conf.

traceroute

Mostra as rotas percorridas por um pacote até chegar ao seu destino. Limitando o campo TTL (Time To Live) dos pacotes, traceroute recebe respostas de erro ICMP TIME_EXCEEDED de cada host percorrido.

whois

Pesquisa por um nome de domínio na base de dados whois, que retorna informações sobre o registro do domínio e sobre o proprietário, como nome e contatos.

Objetivo 1.112.3: Configuração e Resolução de Problemas de Redes TCP/IP

Peso: 7

Arquivos de configuração

Os principais arquivos de configuração da rede são:

`/etc/HOSTNAME` ou `/etc/hostname`

Contém o nome atribuído a máquina local.

`/etc/hosts`

Associa os números IP da rede a nomes. Mais prático para pequenas redes.

`/etc/networks`

Semelhante ao `/etc/hosts`, atribui nomes à números de rede.

`/etc/nsswitch.conf`

Determina os locais de busca por aliases, números de rede, bancos de usuários e senhas, etc. Palavras chaves como `files`, `nis` e `dns` determinam qual deve ser a origem para o requerimento de sistema.

Exemplo de `/etc/nsswitch.conf`:

```
hosts:          files dns
networks:       files

services:      files
protocols:     files
rpc:           files
ethers:        files
netmasks:     files
netgroup:     files
bootparams:   files
```

`/etc/host.conf`

Mesma função de `/etc/nsswitch.conf` em sistemas pré glibc2.

`/etc/resolv.conf`

Especifica o(s) servidor(es) DNS.

Exemplo de `/etc/resolv.conf`:

```
domain meuservidor.com.br
nameserver 200.230.224.1
nameserver 200.230.224.2
```

Cada distribuição organiza de maneira própria os scripts de configuração de interfaces de rede. No Slackware, informações para cada interface de rede podem constar no arquivo `/etc/rc.d/rc.inet1.conf`. No Debian, as configurações para cada interface de rede podem ser manualmente determinadas em `/etc/network/interfaces`.

Configuração da interface

Fundamental para o funcionamento da rede é que a interface de rede esteja configurada corretamente. Estando o hardware corretamente preparado – tanto na parte física da rede quanto quanto ao carregamento do módulo referente à interface local – a interface pode ser configurada manualmente através do programa **ifconfig**. Uma interface de rede também pode ser configurada automaticamente pelo sistema na hora do boot, dependendo dos scripts de configuração da distribuição.

O comando **ifconfig** possui muitas opções, mas seu uso fundamental é definir um endereço IP para a interface de rede, por exemplo:

```
ifconfig eth0 192.168.1.2 up
```

À interface **eth0** foi atribuído o IP **192.168.1.2**. Para desfazer as alterações, usa-se **down** no lugar de **up** e os argumentos são desnecessários. A máscara de rede para a interface também pode ser especificada:

```
ifconfig eth0 192.168.1.2 netmask 255.255.255.0 up
```

O **ifconfig** também é usado para inspecionar as configurações de uma interface. Sem argumentos, mostra as configurações de todas as interfaces ativas do sistema.

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:D0:09:76:B7:6C
          inet addr:200.228.60.237  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:920058 errors:0 dropped:0 overruns:0 frame:0
          TX packets:826694 errors:0 dropped:0 overruns:0 carrier:0
          collisions:434 txqueuelen:1000
          RX bytes:397187958 (378.7 Mb)  TX bytes:471065517 (449.2 Mb)
          Interrupt:9 Base address:0xde00
```

Configuração de Rotas

O comando **route** mostra e cria rotas de rede.

Exemplo de rotas em um sistema:

```
# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
200.228.60.0    0.0.0.0         255.255.255.0  U        0      0      0 eth0
127.0.0.0       0.0.0.0         255.0.0.0      U        0      0      0 lo
0.0.0.0         200.228.60.1   0.0.0.0        UG       0      0      0 eth0
```

O campo **Flags** mostra o estado de funcionamento da rota, podendo conter os seguintes caracteres:

U → Rota ok

H → O alvo é um host

G → É a rota Gateway

R → Restabelecer rota por roteamento dinâmico

D → Rota estabelecida dinamicamente por daemon ou redirecionamento

M → Modificada por daemon ou redirecionada

! → Rota rejeitada.

Exemplo: Cria uma rota na interface eth0, para a rede 192.168.1.0, usando máscara de rede 255.255.255.0:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
```

O termo dev pode ser omitido.

Exemplo: Cria uma rota padrão para o host 200.228.60.1:

```
# route add default gw 200.228.60.1
```

Exemplo: Remove a rota padrão para o host 200.228.60.1:

```
# route del default gw 200.228.60.1
```

Clientes DHCP

O sistema DHCP permite que uma interface seja configurada automaticamente pelo servidor. O daemon **dhcpcd** envia uma requisição para a rede através da interface especificada e o servidor responde com informações de endereço IP, máscara de rede, broadcast, etc, que serão usadas para configurar a interface local. Se o nome da interface não for especificado, eth0 será usado.

Os arquivos de configuração do dhcpcd são armazenados em /var/lib/dhcpc. Informações para cada interface utilizando dhcp são armazenadas em arquivos nesse diretório. O PID para dhcpcd é armazenado em /var/run/dhcpcd-interface.pid. Interface é o nome da interface à qual o cliente dhcpcd está vinculado.

Comandos de Configuração e Inspeção

hostname

Mostra ou altera o nome de host do sistema

domainname

Mostra ou altera o nome de domínio do sistema

dnsdomainname

Mostra o nome de domínio do DNS do sistema

ping

Envia pacotes ICMP ECHO_REQUEST para o host especificado e aguarda retorno de ECHO_REPLY.

Exemplo:

```
# ping -c 3 www.gnu.org
PING gnu.org (199.232.41.10) 56(84) bytes of data.
64 bytes from 199.232.41.10: icmp_seq=1 ttl=52 time=1273 ms
64 bytes from www.gnu.org (199.232.41.10): icmp_seq=2 ttl=52 time=325 ms
64 bytes from www.gnu.org (199.232.41.10): icmp_seq=3 ttl=52 time=567 ms

--- gnu.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 10617ms
rtt min/avg/max/mdev = 325.392/722.193/1273.725/402.319 ms, pipe 2
```

host

Converte nomes para IPs e vice-versa, através de pesquisa no servidor DNS.

Exemplo:

```
# host www.gnu.org
www.gnu.org is an alias for gnu.org.
gnu.org has address 199.232.41.10
;; reply from unexpected source: 200.212.223.100#53, expected 200.230.210.6#53
;; Warning: ID mismatch: expected ID 1737, got 56531
www.gnu.org is an alias for gnu.org.
www.gnu.org is an alias for gnu.org.
gnu.org mail is handled by 30 mx30.gnu.org.
gnu.org mail is handled by 10 mx10.gnu.org.
gnu.org mail is handled by 20 mx20.gnu.org.
```

traceroute

Mostra as rotas percorridas por um pacote até chegar ao seu destino. Limitando o campo TTL (Time To Live) dos pacotes, traceroute recebe respostas de erro ICMP TIME_EXCEEDED de cada host percorrido.

tcpdump

Retorna os cabeçalhos de pacotes numa determinada interface de rede, opcionalmente utilizando filtros.

netstat

Mostra as tabelas de rotas, as conexões ativas e estatísticas relacionadas.

Exemplo: Mostrar as tabela de rotas do kernel, sem resolver nomes de IPs:

```
# netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
200.228.60.0    0.0.0.0         255.255.255.0   U        0  0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        0  0        0 lo
0.0.0.0         200.228.60.1   0.0.0.0         UG        0  0        0 eth0
```

Exemplo: Mostrar todas conexões ativas do tipo TCP, sem resolver nomes de IPs:

```
# netstat -nta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:4662            0.0.0.0:*               LISTEN
tcp    0      0 0.0.0.0:631            0.0.0.0:*               LISTEN
tcp    0 13032 200.228.60.237:55022    84.120.229.63:4662     ESTABLISHED
tcp    0      0 200.228.60.237:4662    87.235.107.221:4314    ESTABLISHED
tcp    0      0 200.228.60.237:50648    207.46.27.21:1863     ESTABLISHED
tcp    0      0 200.228.60.237:4662    200.114.228.210:2243   ESTABLISHED
tcp    0      0 200.228.60.237:4662    62.43.114.0:2121      ESTABLISHED
tcp    0 13068 200.228.60.237:4662    83.58.240.201:22493    ESTABLISHED
tcp    0      0 200.228.60.237:49824    207.46.6.46:1863      ESTABLISHED
```

Exemplo: Mostrar as estatísticas de transmissão para todas interfaces:

```
# netstat -i
Kernel Interface table
Iface    MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500  0 1023387      0      0      0  941820      0      0      0 BMNRRU
lo      16436  0   30881      0      0      0   30881      0      0      0  LRU
```

Os campos RX correspondem aos pacotes recebidos e TX aos pacotes enviados.

Objetivo 1.112.4: Configurar o Linux como um Cliente PPP

Peso: 3

O Protocolo Ponto-a-Ponto (PPP) permite estabelecer uma conexão de rede através de uma interface serial. É o tipo de protocolo usado em conexões via linha telefônica. Os dispositivos necessários para criar uma conexão PPP são interface serial e modem. Esses dispositivos podem se apresentar separadamente – modem externo ligado à porta serial – ou serem integrados – placa interna ISA ou PCI.

Nesse tipo de transmissão cada byte é enviado à porta serial bit a bit, numa frequência denominada **baud rate**. Estando o modem corretamente instalado no sistema, é possível enviar comandos como ATZ e ATDT diretamente para sua respectiva porta serial. O programa `minicom` age como um terminal de comunicação com o modem, podendo realizar diagnósticos e discar números.

Outro programa comum para comunicação com o modem é o `wvdial`. Através do comando `wvdialconf` o modem é automaticamente localizado nas portas seriais e o arquivo de configuração `/etc/wvdial.conf` é criado. Neste arquivo são adicionadas as informações sobre o telefone do provedor de acesso e dados de autenticação do usuário. A discagem é realizada invocando `wvdial nome-provedor`.

