FORMAÇÃO EM SEGURANÇA CIBERNÉTICA

CADERNO DE ATIVIDADES

Primeira Semana

Rua Lauro Müller, 116 sala 1103

22290-906 Rio de Janeiro, RJ

Diretor Geral

Nelson Simões

Diretor de Serviços e Soluções

José Luiz Ribeiro Filho

Escola Superior de Redes

Coordenação

Luiz Coelho

Equipe ESR (em ordem alfabética)

Celia Maciel, Cristiane Oliveira, Derlinéa Miranda, Edson Kowask, Elimária Barbosa, Evellyn Feitosa, Felipe Nascimento, Lourdes Soncin, Luciana Batista, Renato Duarte, Sérgio Souza e Yve Abel Marcial.

Versão 0.1.1

Índice

Configuração preliminar das máquinas

1) Topologia geral de rede

A figura abaixo mostra a topologia de rede que será utilizada durante este curso. Nos tópicos que se seguem, iremos verificar que a importação de máquinas virtuais, configurações de rede e conectividade estão funcionais antes de prosseguir.

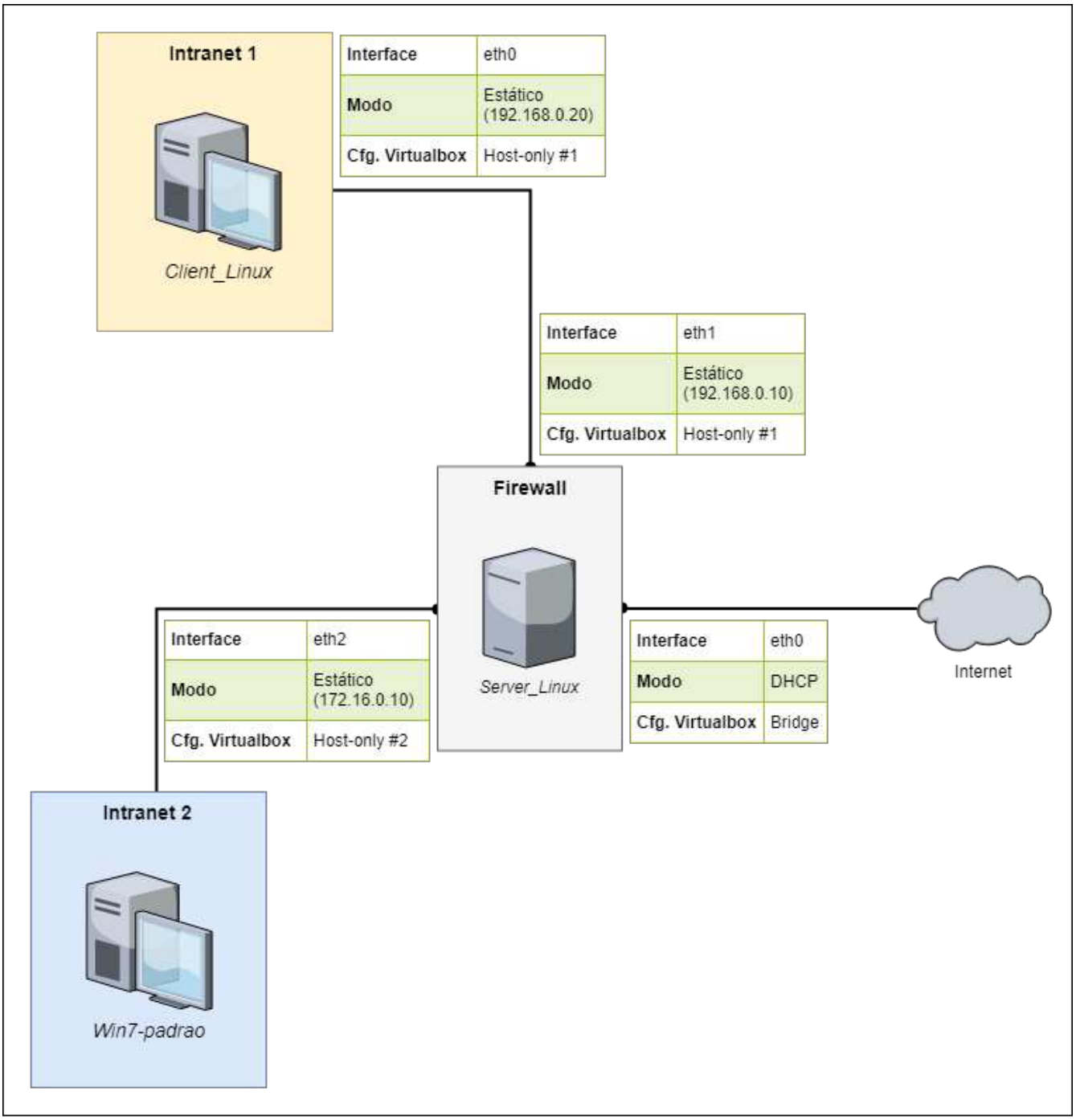

Figura 1: Topologia de rede do curso

2) Configuração do Virtualbox

1. Primeiramente, verifique se todas as máquinas virtuais foram importadas. Você deve ter três VMs, com as seguintes configurações:

   Tabela 1. VMs disponíveis no Virtualbox

Nome VM	Memória
Server_Linux_	512 MB
Client_Linux_	512 MB
Win7-padrao	2048 MB

2. Agora, configure as redes do Virtualbox. Acesso o menu *File* > *Host Network Manager* e crie as seguintes redes:

   Tabela 2. Redes host-only no Virtualbox

Rede	Endereço IPv4	Máscara de rede	Servidor DHCP
Virtualbox Host-Only Ethernet Adapter	192.168.0.254	255.255.255.0	Desabilitado
Virtualbox Host-Only Ethernet Adapter #2	172.16.0.254	255.255.255.0	Desabilitado

3. Finalmente, configure as interfaces de rede de cada máquinas virtual. Para cada VM, acesse *Settings* > *Network* e faça as configurações que se seguem:

   Tabela 3. Interfaces de rede das máquinas virtuais

VM Nome	Interface	Conectado a	Nome da rede
Server_Linux_	Adapter 1	Bridged Adapter	Placa de rede física do *host*
	Adapter 2	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter
	Adapter 3	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter #2
Client_Linux_	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter
Win7-padrao	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter #2

3) Configuração da máquinas virtuais

Agora, vamos configurar a rede de cada máquina virtual de acordo com as especificações da topologia de rede apresentada no começo deste capítulo.

1. Primeiramente, ligue a máquina *Server_Linux* e faça login como usuário `root` e senha `rnpesr`. A seguir, edite o arquivo `/etc/network/interfaces` como se segue, reinicie a rede e verifique o

funcionamento:

```
# hostname
servidor

# whoami
root

# cat /etc/network/interfaces
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0 eth1 eth2

iface eth0 inet dhcp

iface eth1 inet static
  address 192.168.0.10
  netmask 255.255.255.0

iface eth2 inet static
  address 172.16.0.10
  netmask 255.255.255.0

# systemctl restart networking

# ip a s | grep '^ *inet '
    inet 127.0.0.1/8 scope host lo
    inet 10.0.0.204/24 brd 10.0.0.255 scope global eth0
    inet 192.168.0.10/24 brd 192.168.0.255 scope global eth1
    inet 172.16.0.10/24 brd 172.16.0.255 scope global eth2
```

2. Faça o mesmo para a máquina *Client_Linux*:

```
# hostname
cliente

# whoami
root

# cat /etc/network/interfaces
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 192.168.0.20
        netmask 255.255.255.0
        gateway 192.168.0.10

# systemctl restart networking

# ip a s | grep '^ *inet '
    inet 127.0.0.1/8 scope host lo
    inet 192.168.0.20/24 brd 192.168.0.255 scope global eth0
```

3. Na máquina *Win7-padrao*, verifique que a configuração IPv4 da interface de rede está ajustada para obter endreço IP e servidor DNS automaticamente, como mostra a imagem a seguir:

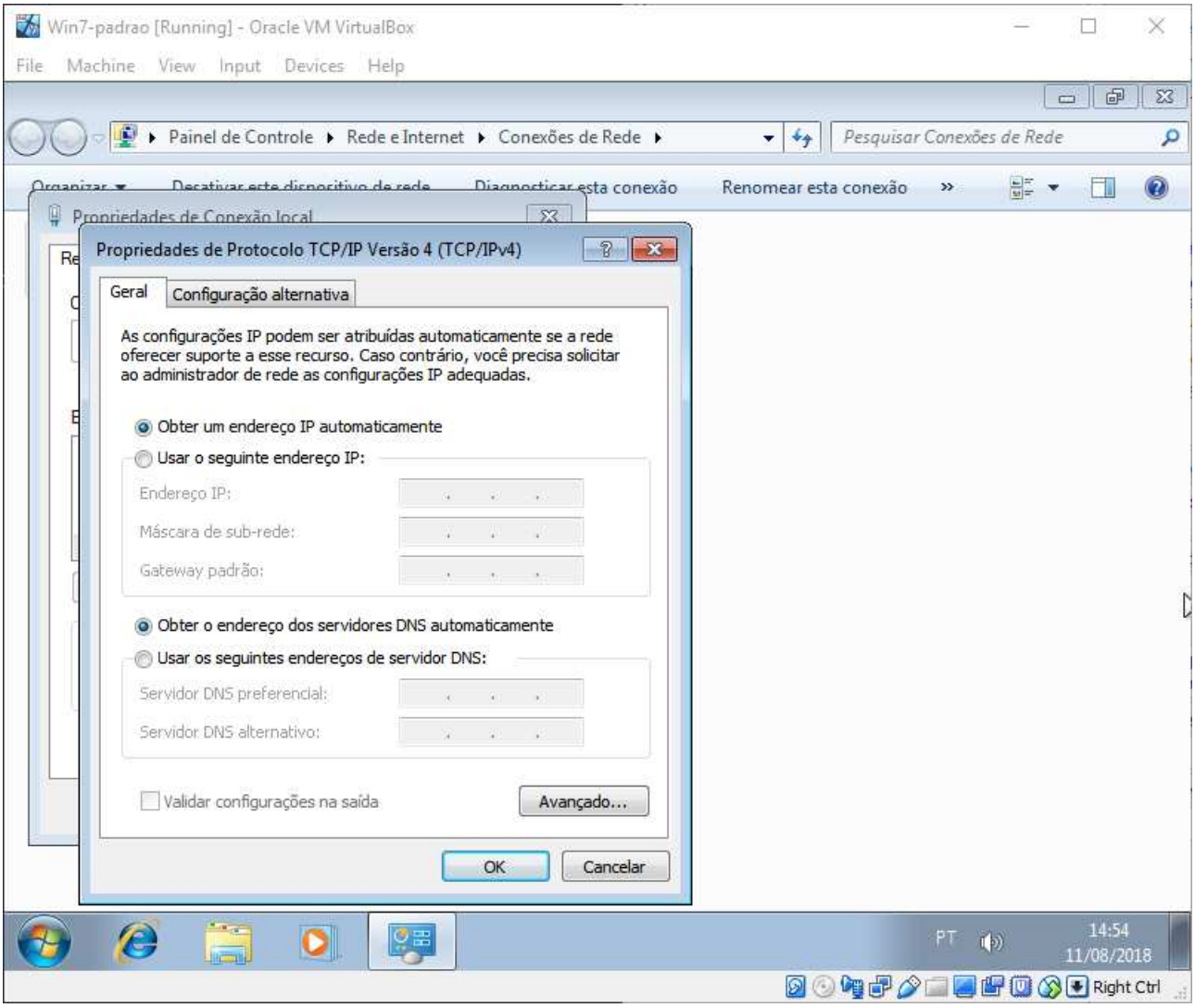

Figura 2: Configuração de rede da máquina Win7-padrao

4) Configuração de firewall e NAT

O passo final é garantir que as máquinas *Client_Linux* e *Win7-padrao* consigam acessar a internet através da máquina *Server_Linux*, que está atuando como um firewall/roteador na topologia de rede do curso.

1. Na máquina *Server_Linux*, verifique que o firewall de host está limpo e permitindo qualquer tipo de conexão:

```
# hostname
servidor

# iptables -L -vn
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

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

# iptables -L -vn -t nat
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

2. A seguir, habilite o repasse de pacotes entre interfaces descomentando a linha `net.ipv4.ip_forward=1` no arquivo `/etc/sysctl.conf`. A seguir, execute `# sysctl -p`:

```
# sed -i 's/^#\(net.ipv4.ip_forward\)/\1/' /etc/sysctl.conf

# grep 'net.ipv4.ip_forward' /etc/sysctl.conf
net.ipv4.ip_forward=1

# sysctl -p
net.ipv4.ip_forward = 1
```

3. Finalmente, habilite IP *masquerading* no firewall através do comando `# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# iptables -L POSTROUTING -vn -t nat
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
```

4. Acesse a máquina *Client_Linux* e faça um teste de conectividade. Você deve conseguir `ping` com um *host* da internet, como `8.8.8.8`, por exemplo:

```
$ ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=31.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=32.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=33.2 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 31.982/32.482/33.291/0.595 ms
```

5. Torne permanente a configuração de *masquerading* na máquina *Server_Linux* editando o arquivo `/etc/rc.local` e adicionando a linha `iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE` antes da linha `exit 0` ao final do arquivo.

```
# cat /etc/rc.local | grep -v '^# \|^#$\|^$'
#!/bin/sh -e
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
exit 0
```

Introdução ao sistema operacional Linux

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

1) Identificando bits de permissão

1. Verifique as permissões do diretório `/tmp`. O que você percebe de diferente em relação às permissões de *outros*?
2. Considerando que há permissão de escrita no diretório para todos, o que o impediria de remover um arquivo de outra pessoa?