Realizada a discagem e a autenticação, o daemon **pppd** deverá ser apropriadamente iniciado para controlar a conexão. Diferente do `minicom`, o `wvdial` executa automaticamente o `pppd`.

É possível invocar a conexão diretamente pelo `pppd`, através da utilização de um script `chat`. Um script `chat` é uma sequência de instruções no formato “*condição esperada* → *resposta*” usadas para a comunicação com o servidor.

Exemplo de Script Chat `/etc/ppp/chat/script:`

```
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
'' 'ATZ'
'OK' 'ATDT 99999999'
'CONNECT' ''
'ogin:' 'adamastor'
'sword:' 'qwerty'
TIMEOUT' '5'
```

O `pppd` pode então ser invocado da seguinte forma:

```
# pppd /dev/ttyS3 115200 \
nodetach lock debug crtscts asyncmap 0000000 \
connect "/usr/sbin/chat -f /etc/ppp/chat/script"
```

Por comodidade, os argumentos do `pppd` podem ser salvos no arquivo `/etc/ppp/options`.

O `pppd` estabelecerá então a conexão PPP. Um número IP será atribuído à interface `pppN` através do script `/etc/ppp/ip-up`. Quando terminada a conexão, a interface será desligada através do script `/etc/ppp/ip-down`.

O diretório `/etc/ppp/peers/` contém arquivos de perfil para possibilitar a conexão à diferentes provedores como diferentes usuários. Dessa forma também é possível que usuários comuns realizem conexões via `pppd`. É necessário que hajam entradas correspondentes no arquivo `/etc/ppp/chap-secrets` ou `/etc/ppp/pap-secrets`. O `pppd` poderá então ser invocado na forma `pppd call nome-peer`.

Tópico 113: Serviços de Rede

Objetivo 1.113.1: Configurar e Administrar o inetd, xinetd e Serviços Relacionados

Peso: 4

Serviços de rede podem rodar continuamente como aplicações independentes esperando por conexões em suas respectivas portas e lidando diretamente com os clientes ou podem ser invocados pelo daemon de rede **inetd** ou **xinetd**.

O daemon inetd

Geralmente disparado na inicialização do sistema, espera por conexões em portas específicas. Dessa forma, o daemon específico para cada serviço será iniciado apenas quando o respectivo serviço for solicitado.

O arquivo **/etc/inetd.conf** configura o daemon **inetd**. Cada linha corresponde à configuração para um serviço válido que deve constar em **/etc/services**. A linha de configuração contém os seguintes campos:

service name

Nome de um serviço válido em **/etc/services**

socket type

stream se TCP e dgram se UDP. Outros valores possíveis são raw, rdm e seqpacket.

protocol

Protocolo válido em **/etc/protocols**, como tcp ou udp.

wait/nowait

Especifica se o **inetd** deve esperar ou não o programa servidor retornar para continuar aceitando conexões para o mesmo.

user.group

Rodar o programa servidor como *user.group*. Dessa forma, é possível rodar o programa servidor com permissões diferentes de root. O grupo é opcional.

server program

Caminho do programa para executar quando um pedido existir no respectivo socket. Para controle dos pedidos, deve ser **/usr/sbin/tcpd**.

server program arguments

Quando **tcpd** é usado para controlar os pedidos, neste campo deverá constar o caminho para o programa que de fato é o servidor do serviço.

Exemplo – Linha de /etc/inetd.conf para o servidor de email pop3:

```
pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/popa3d
```

Para desativar o uso de um servidor, basta comentá-lo com o caracter “#”:

```
# pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/popa3d
```

Após alterar o arquivo `/etc/inetd.conf`, é necessário fazer com que o daemon `inetd` releia as configurações, o que pode ser feito reiniciando o daemon ou enviando o sinal `SIGHUP` através do comando `kill`. O PID para o daemon `inetd` pode ser consultado através do arquivo `/var/run/inetd.pid`.

O daemon `xinetd`

Versão aprimorada do servidor `inetd`. O daemon `tcpd` não é mais necessário para controlar os pedidos, que é feito pelo próprio `xinetd`. A configuração é feita através do arquivo `/etc/xinetd.conf` ou através de arquivos correspondentes a cada serviço em `/etc/xinetd.d/`.

Os valores de configuração para cada serviço são como os do `/etc/inetd.conf`, porém o formato do arquivo difere.

Se iniciado com a opção `-inetd_compat`, o `xinetd` adicionalmente usará as configurações em `/etc/inetd.conf` (se existirem).

Estrutura de configuração de um serviço em `/etc/xinetd.conf`:

```
nome do serviço {
  disable = yes/no
  socket_type = stream,dgram,raw,rdm ou seqpacket
  protocol = Protocolo válido em /etc/protocols
  wait = yes/no
  user = Usuário de início do servidor
  group = Grupo de início do servidor
  server = Caminho para o programa servidor do serviço solicitado
}
```

Como para o daemon `inetd`, é necessário fazer com que o daemon `xinetd` leia novas configurações, o que pode ser feito reiniciando o daemon ou enviando o sinal `SIGHUP` através do comando `kill`.

O log para o `xinetd` é armazenado no arquivo `/etc/xinetd.log`.

Configuração de Serviços

Cada serviço controlado por `inetd/xinetd` é configurado separadamente e de diferentes formas. Um exemplo desses serviços é o FTP – File Transfer Protocol – cujo um dos muitos programas servidores que pode ser utilizado é o `vsftp` – Very Secure FTP.

O arquivo de configuração do `vsftp` é o `/etc/vsftp.conf`. Nele são determinados vários aspectos de funcionamento do daemon `vsftp`, como controle da acessos anônimos. Cada linha encerra uma opção no formato `opção=valor`. Uma opção importante é `listen`, que determina se o `vsftp` deve ou não rodar separadamente do `inetd/xinetd`. Para ser disparado por `inetd/xinetd`, deve ser `listen=no` (que é o padrão).

O daemon deverá constar corretamente no arquivo de configuração do `inetd/xinetd`.

em `/etc/inetd.conf`:

```
ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/vsftpd
```

Ou em `/etc/xinetd.conf`:

```
ftp {
    disable = no
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /usr/sbin/vsftpd
}
```

Se as configurações foram alteradas, os daemons deverão reler os respectivos arquivos para que as alterações tenham efeito.

Controle de Pedidos

Tanto quanto o `inetd` (através do `tcpd`) quanto o `xinetd` permitem o uso de regras para aceitar ou recusar pedidos de serviços feitos por determinados hosts na rede. Essas regras de controle são chamadas *tcpwrappers* e são configuradas através dos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`.

O arquivo `/etc/hosts.allow` contém as regras para os hosts que poderão acessar a máquina local. Se um host corresponder a uma regra em `/etc/hosts.allow`, o mesmo será liberado e o arquivo `/etc/hosts.deny` não será consultado.

O arquivo `/etc/hosts.deny` contém as regras para os hosts que não poderão acessar a máquina local. Se um host não constar em `/etc/hosts.allow` nem em `/etc/hosts.deny`, o mesmo será liberado.

Cada regra é escrita em uma linha e o formato é o mesmo tanto para `/etc/hosts.allow` quanto para `/etc/hosts.deny`:

```
serviço : host : comando
|
|           |           |
|           |           |--> Executar comando no
|           |           |   caso de cumprimento
|           |           |   da regra (opcional)
|           |           |
|           |--> Um ou mais endereços ou
|           |   instruções especiais
|
|--> Um ou mais nomes de
    daemon de serviço ou
    instruções especiais
```

Hosts podem vir na forma de domínios, IPs de rede ou IPs incompletos. Caracteres coringa “?” e “*” podem ser utilizados.

Instruções especiais são ALL, LOCAL, KNOW, UNKNOWN E PARANOID. O operador EXCEPT exclui um host ou grupo de hosts de uma determinada regra.

Em `/etc/hosts.allow`, liberar todos os serviços a todos os hosts no domínio **no-ip.org** com exceção do host **castor**:

```
ALL: .no-ip.org EXCEPT castor.no-ip.org
```

Bloquear todos os serviços a todo host que não constar em regra de `/etc/hosts.allow`, em `/etc/hosts.deny`:

```
ALL: ALL
```

A documentação completa para a criação de regras pode ser encontrada na página manual `hosts_access(5)`.

Objetivo 1.113.2: Operação e Configuração Fundamental de MTA

Peso: 4

O programa responsável por administrar o envio e o recebimento de mensagens de correio eletrônico, local e remotamente, é chamado MTA – Mail Transport Agent. Há várias opções de MTAs, dentre as quais o *sendmail*, o *postfix*, o *qmail* e o *exim*. O MTA roda como um daemon do sistema, geralmente monitorando a porta 25 (SMTP).

O MTA mais utilizado é o *sendmail*, e os demais geralmente oferecem opções de interoperabilidade com ele, salvo as peculiaridades de cada um.

sendmail

O principal arquivo de configuração do *sendmail* é o `/etc/mail/sendmail.cf` (pode estar em `/etc/sendmail.cf`). Por ser deveras complicado editá-lo diretamente, o comando `m4` é usado para gerar o arquivo a partir de um arquivo matriz, de configurações mais simples. Esses arquivos matriz geralmente acompanham o *sendmail*, e seus nomes são terminados em `.mc`. Nesses arquivos pode ser configurado o nome do servidor, opções de redirecionamento, etc.

Exemplo de um arquivo matriz .mc:

```
FEATURE(`use_cw_file')dnl
FEATURE(`use_ct_file')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
FEATURE(`access_db',`hash -T<TMPF> /etc/mail/access')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`local_procmail',`,`',`procmail -t -Y -a $h -d $u')dnl
FEATURE(`always_add_domain')dnl
FEATURE(`redirect')dnl
```

O arquivo de configuração do *sendmail* pode então ser gerado através do comando:

```
# m4 arquivo.mc > /etc/mail/sendmail.cf
```

O *sendmail* depende de arquivos e diretórios de suporte para funcionar. Os caminhos podem ser alterados no arquivos de configuração do *sendmail*, mas geralmente encontram-se nos locais padrão:

`/etc/mail/access`

Lista de hosts autorizados a enviar email por este servidor.