2) Identificando e entendendo *hard links*

O número de *links* (*link counter*) que apontam para um arquivo é mantido em seu *inode*. Esse contador é utilizado pelo sistema para controlar a liberação dos blocos do disco alocados ao arquivo quando o contador atingir o valor zero ,ou seja, quando nenhum outro arquivo estiver apontando para o *inode*.

1. Qual o número de *links* do seu diretório *home*?
2. Crie o arquivo `arqses1ex3` no seu diretório *home*. Utilize o comando *touch*.
3. Verifique o número de *links* do arquivo `arqses1ex3` e anote o resultado. Você pode utilizar o redirecionamento de saída para registrar esse resultado no próprio arquivo criado. Essa informação será necessária para uma atividade posterior.
4. Verifique se mudou o número de *links* do seu diretório *home*.
5. Crie um diretório com o nome de `dirses1ex3`, também no seu diretório *home*.
6. Mais uma vez, verifique o número de *links* do seu diretório *home*. Ele mudou? Você saberia dizer por quê?
7. Qual o número de links do diretório `dirses1ex3`?
8. Verifique qual opção deve ser passada ao comando `ls` para que ele liste as informações do diretório `dirses1ex3` e não o seu conteúdo.
9. Você saberia explicar por que o número de *links* do diretório `dirses1ex3` é maior que um?

3) Conhecendo diferenças entre *hard link* e *symbolic link*

Foi explicada a importância dos *links* criados com o comando `ln`. Para criar um *symbolic link*, a opção `-s` deve ser informada na linha de comando. Consulte as páginas do manual para conhecer outras opções.

1. No seu diretório de trabalho, crie um *hard link* para o arquivo `arqses1ex3`. O nome do arquivo criado deverá ser `hosts.hard`.
2. Verifique agora o número de links do arquivo `arqses1ex3` e compare com aquele obtido na

atividade 2. Explique a diferença.

3. Crie um *symbolic link* para o arquivo `arqses1ex3`, que deverá se chamar `hosts.symbolic`.
4. O número de *links* do arquivo `arqses1ex3` aumentou?
5. Caso não tenha aumentado, por que isso aconteceu, considerando que foi criado um *link* para ele?
6. Qual o tamanho do arquivo *hosts.symbolic*?
7. Você percebe alguma correlação entre o tamanho e o arquivo para o qual ele aponta?

4) Trabalhando com *hard link* e *symbolic link*

1. Se o arquivo original `arqses1ex3` fosse removido, o que aconteceria se tentássemos acessá-lo pelo *hard link*? E pelo *symbolic link*?
2. Depois de responder a essas questões, remova o arquivo criado (`arqses1ex3`) e verifique se as suas respostas estão corretas.

5) Conhecendo algumas limitações do *hard link*

1. Crie um arquivo chamado `arqses1ex6`. Em seguida, crie um *hard link* para esse arquivo com o nome `link-arqses1ex6` no diretório `/tmp`. O que aconteceu? Por quê? Como resolver esse problema?

Para que esta atividade tenha efeito, o diretório `/tmp` deverá ter sido criado numa partição diferente da partição onde se encontra o *home* do usuário. Caso essa situação não ocorra, verifique se existe o diretório `/var/tmp` e veja se ele está em outra partição. Se for o caso, use este último para fazer o exercício.

6) Criando *links* para diretórios

Crie, no seu diretório *home*, um *link* simbólico para o diretório `/usr/bin` com o nome de `link-bin`. Com o *link* criado, execute o seguinte:

1. Mude para o diretório `link-bin`.
2. Agora, vá para o diretório pai (utilize a notação ".."). Você saberia explicar por que se encontra no seu diretório *home* e não no diretório `/usr`?

7) Alterando permissões de arquivos e diretórios

O comando `chmod` é utilizado para modificar as permissões de um arquivo. Utilizando a notação octal, execute a seguinte sequência:

1. Modifique a permissão do seu diretório *home* de modo a retirar a permissão de escrita do seu dono.
2. Verifique as permissões associadas ao arquivo `arqses1ex6`. Você tem permissão para escrever

nesse arquivo? O grupo tem?

3. Tente remover o arquivo `arqses1ex6`. Você conseguiu? Em caso negativo, você sabe explicar o motivo?
4. Modifique as permissões do arquivo `arqses1ex6` de forma a retirar a permissão de escrita para o dono e colocá-la para o grupo.
5. Com o uso de redirecionamento, tente copiar o conteúdo do seu diretório *home* para dentro do arquivo `arqses1ex6`.
6. Torne a colocar a permissão para escrita no seu diretório *home* para o dono.

8) Atribuindo as permissões padrão

1. Crie arquivos (`arq1ses1ex9`, `arq2ses1ex9`, etc.) e diretórios (`dir1ses1ex9`, `dir2ses1ex9`, etc.) em seu diretório *home*, após definir cada uma das seguintes *umasks*: `000`; `002`; `003`; `023`; `222`; `022`. Em seguida, observe as permissões que foram associadas a cada um dos arquivos e diretórios.

9) Entendendo as permissões padrões

1. Na execução do exercício anterior, você saberia explicar por que, ainda que utilizando a mesma *umask*, as permissões associadas ao arquivo criado diferem das do diretório?

Usuários e grupos

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

1) Criando contas de usuários

Uma das atividades que fazem parte da rotina diária de um administrador de sistemas é o gerenciamento de contas de usuários. Frequentemente, usuários são criados, modificados, desabilitados ou excluídos do sistema.

1. Descubra se o sistema faz uso de *shadow passwords* ou se ainda utiliza o esquema tradicional.
2. Crie uma conta para você no sistema, seguindo os passos descritos na aula teórica e no material didático.
3. Agora, crie uma conta para o instrutor, utilizando, desta vez, o comando `useradd`. Faça com que a conta criada tenha sete dias de duração e com que o seu diretório de trabalho seja `/NOME`, onde `NOME` é o nome de usuário para o qual a conta deve ser aberta.

Consulte a página de manual do comando `useradd` e procure as informações necessárias para incluir a data de expiração (*expire date*) e criar o diretório de trabalho (*homedir*) em um local diferente do padrão, que é `/home/NOME`. Ainda, não se deve esquecer de escolher e atribuir uma senha para as contas que obedeça aos padrões de segurança apresentados no texto. Observe, ainda, que o diretório *home* não é criado automaticamente pelo comando `useradd`.

4. O comando `useradd` não é uma boa opção para informar a senha do usuário. Por quê?
5. Faça um *script* que simule o comando `newusers`. Para isso, você deve criar um arquivo texto contendo as informações a respeito dos usuários, mantendo o mesmo padrão dos arquivos lidos pelo comando `newusers` (para descobrir o formato, consulte a página de manual: `$ man 8 newusers`). Como este arquivo conterá as senhas dos usuários, é importante removê-lo logo após a criação das contas.

Utilize a variável de sistema `IFS` (*Internal Field Separator*) em seu *script* para definir o caractere ":" como campo que separa as informações sobre as contas.

2) Verificando e modificando informações de contas de usuário

Após a criação de uma conta, é fundamental que o administrador verifique se ela foi criada corretamente.

1. Entre no sistema com o usuário criado no item 3 da atividade 1 e execute os comandos indicados para verificação de uma conta.
2. Seria possível inserir o número de telefone de trabalho desse mesmo usuário, junto com a informação de quem ele é? Faça isso e torne a checar se a sua mudança surtiu efeito.

3) Criando grupos de usuários

O recurso de grupos de usuários é muito útil para compartilhar informações. No momento em que a conta `instrutor` foi criada, no item 3 da atividade 1 deste roteiro, o grupo primário ficou sendo o seu próprio nome de usuário. Isso ocorre sempre que não é atribuído um valor para o grupo primário, no momento da criação de um novo usuário. Como o usuário criado não faz parte de outro grupo, a não ser do seu próprio, ele somente poderá acessar seus arquivos ou aqueles arquivos para os quais haja permissão de acesso para outros usuários.

1. Use o comando apropriado para criar um grupo chamado `grupoteste`.
2. Liste o arquivo `/etc/group` e anote o `GID` que foi atribuído ao grupo criado.
3. Aproveite para observar, no arquivo `/etc/group`, quais são os outros grupos existentes no sistema. Qual o grupo associado ao usuário `root`?
4. Altere o grupo primário do usuário `instrutor`, de modo que este passe a ser o grupo criado no item 1 da atividade 3, `grupoteste`.
5. Se autentique no sistema utilizando a sua conta e inclua seu usuário como administrador do grupo `grupoteste`. Em seguida inclua o usuário `instrutor` no grupo `grupoteste`. Você conseguiu executar as tarefas propostas? Por quê? Como você deve fazer para realizar as tarefas?
6. Altere novamente o grupo primário do usuário `instrutor` para o grupo `instrutor`.

4) Incluindo usuários em grupos secundários

1. Editando o arquivo `/etc/group`, inclua, no grupo `grupoteste`, o usuário criado no terceiro item da atividade 1 desse roteiro (`instrutor`). Note que o grupo primário do usuário não deve mudar; continua sendo o nome do usuário.
2. Agora, utilize um comando apropriado para inserir nesse mesmo grupo o usuário criado para você no primeiro item da atividade 1.

5) Bloqueando contas de usuários

No Linux, é possível impedir temporariamente o acesso ao sistema mesmo que o usuário esteja utilizando uma conta com acesso liberado a este.

1. Utilizando um comando apropriado, bloqueie a conta criada para o instrutor e teste se obteve sucesso no bloqueio.
2. Agora desbloqueie a conta e faça o teste de acesso para verificar se sua alteração surtiu efeito.

6) Removendo uma conta de usuário manualmente

No Linux, é possível executar uma mesma tarefa de diversas maneiras. Para um administrador de sistemas, é importante conhecer essas alternativas, porque elas podem ser úteis em situações específicas em que não seja possível utilizar um dado recurso ou ferramenta do sistema.

1. Sem utilizar o comando `userdel`, remova a conta criada para você no segundo item da atividade

1.

2. Certifique-se de que esse usuário foi realmente excluído do sistema, utilizando um dos comandos que fornecem informações sobre os usuários.
3. Faça um backup de seus dados de modo que o instrutor possa ter sobre eles o mesmo tipo de acesso que você.

7) Obtendo informações sobre usuários

Muitas vezes, é necessário obter informações sobre os usuários de um sistema. Dois comandos que fornecem informações sobre usuários são `finger` e `id`.

1. Verifique os parâmetros do usuário criado na atividade 1 utilizando esses comandos, e descreva a diferença entre os dois a partir dos resultados obtidos. Consulte as páginas de manual para verificar as opções disponíveis nestes comandos.

8) Removendo contas de usuários

1. Utilizando os comandos apropriados, remova a conta criada para o instrutor. Não se esqueça de que um grupo foi especialmente criado para ele e que ele também possui um grupo secundário.

9) Alterando o grupo a que um arquivo pertence

O arquivo `/etc/passwd` contém informações importantes sobre os usuários do sistema. Esse arquivo pertence ao usuário `root` e ao grupo `root`. As permissões de acesso desse arquivo definem que ele só poderá ser modificado pelo usuário `root`.

1. Faça com que esse arquivo pertença ao grupo `grupoteste`, criado na atividade 3. Com isso, os usuários desse grupo, incluindo o usuário criado na atividade 1 poderão acessar esse arquivo por meio das permissões definidas para os usuários do grupo.

10) Alterando permissões de acesso de arquivos

É muito comum o administrador ter que modificar a permissão de arquivos para possibilitar ou impedir que eles sejam lidos ou modificados por diferentes categorias de usuários. A melhor forma de fazer isso é utilizando o comando `chmod`.

1. O arquivo `/etc/passwd` tem apenas permissão de leitura para os usuários do seu grupo proprietário. Use o comando `chmod` para atribuir permissão de escrita ao grupo proprietário desse arquivo. A permissão de escrita nesse arquivo é inicialmente atribuída apenas ao usuário proprietário do arquivo.
2. O setor de controladoria de uma empresa só possuía um funcionário, que pediu demissão. Como não há um diretório específico para armazenar os arquivos do setor, todos os seus arquivos de trabalho estão armazenados em seu diretório *home*. Que passos você deve fazer para disponibilizar estes arquivos para o novo funcionário que será contratado e para que este tipo de problema não volte a ocorrer?

Por motivos de segurança, ao final das atividades, retorne a permissão e o grupo do arquivo `/etc/passwd` para os valores originais.

```
# chown root.root /etc/passwd
# chmod 644 /etc/passwd
# ls -lh /etc/passwd
-rw-r--r-- 1 root root 1,7K Ago  7 16:22 /etc/passwd
```

Processos

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

1) Descobrindo o número de processos em execução

1. Quantos processos estão sendo executados na máquina no momento? Use o comando `wc` para contá-los.
2. Faça um *script* que liste o número de processo que cada usuário está executando.

2) Descobrindo o PID e o PPID de um processo

1. Quais os valores de `PID` e `PPID` do shell que você está utilizando no sistema?
2. Faça um *script* que liste todos os processos que foram iniciados pelo processo `init`. A lista não deve conter mais de uma ocorrência do mesmo processo.

3) Estados dos processos

1. Qual o status mais frequente dos processos que estão sendo executados no sistema? Você saberia explicar por quê?

4) Alternando a execução de processos

1. Execute o comando `$ sleep 1000` diretamente do terminal.
2. Pare o processo e mantenha-o em memória.
3. Liste os processos parados.
4. Coloque-o em *background*.
5. Verifique se o comando sleep 1000 está rodando.
6. É possível cancelar a execução desse comando quando ele está rodando em *background*? Caso seja possível, faça-o.