`/etc/mail/aliases` ou **`/etc/aliases`**

Vincula nomes diferentes para destinatários no sistema. Após alterar este arquivo, é necessário executar o comando `newaliases` para gerar o arquivo `/etc/mail/aliases.db` e só então os aliases serão utilizados pelo *sendmail*.

Exemplo de `/etc/mail/aliases`:

manager:	root
dumper:	root
webmaster:	root
abuse:	root

~/ .forward

Pode conter um ou mais endereços para os quais os emails recebidos pelo usuário em questão serão direcionados.

/var/spool/mail/

Diretório onde são armazenados os emails após serem recebidos pelo sendmail. É criado um arquivo para cada usuário, que será lido pelo cliente de email do usuário.

/var/spool/mqueue/

Diretório de fila para os emails enviados pelos usuários do sistema.

Objetivo 1.113.3: Operação e Configuração Fundamental do Apache

Peso 4

O Apache é o servidor Web (http) mais utilizado no mundo. O daemon responsável pelo apache é o **/usr/sbin/httpd**. Os arquivos de configuração encontram-se em **/etc/apache/**, e dentre eles o mais importante é o **/etc/apache/httpd.conf**. Outros arquivos, embora em desuso, ainda podem ser utilizados, como o **/etc/apache/access.conf** e **/etc/apache/srm.conf**.

Algumas configurações fundamentais em /etc/apache/httpd.conf:

ServerType *valor*

Define se o httpd deve rodar separado ou através do daemon inetd. Valor pode ser **standalone** ou **inetd**.

ServerRoot *caminho*

Define o topo do caminho de diretório onde estão os arquivos de configuração, erro e log do apache.

PidFile *caminho*

Define o arquivo que armazenará o valor do PID para o processo httpd. O padrão é **/var/run/httpd.pid**.

ServerAdmin *email*

Endereço de email do administrador do servidor, para onde deverão ser encaminhadas informações de erro.

DocumentRoot *caminho*

Caminho do diretório que armazena os documentos disponibilizados no site. Geralmente **/var/www/html/** ou **/var/www/htdocs**.

O servidor pode ser iniciado, terminado ou reiniciado através do comando **apachectl**, usando as seguintes opções:

`start` → Inicia o servidor.

`stop` → Termina o servidor.

`restart` → Reinicia ou inicia o servidor se houverem alterações de configuração.

`graceful` → Reinicia ou inicia o servidor se houverem alterações de configuração, mas antes espera as conexões ativas terminarem.

`configtest` → Verifica se há erros de sintaxe nas configurações.

Os arquivos de log registram todas as transações realizadas pelo Apache. São eles `/var/log/apache/error_log` e `/var/log/apache/access_log`.

Objetivo 1.113.4: Administração Adequada dos Demons NFS e SAMBA

Peso: 4

NFS

Através do NFS – Network File System – é possível montar diretórios compartilhados remotos como se fossem dispositivos locais. O NFS precisa estar habilitado no kernel, seja nativo ou carregado como módulo, tanto no servidor quanto no cliente.

Para montar dispositivos remotos, é necessário que o daemon `/sbin/rpc.portmap` esteja ativo no cliente e no servidor. Sua execução é controlada através do script `/etc/init.d/portmap` `start|stop|restart` ou `/etc/rc.d/rc.portmap` `start|stop|restart`.

Para que um host possa oferecer diretório através do NFS, é necessário ativar os demons:

`/usr/sbin/rpc.rquotad`

`/usr/sbin/rpc.nfsd`

`/usr/sbin/rpc.mountd`

`/usr/sbin/rpc.lockd`

`/usr/sbin/rpc.statd`

A execução de todos esses demons é unificada através do script `/etc/rc.d/init.d/nfs` `start|stop|restart` ou `/etc/rc.d/rc.nfsd` `start|stop|restart`.

Os compartilhamentos são configurados através do arquivo `/etc/exports`. Cada linha contém um diretório compartilhado seguido de uma lista, separada por espaços, dos hosts que poderão montá-lo. Cada host pode estar acompanhado de parênteses imediatamente a sua direita, sem que haja espaço depois dele, contendo opções de acesso para o mesmo.

Exemplo de compartilhamento em `/etc/exports`:

```
/mnt/hdb1 192.168.1.0/26(ro)
```

O diretório local `/mnt/hdb1` poderá ser montado por todos os hosts da subrede local `192.168.1.0/26`, apenas como somente leitura. Uma opção de acesso importante é `no_root_squash`, que permite que o usuário remoto de ID 0 (root) monte o compartilhamento. Essa opção é especialmente útil quando o diretório local compartilhado é o diretório raiz no cliente remoto. Há várias opções de controle acesso que podem ser consultadas através de `man 5 exports`.

Para atualizar as alterações feitas ao arquivo `/etc/exports` no servidor NFS ativo, é necessário executar o comando **exportfs -a**. Para desativar os compartilhamentos em `/etc/exports`, executar **exportfs -ua**.

No cliente, o próprio comando `mount` é usado para montar o diretório remoto. Por exemplo, montar o diretório compartilhado no exemplo anterior num host da subrede local 192.168.1.0/26:

```
# mount -t nfs 192.168.1.1:/mnt/hdb1 /mnt/remoto
```

Este exemplo presume ser 192.168.1.1 o IP do servidor do compartilhamento e existir no cliente o diretório alvo `/mnt/remoto`.

O comando `nfsstat` mostra estatísticas de uso dos compartilhamentos NFS no servidor.

SAMBA

Máquinas rodando Linux podem acessar e fornecer recursos compartilhados a máquinas rodando MS-Windows®, como arquivos e impressoras.

Os daemons responsáveis são `/usr/sbin/smbd` (cliente/servidor SMB) e `/usr/sbin/mbd` (cliente/servidor NetBIOS). São geralmente disparados pelos scripts de inicialização do sistema. As configurações de acesso e compartilhamento do SAMBA são feitas no arquivo `smb.conf`, geralmente em `/etc/samba/` ou em `/etc`.

Exemplo básico de arquivo `smb.conf`:

```
[global]
    server string = Slackware Samba Server
    log file = /var/log/samba.%m
    max log size = 50
    dns proxy = No

[homes]
    comment = Home Directories
    read only = No
    browseable = No

[printers]
    comment = All Printers
    path = /var/spool/samba
    printable = Yes
    browseable = No

[Montados]
    comment = Dispositivos Montados no servidor
    path = /mnt
```

Há três seções básicas no arquivo `smb.conf`:

[global]

Define o comportamento geral do samba, como o nome do grupo de trabalho, redes permitidas, tipo de acesso aos recursos, etc.

[homes]

Permite que cada usuário acesse seu diretório pessoal no servidor. Pode conter opções restritivas e outras. Conseqüentemente, este compartilhamento pressupõe que o usuário logado

no cliente tenha uma conta no servidor.

[printers]

Compartilha as impressoras instaladas no servidor com os clientes. O funcionamento da impressora no servidor não depende de configuração no samba.

Demais compartilhamentos podem ser criados através de seções específicas para cada um. Diretórios e impressoras podem ser compartilhados e configurados individualmente dessa forma.

O SAMBA pode ser configurado através de uma interface WEB chamada **swat**. Para poder usar o swat, é necessário que o daemon `inetd/xinetd` esteja ativo e o swat liberado em `/etc/services`. A porta do swat é a **901**. Para acessá-lo, portanto, basta direcionar um navegador no servidor para **http://localhost:901**.

Os recursos compartilhados numa máquina MS-Windows© podem ser acessados usando o comando **smbclient**.

Listar recursos disponíveis ao usuário "administrador" numa máquina MS-Windows© 2000, através do smbclient:

```
$ smbclient -L pc-w2k -U administrador
Domain=[PC-W2K] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]

      Sharename      Type      Comment
      -----      -
IPC$                IPC       IPC remoto
Fonts               Disk     Fontes True Type
DOCS                Disk     Meus Documentos
ADMIN$              Disk     Administração remota
C$                  Disk     Recurso compartilhado padrão
Domain=[PC-W2K] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]

      Server          Comment
      -----
Workgroup            Master
-----
```

Montar o diretório compartilhado da máquina pc-w2k:

```
# mount -t smbfs //pc-w2k/fonts /usr/share/fonts -o username=adminstrador
```

ou

```
# smbmount //pc-w2k/fonts /usr/share/fonts -o username=adminstrador
```

A entrada em `/etc/fstab` para essa montagem poderia ser:

```
//pc-w2k/fonts /usr/share/fonts smbfs credentials=/etc/passwd-pc-w2k 0 0
```

A opção `credentials` especifica um arquivo protegido contendo o usuário e a senha para montar o diretório compartilhado. Dessa forma evita-se deixar a senha à mostra em `/etc/fstab`.

Um usuário comum poderá montar recursos remotos com `smbclient` ou `mount` se estes comandos tiverem as permissões adequadas (SUID → 1000,1000).

Objetivo 1.113.5: Configurar um Serviço Básico de DNS

Peso: 4

Em redes pequenas, é possível que os hosts resolvam (convertam) os nomes de máquinas para números IP e vice-versa apenas através do arquivo `/etc/hosts`.

Exemplo de /etc/hosts:

```
127.0.0.1          localhost
192.168.1.1       slack102
192.168.1.6       debian
192.168.1.20      pc-w2k
```

DNS

Para grandes redes, no entanto, é muito mais prático e até necessário o uso de um servidor DNS – Domain Name System – que converte remotamente nomes de máquinas para seus respectivos números IP e vice-versa. A correspondência entre o nome e o número IP é chamada mapeamento, e é organizado de forma hierárquica.

Em outras palavras, um domínio como **howtos.linux.com** será quebrado e resolvido começando por **com**, depois **linux** e finalmente **howtos**, itens chamados respectivamente *top-level domain*, *second-level domain* e *third-level domain*. É nessa ordem que o endereço IP para *howtos.linux.com* será obtido. Os top-level domains mais comuns são *.com*, *.org* e *.net*, mas há outros.