5) Identificando o RUID e o EUID de um processo

1. Logado como o usuário `aluno`, execute o comando `passwd` no seu terminal. Antes de mudar a senha, abra uma segunda console e autentique-se como `root`. Verifique o `RUID` e o `EUID` associados ao processo `passwd`. Esses valores são iguais ou diferentes? Você saberia explicar por quê? Por fim, cancele a execução do processo `passwd`.

6) Definindo a prioridade de processos

1. Verifique as opções do comando `nice` e em seguida, execute o comando abaixo, verificando sua prioridade, utilizando o comando `ps`:

```
# nice -n -15 sleep 1000 &
[1] 2289
```

2. Repita o comando do primeiro item, passando para o comando `nice` o parâmetro `-n -5`. Verifique como isso afeta a prioridade do processo. Ela aumentou, diminuiu ou permaneceu a mesma?

7) Editando arquivos crontab para o agendamento de tarefas

Neste exercício, trabalharemos com o comando `crontab`, utilizado para editar os arquivos `cron` do agendador de tarefas do sistema. Esses arquivos serão verificados pelo *daemon* `cron` periodicamente em busca de tarefas para serem executadas pelo sistema.

Para entender o funcionamento do `crontab`, o primeiro passo é ler as páginas do manual relevantes. Para o comando `crontab` em si, consulte a seção 1 do manual:

```
$ man 1 crontab
```

Para o formato de um arquivo de configuração `crontab`, consulte a seção 5:

```
$ man 5 crontab
```

1. Existe alguma entrada de `crontab` para o seu usuário?
2. Que opção deve ser usada para editar o seu arquivo de `crontab`?

8) Agendando uma tarefa no daemon cron

Neste exercício, será necessário enviar mensagens de correio eletrônico. Para isso, você deverá utilizar o comando `mail`; o instrutor pode fornecer as informações básicas sobre ele. Um exemplo do uso desse comando para enviar uma mensagem ao endereço `fulano@dominio` com o assunto *Mensagem de teste* é:

```
$ mail fulano@dominio -s "Mensagem de teste" < /dev/null
```

1. Configure o `crontab` para que uma mensagem de correio eletrônico seja enviada automaticamente pelo sistema, sem interferência do administrador às 20:30 horas.
2. Como verificar se a configuração foi feita corretamente?
3. Qual o requisito fundamental para garantir que a ação programada será executada?
4. Há como confirmar se a mensagem foi efetivamente enviada, sem consultar o destinatário?
5. Dê dois exemplos de utilização desse mecanismo para apoiar atividades do administrador de

sistemas.

6. Faça um script que liste os arquivos sem dono do sistema e envie a lista por e-mail ao usuário root.
7. Agende no crontab do usuário `root` o script do item 6, de modo que ele seja executado de segunda a sexta às 22:30 horas.

9) Listando e removendo arquivos crontab

1. Liste o conteúdo do seu arquivo de `crontab` e, em seguida, remova-o. Quais as opções utilizadas para executar as ações demandadas?

10) Entendendo o comando exec

1. Execute o comando `$ exec ls -l`. Explique o que aconteceu.

Sistema de arquivos

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

Em algumas atividades, você trabalhará com a conta `root`, o que lhe dará todos os direitos sobre os recursos do sistema. Seja cauteloso antes de executar qualquer comando.

1) Obtendo informações sobre sistemas de arquivos e partições

Verifique quais são as opções do comando `df` e responda:

1. Quais *file systems* foram definidos no seu sistema?
2. Qual partição ocupa maior espaço em disco?
3. Qual é o *device* correspondente à partição raiz?
4. Os discos do computador que você está utilizando são do tipo `IDE` ou `SCSI`?
5. A que partição pertence o arquivo `/etc/passwd`?
6. Você faria alguma crítica em relação ao particionamento do disco do computador que você está utilizando? Como você o reparticionaria?

2) Determinando o espaço utilizado por um diretório

1. Que subdiretório do diretório `/var` ocupa maior espaço em disco?
2. Faça um *script* para monitorar a taxa de utilização das partições de um servidor. Este script deve enviar um e-mail ao usuário `root` caso a taxa de utilização de um ou mais partições ultrapasse 90% de uso. O e-mail deve informar o(s) *filesystem(s)* e sua(s) respectiva(s) taxa(s) de utilização (somente se estiver acima de 90%).

3) Criando uma nova partição e definindo um novo sistema de arquivos

Você, como administrador de um sistema, pode, a qualquer instante, deparar-se com um problema gerado por uma aplicação que necessita de maior espaço em disco para armazenar informações (isso é muito comum em sistemas de banco de dados). Nessas situações, normalmente, um novo disco é adicionado ao sistema.

A execução desta atividade depende da existência de um espaço não alocado no sistema. Caso não exista este espaço e esta atividade esteja sendo executada em um ambiente virtualizado, pode-se ter a facilidade de adicionar um novo disco à máquina virtual. Consulte o instrutor sobre como proceder.

1. Faça login como usuário `root`. Deve haver um espaço não utilizado no disco do seu cliente. Você deve adicionar esse espaço ao sistema, criando uma partição do tipo utilizado pelo Linux.
2. Formate a partição com o sistema de arquivos `ext4`.
3. Crie um *mount point* chamado `/dados` e monte nele a nova partição.
4. Qual a quantidade de espaço em disco que foi reservada para armazenar os dados dos *inodes*? E da partição em si?
5. Cheque a partição criada com o comando apropriado. Que tipos de checagens foram realizados?
6. Tome as medidas necessárias para que essa partição seja montada toda vez que o sistema for reiniciado, e verifique se isso acontece de fato.

4) Trabalhando com o sistema de *quotas*

Em sistemas compartilhados por muitos usuários, a competição por espaço em disco costuma gerar conflitos que acabam prejudicando o desempenho do sistema e os próprios usuários, caso não haja controle de uso dos recursos. Neste exercício, veremos como habilitar e configurar o sistema de *quotas* do Linux.

1. Faça login com a conta do usuário `root`. Verifique se o sistema de *quotas* está instalado. Se ainda não estiver, execute a instalação.
2. O próximo passo é habilitar o sistema de *quotas* para a partição raiz. Faça isso seguindo os procedimentos descritos na parte teórica dessa sessão de aprendizagem.
3. Crie uma conta de usuário para teste e configure o limite desse novo usuário para 200 MB, utilizando o comando `edquota`.
4. Saia do sistema e entre novamente como o usuário de teste que acaba de ser criado. Como pode ser verificado, a partir dessa conta, as *quotas* de uso de disco? E o espaço efetivamente utilizado?
5. Crie dois arquivos no diretório, utilizando os comandos `cp` e `ln` (criando um link simbólico). Há diferença na forma como o espaço ocupado por esses dois arquivos é contabilizado no sistema de quotas?
6. Como determinar se o sistema de *quotas* está habilitado na inicialização do sistema? E, se não estiver como habilitá-lo?
7. Teste a efetividade do sistema de *quotas*:
8. Faça um *script* que defina o esquema de *quota* para todos os usuários do sistema baseado nas cotas de um usuário passado como parâmetro para esse *script*.