Servidor DNS

O programa servidor responsável por pelo mapeamento é o `/usr/sbin/named`, que é parte do pacote chamado *BIND*, cujas especificações são definidas pelo *Internet Systems Consortium*. Seu arquivo de configuração é `/etc/named.conf`. Em versões do BIND anteriores à versão 8, o arquivo de configuração chama-se `/etc/named.boot`. Para ilustrar, eis as configurações para um servidor cachê de DNS:

Exemplo de /etc/named.conf:

```
options {
    directory "/var/named";
};

zone "." IN {
    type hint;
    file "root.hints";
};
```

Com essa configuração, o servidor local propriamente não mapeará nenhum nome, mas resgatará os pedidos de servidores principais e os armazenará temporariamente para que seus clientes os acessem mais rapidamente. Zonas convencionais têm seções definidas de forma semelhante (Ex: `zone "localhost IN {...}"`) e é importante que para cada uma delas haja uma zona para DNS reverso (Ex: `zone "0.0.127.in-addr.arpa IN {...}"`). O DNS reverso é responsável por converter números IP para seus respectivos nomes.

Voltando ao exemplo, na seção `options` é indicado onde estão os arquivos de mapeamento. (`directory "/var/named";`). Está definido apenas o mapeamento para ".", que será consultado caso nenhum outro mapeamento seja encontrado (é este o caso do exemplo).

No arquivo `/var/named/root.hints` está uma lista com os servidores de nomes principais da internet, conseguidos através do comando `dig`.

Trecho de /var/named/root.hints:

```
(...)  
D.ROOT-SERVERS.NET.      3600000 IN      A      128.8.10.90  
A.ROOT-SERVERS.NET.      3600000 IN      A      198.41.0.4  
H.ROOT-SERVERS.NET.      3600000 IN      A      128.63.2.53  
C.ROOT-SERVERS.NET.      3600000 IN      A      192.33.4.12  
(...)
```

Após alterar os arquivos de configuração, será necessário reiniciar o daemon `named`.

Cliente DNS

O arquivo `/etc/nsswitch.conf` determina de que maneiras e ordem o host local tentará resolver os nomes de endereços.

Trecho exemplo de /etc/nsswitch.conf:

```
hosts:          files dns  
networks:       files  
  
services:       files  
protocols:      files
```

Os termos que precedem o caracter “:” especificam o tipo dos nomes procurados (*hosts*, *networks*, *etc*). Os termos que sucedem o “:” indicam de que forma o tipo de nome em questão deverá ser resolvido. O termo `files` determina o uso de arquivos locais (como o `/etc/hosts` ou `/etc/networks`) e `dns` determina o uso de um servidor DNS.

O servidor DNS a ser utilizado é especificado através do arquivo `/etc/resolv.conf`.

Exemplo de um arquivo /etc/resolv.conf:

```
nameserver 200.230.1.1  
nameserver 200.230.1.2
```

A entrada fundamental é `nameserver`, que define o servidor DNS. Outros `nameserver` podem ser especificados para serem servidores DNS secundários.

Registro de Domínios

Um nome de domínio é registrado para uso na internet através de autoridades competentes como a Internic - <http://www.internic.net/> e a Fapesp - <http://www.registro.br/>. Após registrado, é necessário fornecer os endereços DNS para onde serão direcionadas as solicitações para o domínio em questão.

Objetivo 1.113.7: Utilização do Shell Seguro (OpenSSH)

Peso: 4

O **OpenSSH** é o substituto para ferramentas de acesso remoto como `telnet`, `rlogin`, `rsh` e `rcp`. No host de destino (que aceitará conexões externas), o daemon `/usr/sbin/sshd` deverá estar ativo para possibilitar que clientes se conectem. O `sshd` geralmente é disparado por um script de início do sistema, situado em `/etc/rc.d/rc.sshd` ou `/etc/init.d/sshd`. O daemon

`sshd` utiliza a porta **22** para esperar por pedidos de conexão. O comando usado pelo cliente para se conectar é o **ssh**. O comportamento do `sshd` pode ser modificado através do arquivo de configuração **/etc/ssh/sshd_config**. Personalizações do comando `ssh` podem ser feitas nos arquivos `/etc/ssh/ssh_config`, `/etc/sshrc` ou apenas por usuário em `~/.ssh/config`. Outros comandos úteis do `ssh` são `scp` (copiar através de `ssh`), `sftp-server` (servidor FTP por `ssh`) e `sftp` (cliente FTP por `ssh`).

Conectar-se como usuário root no host 192.168.1.1:

```
$ ssh root@192.168.1.1
```

Como outros serviços de rede, o `ssh` está sujeito ao controle via *tcpwrappers*, através dos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`. E por tratar-se de uma forma de login de usuário na máquina em questão, é respeitado o bloqueio à usuários comuns imposto pelo arquivo `/etc/nologin`. Se o arquivo `/etc/nologin` existir, apenas o `root` pode entrar no sistema, aos demais usuários é vetado o login e apenas mostrado o conteúdo de `/etc/nologin`.

Existem dois protocolos de chaves usados pelo `ssh`, o **rsa** (protocolo 1) e **dsa** (protocolo 2). Dependendo do protocolo usado, os arquivos de chaves chamar-se-ão `ssh_host_rsa_key` ou `ssh_host_dsa_key`. Para cada chave há uma chave pública usada para autenticação por terceiros, armazenada num arquivo de mesmo nome mais o sufixo `.pub`. Por padrão, os arquivos `ssh_host_rsa_key` e `ssh_host_dsa_key` terão permissão `-rw-----` e seus respectivos arquivos `.pub` `-rw-r--r--`.

Na primeira vez que o cliente `ssh` conecta-se a um host, será perguntado sobre aceitar uma chave pública. Se for aceita, será armazenada em `~/.ssh/known_hosts` e garantirá a confiabilidade da conexão entre os dois hosts. O conteúdo deste arquivo pode ser incluído em `/etc/ssh/known_hosts` para que a chave passe a valer para os demais usuários. Ainda assim será necessário que o usuário forneça sua senha ao host de destino.

Para evitar a necessidade da senha em todo login, pode-se criar um arquivo chamado `authorized_keys` para que o `ssh` realize a autenticação de usuário via chave no lugar de senha. O arquivo `authorized_keys` deve existir no host de destino e pode conter uma ou mais chaves que foram criadas no host de origem através do comando **ssh-keygen**. Para gerar uma chave *dsa* de 1024 bits:

```
$ ssh-keygen -t dsa -b 1024
```

Esse comando gerará as chaves `id_dsa` e `id_dsa.pub` em `~/.ssh/` no host de origem. O conteúdo de `id_dsa.pub` poderá então ser incluído em `~/.ssh/authorized_keys` para o usuário específico no host de destino. Supondo ser 192.168.1.1 o IP do host de destino e lá o usuário possuir conta com o mesmo nome, o arquivo pode ser copiado através do comando:

```
$ scp ~/.ssh/id_dsa.pub 192.168.1.1:~/.ssh/authorized_keys
```

Por questão de segurança, é importante que todos os arquivos contendo chaves em `/etc/ssh/` e `~/.ssh/` tenham permissão 600 – escrita e leitura apenas para o dono do arquivo.

Tópico 114: Segurança

Objetivo 1.114.1: Tarefas Administrativas de Segurança

Peso: 4

TCP wrappers

Daemons de serviços de rede compilados com suporte à biblioteca *libwrap* podem utilizar-se do mecanismo chamado TCP wrappers para controlar o acesso por hosts na rede. Esse controle é estabelecido através de regras criadas nos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`.

O arquivo `/etc/hosts.allow` contém as regras para os hosts que poderão acessar a máquina local. Se um host corresponder a uma regra em `/etc/hosts.allow`, o mesmo será liberado e o arquivo `/etc/hosts.deny` não será consultado.

O arquivo `/etc/hosts.deny` contém as regras para os hosts que não poderão acessar a máquina local. Se um host não constar em `/etc/hosts.allow` nem em `/etc/hosts.deny`, o mesmo será liberado.

Cada regra é escrita em uma linha e o formato é o mesmo tanto para `/etc/hosts.allow` quanto para `/etc/hosts.deny`:

```
serviço : host : comando
|       |       |
|       |       |--> Executar comando no
|       |       |   caso de cumprimento
|       |       |   da regra (opcional)
|       |       |
|       |       |--> Um ou mais endereços ou
|       |       |   instruções especiais
|       |       |
|--> Um ou mais nomes de
|       |       |   daemon de serviço ou
|       |       |   instruções especiais
```

Hosts podem vir na forma de domínios, IPs de rede ou IPs incompletos. Caracteres coringa “?” e “*” podem ser utilizados.

Instruções especiais são **ALL**, **LOCAL**, **KNOW**, **UNKNOWN** E **PARANOID**. O operador **EXCEPT** exclui um host ou grupo hosts de uma determinada regra.

Em `/etc/hosts.allow`, liberar todos os serviços a todos os hosts da rede 192.168.1.0 com exceção do 192.168.1.20:

```
ALL: 192.168.1.* EXCEPT 192.168.1.20
```

Bloquear todos os serviços a todo host que não constar em regra de `/etc/hosts.allow`, em `/etc/hosts.deny`:

```
ALL: ALL
```

A documentação completa para a criação de regras pode ser encontrada na página manual `hosts_access(5)`.

SUID/SGID

Arquivos com permissão SUID e SGID garantem privilégios especiais a quem os executa. Portanto, é importante monitorar quais arquivos detêm essas permissões para evitar que programas estranhos ou alterações nos programas conhecidos com essa permissão possam possibilitar a invasão ou dano ao sistema.

Encontrar arquivos SUID e SGID com o find:

```
# find / -perm -4000 -or -perm -2000
/bin/su
/bin/ping
/bin/mount
/bin/ping6
/bin/umount
(...)
```

Para gerar uma lista detalhada, passar a saída para o comando `ls`:

```
# find / \( -perm -4000 -or -perm -2000 \) -exec ls -l '{}' \;
-rws--x--x 1 root bin 35780 2004-06-21 16:20 /bin/su
-rws--x--x 1 root bin 29364 2005-09-07 17:46 /bin/ping
-rwsr-xr-x 1 root bin 61308 2005-09-13 01:42 /bin/mount
-rws--x--x 1 root bin 26764 2005-09-07 17:46 /bin/ping6
-rwsr-xr-x 1 root bin 32212 2005-09-13 01:42 /bin/umount
(...)
```

Essa lista pode ser salva diariamente (provavelmente por um agendamento no crontab) através do comando:

```
# find / \( -perm -4000 -or -perm -2000 \) \
-exec ls -l '{}' \; > /var/log/setuid-$(date +%F)
```

Que gerará um arquivo de nome `setuid-ano-mês-dia`, que poderá ser comparado ao arquivo do dia anterior através do comando `diff`:

```
# diff /var/log/setuid-2006-05-02 /var/log/setuid-2006-05-03
2c2
< -rws--x--x 1 root bin 29364 2005-09-07 17:46 /bin/ping
---
> -rws--x--x 1 root bin 29974 2005-09-07 17:46 /bin/ping
```

Essa saída mostra que o tamanho do arquivo `/bin/ping` mudou de tamanho em relação ao registro anterior. Supõe-se que tenha sido substituído por um programa malicioso, devendo ser excluído e reinstalado adequadamente. É importante rastrear os logs do sistema atrás de possíveis origens dessa alteração.

Outras buscas por brechas no sistema:

Procurar por arquivos com permissão de escrita para todos usuário, com exceção do diretório /dev:

```
# find / -path /dev -prune -perm -2 -not -type l
```

Arquivos de configuração do sistema poderiam ser alterados com intuito de viabilizar invasões ou danos ao sistema.

Procurar por arquivos sem dono ou sem grupo:

```
# find / \( -nouser -o -nogroup \)
```

A existência de arquivos sem dono ou sem grupo indica que o sistema pode ter sido invadido.

Verificação de pacotes

Semelhante à verificação nativa de pacotes `.deb` e `.rpm`, é possível verificar a maioria dos pacotes compilados ou de códigos fonte fornecidos por um desenvolvedor. As maneiras mais comuns de verificação são a soma MD5 e as assinaturas PGP, cujas ferramentas são disponíveis na maioria das distribuições.

Praticamente todos os pacotes de programas tradicionais oferecem assinaturas de verificação PGP, como o código fonte do kernel do Linux disponível em `ftp://ftp.kernel.org/pub/`. Além do arquivo `.tar.gz` ou `.bz2`, deve ser copiado um arquivo de mesmo nome acrescido do sufixo `.sign` (ocasionalmente este arquivo aparece com o sufixo `.asc`). Para verificação, primeiro é necessário importar a chave pública referente ao do kernel:

```
# gpg --keyserver wwwkeys.pgp.net --recv-keys 0x517D0F0E
```

Esse procedimento é necessário somente uma vez. A chave pública deve ser obtida somente de fonte confiável, especificada pelo próprio desenvolvedor do programa. Informações sobre obtenção da chave pública do kernel podem ser consultadas em `http://www.kernel.org/signature.html`.

Agora a verificação do arquivo propriamente:

```
# gpg --verify linux-2.6.16.13.tar.bz2.sign linux-2.6.16.13.tar.bz2
gpg: Signature made Ter 02 Mai 2006 19:15:20 BRT using DSA key ID 517D0F0E
gpg: Assinatura correta de "Linux Kernel Archives Verification Key
<ftpadmin@kernel.org>"
```

Essa saída informa que o referido arquivo é autêntico.

De maneira mais simples agem as verificações MD5. Um arquivo com sufixo `.md5` correspondente ao arquivo `.tar.gz` ou `.bz2` contém um número referente ao resultado de cálculo envolvendo os bytes contidos no pacote. Para verificar a soma MD5 do pacote copiado `bluefish-1.0.5.tar.bz2`, através do seu arquivo MD5 correspondente `bluefish-1.0.5.tar.bz2.md5`:

```
# md5sum -c bluefish-1.0.5.tar.bz2.md5
bluefish-1.0.5.tar.bz2: A soma coincide
```

Como no caso das assinaturas PGP, é fundamental que o arquivo MD5 tenha sido copiado de fonte segura, indicada pelo próprio desenvolvedor do programa.

Senhas

As definições sobre a vida útil de senhas e aspectos relacionados são armazenadas no arquivo `/etc/shadow` (quando usado o sistema de senhas *shadow*). Cada linha corresponde a uma conta, em campos separados por “:”, representando:

1. Nome de acesso.
2. Senha criptografada.
3. Dias decorridos entre 1 de janeiro de 1970 e a última alteração da senha.
4. Número de dias até que a senha deva ser alterada.
5. Número de dias após o qual a senha deve ser alterada.
6. Número de dias antes da expiração da senha no qual o usuário será avisado.

7. Número de dias após a expiração da senha que a conta deve ser bloqueada.
8. Dias decorridos entre 1 de janeiro de 1970 e a data em que a conta foi bloqueada.
9. Campo reservado.

Além de alterar senhas, o comando **passwd** também pode alterar essas definições, através das opções:

-x dias

Número máximo de dias que a senha permanecerá válida.

-n dias

Mínimo de dias até que o usuário possa trocar uma senha modificada.

-w dias

Dias anteriores ao fim da validade da senha nos quais será emitido um aviso a respeito.

-i dias

Inatividade, tolerância de dias após a senha ter expirado até que a conta seja bloqueada.

Exemplo - Alterar validades de senha para a conta "ataliba":

```
# passwd -x 30 -n 1 -w 7 -i 7 ataliba
```

A opção **-e** provoca a expiração imediata da senha e **-d** apaga a senha para a conta especificada.

Quando a opção **-g** é usada, a senha do grupo especificado é alterada; seguido de **-r** remove a senha e de **-R** restringe o acesso à todos usuários. Essa tarefa só pode ser realizada pelo super-usuário ou pelo administrador do grupo.

A conta especificada pode ser bloqueada através da opção **-l** e liberada pela opção **-u**. O estado da conta pode ser verificado fornecendo a opção **-S**:

```
# passwd -S ataliba
ataliba P 05/03/2006 1 30 7 7
```

Onde a saída representa:

```
ataliba P 05/03/2006 1 30 7 7
|      |      |      |      |      |
|      |      |      |      |      | `--> Limite de dias de inatividade após a senha
|      |      |      |      |      |         ter expirado até a conta ser bloqueada.
|      |      |      |      |      |
|      |      |      |      |      | `--> Dias de aviso
|      |      |      |      |      |
|      |      |      |      |      | `--> Limite máximo de dias da senha.
|      |      |      |      |      |
|      |      |      |      |      | `--> Limite mínimo de dias da senha.
|      |      |      |      |      |
|      |      |      |      |      | `--> Data da última mudança de senha.
|      |      |      |      |      |
|      |      |      |      |      | `--> "P": Tem senha usável, "NP": Não tem senha,
|      |      |      |      |      |         "L": Conta bloqueada.
|      |      |      |      |      |
|      |      |      |      |      | `--> Login respectivo a conta.
```

Os atributos da senha também podem ser alterados através do comando **chage**, através dos argumentos:

-m dias

Mínimo de dias até que o usuário possa trocar uma senha modificada.

-M dias

Número máximo de dias que a senha permanecerá válida.

-d dias

Número de dias decorridos em relação a 01/01/1970 em que a senha foi mudada. Também pode ser expresso no formato de data local (dia/mês/ano).

-E dias

Número de dias decorridos em relação a 01/01/1970 a partir do qual a conta não estará mais disponível. Também pode ser expresso no formato de data local (dia/mês/ano).

-I dias

Inatividade, tolerância de dias após a senha ter expirado até que a conta seja bloqueada.

-W dias

Dias anteriores ao fim da validade da senha nos quais será emitido um aviso a respeito.

Exemplo – determinar data de bloqueio de uma conta:

```
# chage -E 04/05/2006 ataliba
```

O uso do `chage` é restrito ao super-usuário (`root`). Porém usuários comuns podem usar o `chage` com a opção `-l` para checar as definições de suas respectivas contas:

```
$ chage -l ataliba
Minimum:          1
Maximum:          30
Warning:           7
Inactive:          1
Last Change:      Mai 03, 2006
Password Expires: Jun 02, 2006
Password Inactive: Jun 03, 2006
Account Expires:  Abr 05, 2006
```

Tanto `passwd` quanto `chage` entram em modo de configuração interativa se não forem passadas opções. O usuário assumido será sempre o atual se um usuário não for especificado como argumento.

Atualização de programas

Como nenhum programa é imune à falhas, é recomendado instalar todas as correções disponibilizadas pelo desenvolvedor. Programas desatualizados com falhas conhecidas são alvos fáceis para invasão e possível danificação do sistema.

Todas as principais distribuições mantêm atualizações para seus programas compilados. A anúncio das atualizações é geralmente feito através de mala direta por email, cuja inscrição pode ser realizada no site da distribuição.

Grupos especializados em segurança também informam sobre falhas e procedimentos necessários para correção. O *CERT* (Computer Emergency Response Team) - www.cert.org – e *BUGTRAQ* - www.securityfocus.com – divulgam questões pertinentes à falhas e correções de sistemas.

Filtragem de Pacotes – iptables

A filtragem de pacotes de dados em rede permite controlar o fluxo das transmissões através de regras específicas. Dessa forma é possível criar um *firewall* ou um redirecionamento do tipo NAT (Network Address Translation).

O programa utilizado para criação dessas regras é o **iptables**. É necessário que o kernel em uso seja capaz de trabalhar com filtragem de pacotes, o que é regra nos kernels recentes. O item de configuração do kernel para filtragem de pacotes é “*Network Packet Filtering*”.