Registro de eventos

As atividades 1, 2 e 3 desta sessão serão realizadas na máquina virtual *Client_Linux*. As atividades 4, 5, 6 e 7 serão realizadas em ambas as máquinas *Server_Linux* e *Client_Linux*, de acordo com o enunciado de cada exercício.

Em algumas atividades, você trabalhará com a conta `root`, o que lhe dará todos os direitos sobre os recursos do sistema. Seja cauteloso antes de executar qualquer comando.

1) Registrando os eventos do kernel

1. Configure seu sistema de modo que os eventos gerados pelo kernel sejam registrados em um arquivo chamado `kernel.log`, no diretório `/var/log`.

2) Analisando os arquivos de log do sistema

Para esta atividade você terá que ter acesso `ssh` à máquina em que está configurando o sistema de logs para que você possa acompanhar, em tempo real, os registros gravados nos arquivos de log.

1. Crie, em sua máquina, uma conta com senha para acesso via `ssh`.
2. A partir de uma máquina remota, faça login via `ssh` utilizando a conta criada no passo anterior. Utilize o comando `tail` com a opção `-f` para verificar em tempo real os registros gerados pelo `syslog` no arquivo `/var/log/auth.log`.
3. Faça um *script* que contabilize o número de tentativas de login mal sucedidas através do `ssh`, listando os IPs de origem e quantas tentativas foram feitas por cada IP.

3) Analisando os arquivos de log binários do sistema

Nesta atividade, você irá trabalhar com os arquivos de log binários armazenados no diretório `/var/log`.

1. Verifique quais foram os dois últimos usuários a efetuarem login em seu computador.
2. Como você poderia verificar as contas existentes em seu computador que nunca efetuaram login?
3. Qual a maneira mais fácil de identificar um login remoto efetuado em seu computador?
4. Faça um *script* que mostre o tempo total que cada usuário ficou logado no sistema utilizando as informações obtidas com o comando `last`.

4) Servidor de log remoto

1. Este exercício deve ser feito utilizando duas máquinas virtuais Linux. Configure um servidor de logs na máquina virtual *Server_Linux*; posteriormente, configure a máquina virtual *Client_Linux* para enviar os registros dos eventos gerados para esse servidor de logs.

2. Após terminar a configuração, efetue um login na máquina *Client_Linux* em um terminal qualquer e verifique onde foi registrado esse evento no servidor de logs *Server_Linux*.
3. Cite três vantagens obtidas com o uso de um servidor de logs.

5) Utilizando o logger

Nesta atividade, você irá verificar uma funcionalidade importante do comando `logger`.

1. Na máquina *Server_Linux*, inclua uma nova regra no arquivo `/etc/rsyslog.conf`, de modo que qualquer evento gerado pelo daemon `cron` seja registrado no arquivo `/var/log/cron.log`.
2. Utilize o comando `logger` para testar se a alteração feita no passo anterior produziu o efeito esperado.

6) Rotacionando arquivos de log do sistema

Nesta atividade, você irá configurar o rotacionamento dos arquivos de log de seu computador.

1. Na máquina *Server_Linux*, realize o rotacionamento mensal do arquivo recém-criado `/var/log/cron.log`, mantendo uma cópia dos dois últimos arquivos compactados e criando, automaticamente, um novo arquivo vazio após o rotacionamento.

7) Aplicativos para análise de arquivos de log

1. Na máquina *Server_Linux*, instale o pacote `logwatch` através do comando `apt-get` e configure-o para enviar um relatório diário do sistema para o usuário `root`. Um exemplo do arquivo de configuração está disponível em `/usr/share/logwatch/default.conf/logwatch.conf`.
2. Ainda na máquina *Server_Linux*, crie uma regra para o `swatch` que envie um e-mail de notificação ao administrador quando alguma tentativa de login via `ssh`, ou `su` para o usuário `root`, falharem.
3. Ainda na máquina *Server_Linux*, habilite o `logcheck` para enviar relatórios ao usuário `root` de 30 em 30 minutos (ex: 1:00, 1:30, etc.).

8) Recomendações básicas de segurança

1. O que você faria para aumentar o nível de segurança em um servidor de logs centralizado? Cite duas opções.

Segurança básica e procedimentos operacionais

As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

1) Identificando senhas fracas

Uma das formas de verificar se o seu sistema atende às recomendações básicas de segurança é utilizar os programas "quebradores" de senha, ou *password crackers*. Neste exercício, utilizaremos um desses programas para mostrar seu funcionamento.

1. Obtenha e instale o *password cracker* John the Ripper, ou simplesmente `john`.
2. Crie o arquivo `/root/dicionario.txt` com uma lista de senhas. Caso considere necessário, acrescente palavras que julgue impróprias para uso em senhas. Por exemplo:
3. Rode o *password cracker* com o comando `# john -wordlist=/root/dicionario.txt -rules /etc/shadow`.
4. Veja o resultado da verificação com o comando `# john -show /etc/shadow`.

2) Descobrindo a funcionalidade do bit SGID em diretórios

A utilidade do SUID e SGID foi vista desde a sessão de aprendizagem 1. Execute a sequência de comandos e depois responda as seguintes perguntas:

1. Crie o grupo `corp` e defina-o como grupo secundário do seu usuário.
2. Entre no sistema a partir da sua conta e crie um diretório chamado `dir_corp`.
3. Verifique a qual grupo pertence o diretório criado no passo acima. Modifique-o para que passe a pertencer ao grupo `corp` e mude a sua permissão para `2755`.
4. Crie, no seu diretório *home* um arquivo chamado `arq1`. Em seguida, mude para o diretório criado no segundo item e crie um arquivo chamado `arq2`.
5. Verifique os grupos aos quais pertencem os arquivos criados no item anterior. Você saberia explicar por que os arquivos pertencem a grupos distintos, embora tenham sido criados pelo mesmo usuário?
6. Quais as vantagens desse esquema?

3) Obtendo informações sobre os recursos computacionais

1. Vimos, no texto teórico, que uma das importantes funções de um administrador de sistemas é acompanhar o uso dos recursos computacionais de sua instituição. Discuta com o seu colega quais comandos vistos em todo o módulo podem auxiliar na coleta desse tipo de informação.

4) Controlando os recursos dos usuários

Um dos grandes desafios de um administrador de sistema, nos tempos atuais, é controlar a ocupação do espaço em disco do seu sistema — aplicações do tipo P2P (*peer-to-peer*), por exemplo, são consumidoras vorazes desse tipo de recurso.

1. Que medidas podem ser tomadas para controlar a ocupação de disco de forma automática?

DNS e NFS

Nestas atividades, você deve trabalhar com duas máquinas virtuais (*Server_Linux* e *Client_Linux*). Ambas devem estar na mesma rede. Como estabelecido na topologia de rede de curso, o endereço 192.168.0.10 será o da máquina *Server_Linux*, e o endereço 192.168.0.20 será o da máquina *Client_Linux*. Teste o funcionamento da rede através do comando `ping` antes de prosseguir com os exercícios.