Para cada tipo de operação há uma tabela específica. Cada tabela contém *chains* (correntes) onde são definidos *targets* (ações) para os pacotes que corresponderem à determinada regra na corrente. São as tabelas naturais **filter**, **nat** e **mangle**:

filter

É a tabela padrão. Contém as chains embutidas **INPUT** (para pacotes que chegam ao host local), **FORWARD** (para pacotes sendo roteados pelo host local) e **OUTPUT** (para pacotes gerados no host local). Essa é a tabela utilizada para construção de firewalls.

nat

Para pacotes que criam novas conexões. Contém as chains embutidas **PREROUTING**, **OUTPUT** e **POSTROUTING**.

mangle

Para alterações especializadas de pacotes. Contém as chains **INPUT**, **OUTPUT**, **PREROUTING**, **FORWARD** e **POSTROUTING**.

A tabela de atuação é indicada através da opção **-t** do comando `iptables`. Se nenhuma tabela for especificada, a tabela assumida será a *filter*. As operações dentro de uma *chain* são determinadas através de argumentos-comando:

- A** → Adicionar regra na *chain*
- I** → Inserir regra numa posição específica dentro da *chain*
- R** → Substituir regra na *chain*
- D** → Apagar *chain*
- N** → Criar *chain* personalizada
- X** → Apagar *chain* vazia
- P** → Definir política para uma *chain* embutida
- L** → Listar a(s) regra(s) em uma *chain*
- F** → Apagar todas as regras em uma *chain*
- Z** → Zerar os contadores de pacotes em todas as regras de uma *chain*

Especificações de regras (interceptam os pacotes que corresponderem):

-s endereço

Ou **--source** endereço. Endereço de origem do pacote. Pode ser nome de rede, nome de host, IP de rede/máscara de rede ou simplesmente um endereço IP. Se endereço precedido de “!” intercepta os pacotes que não corresponderem à condição.

-d *endereço*

Ou **--destination** *endereço*. Endereço de destino do pacote. Mesmo formato de *-s*. Se endereço precedido de “!” intercepta os pacotes que não corresponderem à condição.

-p *protocolo*

Ou **--protocol** *protocolo*. Define o protocolo. Pode ser tcp, udp, icmp ou all. Se protocolo precedido de “!” intercepta os pacotes que não corresponderem à condição.

-i *interface*

Ou **--in-interface** *interface*. Interface através da qual o pacote chegou. Se o nome interface for seguida do sinal “+” (*interface+*) aplicará a todas interfaces cujos nomes comecem por “*interface*”. Se interface precedido de “!” intercepta os pacotes que não corresponderem à condição. Se *-i interface* não existir, toda interface será assumida.

-o *interface*

Ou **--out-interface** *interface*. Interface através da qual o pacote será enviado. Se o nome interface for seguida do sinal “+” (*interface+*) aplicará a todas interfaces cujos nomes comecem por “*interface*”. Se interface precedido de “!” intercepta os pacotes que não corresponderem à condição. Se interface for omitido, toda interface será assumida.

-j *ação*

Ou **--jump** *ação*. *Targets* (ações) para o(s) pacote(s) interceptados. *Targets* comuns para firewall:

ACCEPT → Permite a passagem normal do pacote

DROP → Descarta o pacote

-m *módulo*

Ou **--match** *módulo*. Usar módulo estendido “*módulo*”. Há muitos tipos de módulos de controle adicionais e opções extras para cada um deles. Um muito usado para firewall é o módulo **state**, cuja opção **--state** estado permite determinar qual a relação de um pacote com as conexões existentes. Possíveis valores para estado são INVALID (o estado não pode ser determinado), ESTABLISHED (o pacote pertence a uma conexão ativa), NEW (indicando que o pacote inicia nova conexão e RELATED (o pacote inicia outra conexão, porém relacionada a uma conexão existente).

Exemplo de criação de firewall simples.

Apagar todas as regras da tabela filter:

```
# iptables -t filter -F
```

Estabelecer política de descartar todos os pacotes em todas chains da tabela filters:

```
# iptables -t filter -P INPUT DROP
# iptables -t filter -P FORWARD DROP
# iptables -t filter -P OUTPUT DROP
```

Liberar todos os pacotes (saindo e entrando) da interface local:

Tópico 114: Segurança

```
# iptables -t filter -A INPUT -i lo -s 0/0 -d 0/0 -j ACCEPT
# iptables -t filter -A OUTPUT -o lo -s 0/0 -d 0/0 -j ACCEPT
```

Liberar todos os pacotes saindo através da interface eth0:

```
# iptables -t filter -A OUTPUT -o eth0 -s 0/0 -d 0/0 -j ACCEPT
```

Liberar para entrar pela interface eth0 somente os pacote pertencentes (ESTABLISHED) ou relacionados (RELATED) a uma conexão existente:

```
# iptables -t filter -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED \
-s 0/0 -d 0/0 -j ACCEPT
```

Listando as novas configurações de filtros:

```
# iptables -t filter -L -v
Chain INPUT (policy DROP 39 packets, 12431 bytes)
  pkts bytes target    prot opt in     out     source         destination
    0    0 ACCEPT    all  --  lo     any     anywhere      anywhere
    0    0 ACCEPT    all  --  eth0   any     anywhere      anywhere
state RELATED,ESTABLISHED

Chain FORWARD (policy DROP 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source         destination

Chain OUTPUT (policy DROP 12 packets, 816 bytes)
  pkts bytes target    prot opt in     out     source         destination
    0    0 ACCEPT    all  --  any    lo     anywhere      anywhere
    0    0 ACCEPT    all  --  any    eth0   anywhere      anywhere
```

Este firewall simples irá descartar qualquer tentativa de conexão por programas remotos, inclusive compartilhadores de arquivos e programas de mensagens instantâneas.

Voltar à configuração padrão (aceite indiscriminado):

```
# iptables -t filter -F INPUT
# iptables -t filter -P INPUT ACCEPT
# iptables -t filter -F FORWARD
# iptables -t filter -P FORWARD ACCEPT
# iptables -t filter -F OUTPUT
# iptables -t filter -P OUTPUT ACCEPT
```

É necessário que se respeite as letras maiúsculas e minúsculas das opções.

Verificando portas abertas no sistema

O programa **nmap** é utilizado para rastrear sistemas em busca de portas de serviços ativas. Seu uso mais simples é sem argumentos, especificando apenas um nome ou endereço de host a ser rastreado:

```
$ nmap localhost

Starting nmap 3.93 ( http://www.insecure.org/nmap/ ) at 2006-05-08 01:39 BRT
Interesting ports on localhost (127.0.0.1):
(The 1666 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
631/tcp   open  ipp
6000/tcp  open  X11
```

A saída mostra que as portas 631/tcp (Serviço de impressão do cups) e 6000/tcp (servidor de janelas X) estão abertas à conexões. Portanto é fundamental estabelecer restrições ao seu uso através da

configuração do *tcpwrapper* ou através da configuração do próprio daemon do serviço.

O *nmap* possui muitas opções de rastreamento que podem ser consultadas através de sua página manual (`man nmap`). É possível, por exemplo, fazer um rastreamento para tentar descobrir as portas passíveis de conexão e qual o sistema operacional do host em questão:

```
# nmap -sS -O localhost

Starting nmap 3.93 ( http://www.insecure.org/nmap/ ) at 2006-05-08 01:50 BRT
Interesting ports on localhost (127.0.0.1):
(The 1666 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
631/tcp   open  ipp
6000/tcp  open  X11
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.0 - 2.5.20, Linux 2.6.3 - 2.6.7 (X86)

Nmap finished: 1 IP address (1 host up) scanned in 3.240 seconds
```

Programa que desempenha função semelhante é o **netstat**. Entre outras funções, com o *netstat* é possível monitorar as conexões ativas. Algumas opções do *netstat*:

- t → Mostra todas as conexões tcp ativas
- l → Mostra todas as portas abertas à conexões
- c → Execução contínua, renova as informações a cada segundo.

O *netstat* é capaz de obter e mostrar várias outras informações (tabelas de rotas, estatísticas de interface, etc). Mais detalhes sobre sua operação na página manual `man netstat`.

Objetivo 1.114.2: Segurança do Host

Peso: 3

syslog

Em geral, todas as mensagens do serviço de sistema *syslog* são importantes para checar e garantir o bom funcionamento do sistema. Porém, a facilidade *authpriv* é especialmente importante pois é responsável por informar questões relativas à mudança e autenticação de usuários. Configuração de *authpriv* no arquivo `/etc/syslog.conf`:

```
authpriv.*          /var/log/secure
```

Essa entrada fará com que todas as mensagens relativas a *authpriv* sejam armazenadas no arquivo `/var/log/secure`.

É importante que os arquivos de log críticos em `/var/log/*` não possam ser lidos ou escritos por usuários comuns. Devem ter portanto a permissão octal 600 (`-rw-----`).

Algumas mensagens mais graves são enviadas por email para usuário root. Para que outro usuário também receba essas mensagens de segurança, basta acrescentá-lo como um alias de root em `/etc/aliases` (ou `/etc/mail/aliases`):

```
root:      ataliba, palimercio
```

Para que o redirecionamento tenha efeito, deve ser executado o comando `newaliases`.

No exemplo, as mensagens de segurança destinadas a root serão enviadas para os usuários *ataliba* e *palimercio* (que devem ser nomes de contas de usuários existentes no sistema).

Sistema de senhas shadow

O uso do sistema de senhas shadow proporciona maior segurança, visto que o arquivo onde as senhas são armazenadas (`/etc/shadow`) não oferece leitura para usuários comuns (`-rw-r-----`) e essas estão sob forte criptografia.

O uso de senhas shadow é verificado pela letra “x” no campo de senha do usuário em `/etc/passwd`. Caso o sistema não use senhas shadow, é necessário instalar o pacote “shadow password suite” (já presente na grande maioria das distribuições) e executar o comando `pwconv` para converter as senhas antigas para o novo formato.

Desativando serviços de rede

Daemons de serviços de rede que não estão sendo utilizados representam um risco adicional de invasões que pode ser evitado. Uma das maneiras de desativar servidores desnecessários é tirar a permissão de execução do script que os inicia. Primeiro o serviço deve ser terminado e depois executado o comando `chmod -x script_daemon`.