1) Servidor de DNS Primário

Esta configuração será realizada na máquina virtual *Server_Linux*.

Considerando a rede 192.168.0.0/24, cujo domínio é `empresa.com.br`, configure o servidor de DNS Primário de modo que ele tenha as seguintes máquinas registradas, com tipos de registro associados:

Tabela 4. Configuração DNS

Nome	Endereço IP	Tipo de registro
`servidor.empresa.com.br`	192.168.0.10	NS
`email.empresa.com.br`	192.168.0.15	MX
`cliente.empresa.com.br`	192.168.0.20	A
`windows.empresa.com.br`	192.168.0.25	A
`www.empresa.com.br`	192.168.0.10	CNAME
`meusite.empresa.com.br`	192.168.0.10	CNAME
`pop.empresa.com.br`	192.168.0.15	CNAME
`smtp.empresa.com.br`	192.168.0.15	CNAME

Não se esqueça de configurar a resolução de nomes reversa.

2) Servidor de DNS Secundário

Esta configuração será realizada na máquina virtual *Client_Linux*.

Configure o servidor de DNS Secundário para o domínio `empresa.com.br`. Importante:

- Não se esqueça de informar o endereço IP do servidor secundário no parâmetro `allow-transfer` do servidor primário.
- Os arquivos de zona que forem transferidos devem ser gravados no diretório `/etc/bind/sec` do servidor secundário já que o *daemon* executa como usuário `bind`, que não tem permissão de escrita direta no diretório `/etc/bind`.

3) Configuração de servidor NFS

Esta configuração será realizada na máquina virtual *Server_Linux*.

Crie e exporte o diretório `/dados` via NFS na máquina *Server_Linux* (192.168.0.10), para a máquina *Client_Linux* (192.168.0.20).

4) Configuração de cliente NFS

Esta configuração será realizada na máquina virtual *Client_Linux*.

Instale e configure o cliente NFS na máquina *Client_Linux* (192.168.0.20), monte o diretório remoto `/dados` do servidor no diretório `/mnt/remoto`. Finalmente, realize as configurações necessárias para que sempre que a máquina for reiniciada o diretório `/dados` seja montado automaticamente.

5) Testando o funcionamento do serviço NFS

Na máquina *Server_Linux*, crie um arquivo de nome `teste` no diretório `/dados` e verifique se este aparece no cliente. Depois, edite o arquivo `teste` a partir da máquina *Client_Linux* adicionando a data atual ao conteúdo do arquivo. Volte ao servidor e verifique se o arquivo foi alterado.

LDAP

1) Instalação do servidor OpenLDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Instale e configure um servidor LDAP na máquina *Server_Linux* (192.168.0.10). Você deverá instalar os seguintes pacotes: `slapd, ldap-utils, migrationtools, attr, libpam-ldap, libnss-ldap, nscd`.

Tabela 5. Configuração libpam-ldap e libnss-ldap

Parâmetro	**Valor**
LDAP URI	`ldap://127.0.0.1`
Search base	`dc=empresa,dc=com,dc=br`
LDAP Admin	`cn=admin,dc=empresa,dc=com,dc=br`
LDAP Admin como usuário `root` local	Sim
LDAP requer autenticação	Não
Versão do LDAP	3

Após a instalação e configuração inicial, execute o comando `# dpkg-reconfigure slapd`.

Tabela 6. Configuração do slapd

Parâmetro	**Valor**
Omitir configuração LDAP	Não
Nome DNS	empresa.com.br
Nome da Organização	Empresa
Backend	MDB
Remover base atual em caso de `purge`	Não
Mover base de dados antiga	Sim
Permitir LDAPv2	Não

Finalmente, edite o arquivo `/etc/ldap/ldap.conf` e edite os parâmetros `BASE` e `URI` de acordo com o configurado nesta atividade. Reinicie o servidor LDAP e verifique se está operacional — faça uma consulta-teste usando o comando `ldapsearch`.

2) Usando o *migrationtools*

Esta configuração será realizada na máquina virtual *Server_Linux*.

O *migrationtools* é um conjunto de *scripts* que permite importar as contas locais de um sistema Linux para um diretório LDAP, que já foi instalado na máquina *Server_Linux* (192.168.0.10) durante a atividade 1.

1. Edite o arquivo `/etc/migrationtools/migrate_common.ph`, substituindo as variáveis `$DEFAULT_MAIL_DOMAIN` e `$DEFAULT_BASE` pelos valores configurados na atividade anterior.
2. Entre no diretório `/usr/share/migrationtools` e execute os *scripts* `migrate_base.pl`, `migrate_passwd.pl` e `migrate_group.pl` para exportar as bases (respectivamente) geral, de usuários/senhas e de grupos. Atente-se para a sintaxe de uso de cada *script*.
3. Remova os registros `dc=com,dc=br` e `dc=empresa,dc=com,dc=br` do topo do arquivo gerado pelo *script* `migrate_base.pl`, que já foram incluídos no diretório LDAP na primeira atividade.
4. Adicione os arquivos `.ldif` gerados anteriormente à base LDAP usando o comando `ldapadd`. Consulte sua página de manual para descobrir as opções apropriadas a passar para o comando. Lembre-se, apenas, que o diretório LDAP está utilizando autenticação simples, não SASL, e que é necessário informar um DN administrativo e senha para inserção de dados.
5. Use o comando `ldapsearch` juntamente com um filtro de pesquisa apropriado para listar todos os grupos que foram adicionados ao diretório LDAP pelos arquivos `.ldif` incluídos no passo anterior.

3) Configurança do cliente Linux para uso do LDAP

Esta configuração será realizada na máquina virtual *Client_Linux*.

Para que as clientes Linux possam se autenticar na base de dados do LDAP, é necessário configurar o PAM (*Pluggable Authentication Modules*) e NSS (*Name Service Switch*) para consultarem logins junto ao servidor LDAP.

Configure a máquina *Client_Linux* (192.168.0.20) para se autenticar na base LDAP que está instalada na máquina *Server_Linux* (192.168.0.10). Você deverá instalar os seguintes pacotes: `ldap-utils, libpam-ldap, libnss-ldap, nscd`.

Tabela 7. Configuração libpam-ldap e libnss-ldap no **Client_Linux**

Parâmetro	**Valor**
LDAP URI	`ldap://192.168.0.10`
Search base	`dc=empresa,dc=com,dc=br`
LDAP Admin	`cn=admin,dc=empresa,dc=com,dc=br`
LDAP Admin como usuário `root` local	Sim
LDAP requer autenticação	Não
Versão do LDAP	3

Não se esqueça de editar os arquivos `/etc/ldap/ldap.conf` e `/etc/nsswitch.conf` para habilitar consulta às bases do LDAP durante procedimentos de login.

Se desejar que diretórios *home* sejam criados automaticamente para usuários LDAP inexistentes na máquina local, insira a linha a seguir ao final do arquivo `/etc/pam.d/common-password`:

```
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022
```

Finalmente, para reiniciar a *cache* de usuários e grupos do LDAP, execute `# systemctl restart nscd`. Se houver algum registro de erro nos arquivos de log quanto à inexistência do arquivo `/etc/netgroup`, crie-o manualmente.