Para serviços disparados pelo servidor `inetd`, basta comentar (acrescentar o caracter “#”) à linha referente ao serviço em `/etc/inetd.conf`.

Servidor telnet desativado em /etc/inetd.conf:

```
#telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

De forma semelhante, os serviços controlados pelo servidor `xinetd` podem ser desativados no arquivo de configuração `/etc/xinetd.conf`, na opção *disable* correspondente ao serviço.

Servidor ftp desativado em /etc/xinetd.conf:

```
ftp {
    disable = yes
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /usr/sbin/vsftpd
}
```

Objetivo 1.114.3: Segurança a Nível de Usuário

Peso: 1

Além do cuidado com as permissões e senhas do usuário, controlados com as ferramentas `passwd`, `usermod`, `umask`, etc, outras medidas podem ser tomadas para aumentar a disponibilidade da máquina, seja ela uma estação ou servidor.

Usuários comuns podem provocar lentidão e até panes no sistema se utilizarem exageradamente recursos da máquina. Semelhante ao controle de espaço em disco exercido através do uso de quotas, limites de memória, criação de arquivos e processos podem ser limitados através do comando `ulimit`.

O `ulimit` é um comando embutido do `bash`. Os limites são válidos para a sessão do shell atual e sessões/processos disparados a partir dela. Geralmente, os limites são estabelecidos no login, através do arquivo `/etc/profile`.

Para cada recurso, pode ser estipulado um limite *soft* e um limite *hard*, especificados pelas opções `-S` e `-H`, respectivamente. O limite *hard* não poderá ser aumentado e o limite *soft* poderá ser aumentado até o limite *hard*. Se não for especificado `-S` ou `-H`, o limite indicado será definido para ambos.

Opções comuns de ulimit:

- `-a` → Mostra os limites atuais
- `-f` → Especifica o número máximo de arquivos que poderão ser criados pelo shell
- `-u` → O número máximo de processos disponíveis ao usuário
- `-v` → O montante máximo de memória virtual disponível ao shell

Estabelecer limite máximo de processos de processos em 100:

```
ulimit -Su 100
```

Permitir que o usuário acresça este limite até o máximo de 200:

```
ulimit -Hu 200
```

Se nenhuma opção for fornecida, o recurso assumido será `-f` (limite de arquivos criados). Sem um valor de limite, `ulimit` mostrará o limite *soft* atual para a opção fornecida.

Apêndice 1

Objetivos detalhados para o exame 102

Estes são os objetivos oficiais para o exame 102, disponíveis através do site <http://www.lpi.org/>. A opção por deixar os objetivos em inglês é para ater-se ao formato original, visto que os objetivos oficiais são publicados apenas nesse idioma.

Exam 102: Detailed Objectives

This is a required exam for LPI certification Level 1. It covers basic system administration skills that are common across all distributions of Linux.

IMPORTANT INFORMATION: These are the current 102 objectives, which are valid **effective** 2006-01-01. The 102 exams which are delivered through test centres all over the world, will reflect these updated objectives for English in early 2006 and all other languages by late 2006/early 2007. Candidates should be prepared to take exams based upon these objectives after 1Q2006.

Each objective is assigned a weighting value. The weights range roughly from 1 to 10 and indicate the relative importance of each objective. Objectives with higher weights will be covered in the exam with more questions.

Maintainer: [Dimitrios "Taki" Bogiatzoules, Product Developer](#)

Last modification: 2005-12-31

Topic 105: Kernel

- **1.105.1 Manage/Query kernel and kernel modules at runtime**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to manage and/or query a kernel and kernel loadable modules.

Key knowledge area(s):

Use command-line utilities to get information about the currently running kernel and kernel modules.

Manually load and unload kernel modules.

Determine when modules can be unloaded.

Determine what parameters a module accepts.

Configure the system to load modules by names other than their file name.

The following is a partial list of the used files, terms and utilities:

`/lib/modules/kernel-version/modules.dep`

`/etc/modules.conf`

`/etc/modprobe.conf`

`depmod`

`insmod`

`lsmod`

`rmmod`

`modinfo`

`modprobe`

uname

- **1.105.2 Reconfigure, build and install a custom kernel and kernel modules**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to customize, build and install a kernel and kernel loadable modules from source.

Key knowledge area(s):

Customize the current kernel configuration.

Build a new kernel and appropriate kernel modules.

Install a new kernel and any modules.

Ensure that the boot manager can locate the new kernel and associated files.

The following is a partial list of the used files, terms and utilities:

/usr/src/linux/*

/usr/src/linux/.config

/lib/modules/kernel-version/*

/boot/*

make

make targets: all, config, menuconfig, xconfig, gconfig oldconfig,

modules, install, modules_install, depmod, rpm-pkg, binrpm-

pkg, deb-pkg

Topic 106: Topic 106 Boot, Initialization, Shutdown and Runlevels

- **1.106.1 Boot the system**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to guide the system through the booting process.

Key knowledge area(s):

Give commands to the boot loader and options to the kernel at boot time.

Check boot events in the log files.

The following is a partial list of the used files, terms and utilities:

/var/log/messages

/etc/modules.conf

/etc/modprobe.conf

dmesg

LILO

GRUB

- **1.106.2 Change runlevels and shutdown or reboot system**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to manage the runlevel of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching runlevel and properly terminate processes. This objective also includes setting the default runlevel.

Key knowledge area(s):

Set the default runlevel.

Change between run levels including single user mode.

Shutdown and reboot from the command line.

Alert users before switching runlevels or other major system event.

Properly terminate processes.

The following is a partial list of the used files, terms and utilities:

/etc/inittab

shutdown

init

Topic 107: Printing

- **1.107.2 Manage printers and print queues**

[Comment this objective!](#)

Weight: 1

Description: Candidates should be able to manage print queues and user print jobs.

Key knowledge area(s):

Configure and monitor a print server.

Manage user print queues.

Troubleshoot general printing problems.

The following is a partial list of the used files, terms and utilities:

CUPS configuration files, tools and utilities

/etc/printcap

lpc

lpq

lprm

lp

- **1.107.3 Print files**

[Comment this objective!](#)

Weight: 1

Description: Candidates should be able to manage print queues and manipulate print jobs.

Key knowledge area(s):

Add and remove jobs from configured printer queues.

Convert text files to postscript for printing.

The following is a partial list of the used files, terms and utilities:

CUPS configuration files, tools and utilities

a2ps

lpr

lpq

- **1.107.4 Install and configure local and remote printers**

[Comment this objective!](#)

Weight: 1

Description: Candidates should be able to install and configure local and remote printers.

Key knowledge area(s):

Install a printer daemon.

Install and configure a print filter.

Make local and remote printers accessible for a Linux system, including postscript, non-postscript and Samba printers.

The following is a partial list of the used files, terms and utilities:

CUPS configuration files, tools and utilities

/etc/printcap

/var/spool/cups/

/var/spool/lpd/*/

lpd

Topic 108: Documentation

- **1.108.1 Use and manage local system documentation**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to use and administer the man facility and the material in /usr/share/doc/.

Key knowledge area(s):

Find relevant man pages.

Search man page sections.

Find commands and man pages related to them.

Configure access to man sources and the man system.

Prepare man pages for printouts.

Use the system documentation stored in /usr/share/doc/ and determine what documentation to keep in /usr/share/doc/.

The following is a partial list of the used files, terms and utilities:

MANPATH man

apropos

whatis

- **1.108.2 Find Linux documentation on the Internet**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to find and use Linux documentation on the internet.

Key knowledge area(s):

This objective includes using Linux documentation at sources such as the *Linux Documentation Project* (LDP), vendor and third-party websites, newsgroups, newsgroup archives and mailing lists.

The following is a partial list of the used files, terms and utilities:

Not applicable

- **1.108.5 Notify users on system-related issues**

[Comment this objective!](#)

Weight: 1

Description: Candidates should be able to notify the users about current issues related to the system.

Key knowledge area(s):

Automate communication with users through logon messages.

The following is a partial list of the used files, terms and utilities:

/etc/issue
/etc/issue.net
/etc/motd

Topic 109: Shells, Scripting, Programming and Compiling

- **1.109.1 Customize and use the shell environment**

[Comment this objective!](#)

Weight: 5

Description: Candidates should be able to customize shell environments to meet users' needs.

Key knowledge area(s):

Set environment variables (e.g. PATH) at login or when spawning a new shell.

Write BASH functions for frequently used sequences of commands.

The following is a partial list of the used files, terms and utilities:

Internal BASH functions and commands

~/ .bash_profile
~/ .bash_login
~/ .profile
~/ .bashrc
~/ .bash_logout
~/ .inputrc
function
export
env
set
lists
seq
unset

- **1.109.2 Customize or write simple scripts**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to customize existing scripts, or write simple new BASH scripts.

Key knowledge area(s):

Use standard sh syntax (loops, tests).

Use command substitution.

Test return values for success or failure or other information provided by a command.

Perform conditional mailing to the superuser.

Correctly select the script interpreter through the shebang (#!) line.

Manage the location, ownership, execution and suid-rights of scripts.

The following is a partial list of the used files, terms and utilities:

for
while
test
chmod

Topic 111: Administrative Tasks

- **1.111.1 Manage users and group accounts and related system files**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to add, remove, suspend and change user accounts.

Key knowledge area(s):

Add, modify and remove users and groups.

Manage user/group info in password/group databases.

Correctly manage shadow password/group databases using the appropriate tools.

Create and manage special purpose and limited accounts.

The following is a partial list of the used files, terms and utilities:

/etc/passwd
/etc/shadow
/etc/group
/etc/gshadow
chage
gpasswd
groupadd
groupdel
groupmod
passwd
useradd
userdel
usermod

- **1.111.2 Tune the user environment and system environment variables**

[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to modify global and user profiles.

Key knowledge area(s):

Set environment variables.

Maintain skeleton directories for new user accounts.

Set command search path with the proper directory.

The following is a partial list of the used files, terms and utilities:

/etc/profile
/etc/skel
env
export

set
unset

- **1.111.3 Configure and use system log files to meet administrative and security needs**
[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to configure and manage system logs.

Key knowledge area(s):

Manage the type and level of information logged.

Manually scan log files for notable activity.

Monitor log files.

Automatically rotate and archive log files.

Track down problems noted in log files.

The following is a partial list of the used files, terms and utilities:

/etc/syslog.conf

/var/log/*

logrotate

tail -f

- **1.111.4 Automate system administration tasks by scheduling jobs to run in the future**
[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to use `cron` or `anacron` to run jobs at regular intervals and to use `at` to run jobs at a specific time.

Key knowledge area(s):

Manage `cron` and `at` jobs.

Configure user access to `cron` and `at` services.

The following is a partial list of the used files, terms and utilities:

/etc/anacrontab

/etc/at.deny

/etc/at.allow

/etc/crontab

/etc/cron.allow

/etc/cron.deny

/var/spool/cron/*

at

atq

atrm

crontab

- **1.111.5 Maintain an effective data backup strategy**
[Comment this objective!](#)

Weight: 3

Description: Candidates should be able to plan a backup strategy and backup filesystems automatically to various media.

Key knowledge area(s):

Dump a raw device to a file or vice versa.
Perform partial and manual backups.
Verify the integrity of backup files.
Partially or fully restore backups.

The following is a partial list of the used files, terms and utilities:

cpio
dd
dump
restore
tar

- **1.111.6 Maintain system time**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to properly maintain the system time and synchronize the clock via NTP.

Key knowledge area(s):

Set the system date and time.
Set the BIOS clock to the correct time in UTC.
Configure the correct timezone.
Configure NTP including correcting for clock drift.

The following is a partial list of the used files, terms and utilities:

/usr/share/zoneinfo
/etc/timezone
/etc/localtime
/etc/ntp.conf
/etc/ntp.drift
date
hwclock
ntpd
ntpdate

Topic 112: Networking Fundamentals

- **1.112.1 Fundamentals of TCP/IP**

[Comment this objective!](#)

Weight: 4

Description: Candidates should demonstrate a proper understanding of network fundamentals.

Key knowledge area(s):

Demonstrate an understanding of IP-addresses including but not limited to:
Network masks, for example: determine the network part and the broadcast address for a host based on its address and subnet mask in "dotted quad" or abbreviated notation or determine the network address, broadcast address and netmask when given an IP-address and number of bits in the address that are used to identify the host.
Demonstrate an understanding of the network classes and classless subnets (CIDR) and the

reserved addresses for private network use.

Understand the function and application of a default route.

Understand basic internet protocols (IP, ICMP, TCP, UDP) and the more common TCP and UDP ports (20, 21, 23, 25, 53, 80, 110, 119, 139, 143, 161).

Basic knowledge about the differences between IPV4 and IPV6.

The following is a partial list of the used files, terms and utilities:

/etc/services

ftp

telnet

host

ping

dig

traceroute

whois

- **1.112.3 TCP/IP configuration and troubleshooting**

[Comment this objective!](#)

Weight: 7

Description: Candidates should be able to view, change and verify configuration settings and operational status for various network interfaces.

Key knowledge area(s):

Manually and automatically configure network interfaces and routing tables to include adding, starting, stopping, restarting, deleting or reconfiguring network interfaces.

Change, view, or configure the routing table and correct an improperly set default route manually.

Configure a DHCP client.

Basic TCP/IP host configuration.

Debug problems associated with the network configuration.

The following is a partial list of the used files, terms and utilities:

/etc/HOSTNAME or /etc/hostname

/etc/hosts

/etc/networks

/etc/host.conf

/etc/resolv.conf

/etc/nsswitch.conf

ifconfig

ifup & ifdown

route

dhcpcd

dhclient

pump

host

hostname

domainname

dnsdomainname

netstat

ping

traceroute
tcpdump
Network scripts run during system initialization.

- **1.112.4 Configure Linux as a PPP client**

[Comment this objective!](#)

Weight: 3

Description: Candidates should understand the basics of the PPP protocol and be able to configure and use PPP for outbound connections.

Key knowledge area(s):

Define the chat sequence to connect (given a login example) and the setup commands to be run automatically when a PPP connection is made.

Initialize and terminate a PPP connection, with a modem, ISDN or ADSL with the appropriate scripts.

Set PPP to automatically reconnect if disconnected.

The following is a partial list of the used files, terms and utilities:

```
/etc/ppp/options.*  
/etc/ppp/peers/*  
/etc/wvdial.conf  
/etc/ppp/ip-up  
/etc/ppp/ip-down  
wvdial  
ppp
```

Topic 113: Networking Services

- **1.113.1 Configure and manage xinetd, inetd and related services**

[Comment this objective!](#)

Weight: 4

Description: Configure and manage xinetd, inetd and related services.

Key knowledge area(s):

Configure which services are available through (x) inetd.

Manually start, stop and restart internet services.

Configure basic network services including ssh and ftp.

Set a service to run as another user instead of the default in (x) inetd configuration.

Basic knowledge of tcpwrappers to allow or deny services on a host-by-host basis.

The following is a partial list of the used files, terms and utilities:

```
/etc/hosts.allow  
/etc/hosts.deny  
/etc/services  
/etc/xinetd.conf  
/etc/xinetd.d/  
/etc/xinetd.log  
/etc/inetd.conf
```

- **1.113.2 Operate and perform basic configuration of Mail Transfer Agent (MTA)**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to operate and perform basic configuration of MTA. Advanced custom configurations not included.

Key knowledge area(s):

Modify simple parameters in MTA configuration files.

Create e-mail aliases.

Manage the e-mail queue.

Start and stop MTA.

Configure e-mail forwarding.

Check for and close an open relay on the mailserver.

Perform basic troubleshooting of MTA.

The following is a partial list of the used files, terms and utilities:

Configuration files, documentation and commands for: postfix, qmail, exim and sendmail

/etc/mail/*

~/.forward

sendmail emulation layer commands

newaliases

- **1.113.3 Operate and perform basic configuration of Apache**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to operate and perform basic configuration of Apache. Advanced custom configurations not included.

Key knowledge area(s):

Modify simple parameters in Apache configuration files.

Start and stop httpd and restart after modification to configuration.

Arrange for automatic starting of httpd upon boot.

The following is a partial list of the used files, terms and utilities:

/etc/apache2/

httpd.conf

apache2ctl

apachectl

httpd

- **1.113.4 Properly manage the NFS and SAMBA daemons**

[Comment this objective!](#)

Weight: 4

Description: Candidates should know how to manage the NFS, smb and nmb daemons.

Key knowledge area(s):

Mount remote filesystems using NFS.

Configure NFS for exporting local filesystems.

Start, stop and restart the NFS server.

Install and configure Samba using the included GUI tools (swat) or direct edit of the

/etc/smb.conf file (Note: this deliberately excludes advanced NT domain issues but

includes simple sharing of home directories and printers, as well as correctly setting the nmbd

as a WINS client).

The following is a partial list of the used files, terms and utilities:

/etc/exports
/etc/fstab
/etc/smb.conf
mount
umount

- **1.113.5 Setup and configure basic DNS services**

[Comment this objective!](#)

Weight: 4

Description: Candidates should be able to configure basic DNS services.

Key knowledge area(s):

Configure hostname lookups and troubleshoot problems with local caching-only name server.
Demonstrate an understanding of the domain registration and DNS translation process.
Understanding configuration files for BIND8 and BIND9.

The following is a partial list of the used files, terms and utilities:

/etc/hosts
/etc/resolv.conf
/etc/nsswitch.conf
/etc/named.conf
named

- **1.113.7 Set up secure shell (OpenSSH)**

[Comment this objective!](#)

Weight: 4

Description: The candidate should be able to obtain and configure OpenSSH.

Key knowledge area(s):

Perform basic OpenSSH installation and troubleshooting.
Configure sshd to start at system boot.

The following is a partial list of the used files, terms and utilities:

/etc/hosts.allow
/etc/hosts.deny
/etc/nologin
/etc/ssh/sshd_config
/etc/ssh_known_hosts
/etc/sshrc
sshd
ssh-keygen

Topic 114: Security

- **1.114.1 Perform security administration tasks**

[Comment this objective!](#)

Weight: 4

Description: Candidates should know how to review system configuration to ensure host security in accordance with local security policies.

Key knowledge area(s):

Configure `tcpwrappers`.

Audit a system to find files with the `suid/sgid` bit set.

Verify packages.

Set or change user passwords and password aging information.

Update binaries as recommended by CERT, BUGTRAQ and/or distribution's security alerts.

Demonstrate basic knowledge of `iptables`.

Being able to use `nmap` and `netstat` to discover open ports on a system.

The following is a partial list of the used files, terms and utilities:

`/proc/net/ip_*`

`find`

`iptables`

`passwd`

`socket`

`nmap`

`netstat`

- **1.114.2 Setup host security**

[Comment this objective!](#)

Weight: 3

Description: Candidates should know how to set up a basic level of host security.

Key knowledge area(s):

Configure `syslog` with an eye to security.

Set up and manage shadow passwords.

Set up a e-mail alias for `root`'s e-mail.

Turn off network services not in use.

The following is a partial list of the used files, terms and utilities:

`/etc/xinetd.d/*`

`/etc/xinetd.conf`

`/etc/inet.d/*`

`/etc/inetd.conf`

`/etc/nologin`

`/etc/passwd`

`/etc/shadow`

`/etc/syslog.conf`

- **1.114.3 Setup user level security**

[Comment this objective!](#)

Weight: 1

Description: Candidates should be able to configure user level security. Tasks include limits on user logins, processes and memory usage.

Key knowledge area(s):

Set up limits on user logins, processes and memory usage.

The following is a partial list of the used files, terms and utilities:

quota
usermod
ulimit

Apêndice 2

GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject

Apêndice 2

(or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the

Apêndice 2

Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this

Apêndice 2

License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such

parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.