4) Configurança do servidor Linux para uso do LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Agora que a máquina *Client_Linux* (192.168.0.20) está configurada para se autenticar na base LDAP remota localizada na máquina *Server_Linux* (192.168.0.10), faça com que o próprio servidor *Server_Linux* autentique-se usando sua base LDAP local.

5) Criação e remoção de usuários e grupos LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Agora que a máquina *Client_Linux* está conectada ao servidor LDAP, adicione um novo usuário e grupo associado, ambos com o mesmo nome, e faça login com o usuário. Para realizar essa tarefa, crie arquivos LDIF manualmente e adicione-os via `ldapadd`. Não esqueça de definir a senha através do comando `ldappasswd`.

Observação: Para evitar confusões entre a base de usuários do LDAP e a base local dos clientes, é recomendável adotar um *buffer* numérico entre os usuários locais e os usuários do diretório. Faça com que o UID e GID dos novos usuários/grupos comece a partir de 5000.

6) Criação e deleção automática de usuários LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

O esquema de criação de usuários manualmente acima funcionou, como visto. Não é, no entanto, muito conveniente do ponto de vista de manutenção do sistema proceder dessa forma. Seria mais interessante, se possível, automatizar essa tarefa para facilitar sua execução no dia-a-dia.

Crie um *script* que faça a adição e deleção automática de usuários na base LDAP. Atente-se para o fato de que os UIDs e GIDs desses usuários não devem se confundir com o dos sistemas locais. Use o valor mínimo de 5000 para ambos.

DHCP, FTP e SSH

1) Configuração do servidor DHCP

Esta configuração será realizada na máquina virtual *Server_Linux*.

O objetivo do serviço *Dynamic Host Configuration Protocol* (DHCP) é automatizar a distribuição de endereços e configurações do protocolo TCP/IP para quaisquer dispositivos conectados a uma rede, como computadores, impressoras, hubs e switches.

Instale um servidor DHCP na máquina *Server_Linux*, usando o pacote `isc-dhcp-server`, e configure-o com as seguintes características:

- Escutar na interface `eth1`, com endereço IP 192.168.0.10/24;
- Distribuir endereços na faixa 192.168.0.200 até 192.168.0.250;
- Definir como roteador o próprio servidor DHCP, 192.168.0.10;
- Distribuir informações de servidor DNS conforme configurações realizadas no capítulo 7 — DNS e NFS.

A seguir, teste seu funcionamento usando a máquina *Client_Linux* — altere as configurações de rede dessa máquina para obter IP de forma dinâmica, e não estática. Que informações podem ser encontradas no arquivo `/var/lib/dhcp/dhcpd.leases`?

2) Configuração de IP fixo por endereço MAC

Esta configuração será realizada na máquina virtual *Server_Linux*.

Configure o servidor DHCP para sempre fornecer o endereço 192.168.0.20 para o *host Client_Linux*, através da fixação de seu endereço físico (MAC). Verifique o funcionamento da sua configuração.

3) Configuração do servidor DHCP para múltiplas sub-redes

Esta configuração será realizada na máquina virtual *Server_Linux*.

Expanda a configuração do servidor DHCP instalado na máquina *Server_Linux* para que, além de servir à rede 192.168.0.0/24, também atenda clientes da rede 172.16.0.0/24 com as seguintes características:

- Escutar na interface `eth2`, com endereço IP 172.16.0.10/24;
- Distribuir endereços na faixa 172.16.0.50 até 172.16.0.80;
- Definir como roteador o próprio servidor DHCP, 172.16.0.10;
- Distribuir informações de servidor DNS conforme configurações realizadas no capítulo 7 — DNS

e NFS.

Note que para o passo de distribuição de informações DNS será necessário fazer ajustes também à configuração do serviço `bind`. Ele deve estar preparado para escutar requisições vindas da rede 172.16.0.0/24.

A seguir, teste seu funcionamento usando a máquina *Win7-padrao*. O IP obtido pela máquina está dentro da faixa estipulada? É possível resolver nomes e navegar normalmente?

4) Configuração do servidor FTP

Esta configuração será realizada na máquina virtual *Server_Linux*.

O protocolo *File Transfer Protocol* (FTP) permite a um usuário remoto transferir arquivos para um servidor ou vice-versa.

Instale e configure o pacote `vsftpd` na máquina *Server_Linux*. A seguir, crie um novo usuário `ftpuser` que não possua shell válido e, utilizando esse usuário, acesse a partir da máquina *Client_Linux* o serviço de FTP.

5) Login remoto seguro usando SSH

O *Secure Shell* (SSH) é um protocolo criptográfico de rede para permitir operação remota de serviços de forma segura, mesmo operando sob uma rede insegura. Ele foi desenvolvido como um substituto seguro para aplicações de shell remoto como `telnet`, `rlogin` e `rsh`.

Se indisponível, instale o serviço `openssh-server` na máquina *Server_Linux*. Em seguida, acesse-o remotamente a partir da máquina *Client_Linux* e execute o comando `hostname`.

6) Conexão SSH via chaves assimétricas

A partir da máquina *Client_Linux*, crie um par de chaves RSA de 4096 bits com o comando `ssh-keygen`. A seguir, utilize o comando `ssh-copy-id` para copiar a chave pública para pasta do usuário `aluno` na máquina *Server_Linux*. Finalmente, faça login na máquina *Server_Linux* e verifique que a senha não é solicitada.

Aponte em qual arquivo a chave pública RSA foi armazenada na máquina *Server_Linux*, e exiba seu conteúdo.

7) Cópia remota de arquivos via SSH

A partir da máquina *Client_Linux*, crie um arquivo texto com conteúdo qualquer e transfira-o para a máquina *Server_Linux* usando o comando `scp`. Após a cópia, exiba seu conteúdo e verifique que a cópia foi completada com sucesso.

8) FTP seguro via SSH

A partir da máquina *Client_Linux*, crie um arquivo texto com conteúdo qualquer e transfira-o para a máquina *Server_Linux* usando o comando `sftp`. Após a cópia, exiba seu conteúdo e verifique que a cópia foi completada com sucesso.

Servidor Web

As atividades desta sessão serão realizadas na máquina virtual *Server_Linux*, com pequenas exceções apontadas pelo enunciado dos exercícios.

O objetivo de um servidor web é, em essência, servir conteúdo para a *world wide web*. Esse objetivo é atingido servindo requisições enviadas ao servidor através do protocolo HTTP, bem como protocolos relacionados. Nesta sessão iremos instalar e configurar o servidor web Apache, um dos mais populares servidores HTTP *open source* do mundo.

1) Instalação do servidor web Apache

Instale o servidor web Apache (pacote `apache2`). Teste o funcionamento da instalação acessando a página web a partir de qualquer navegador (seja na máquina física, *Client_Linux* ou *Win7-padrao*).

2) Configuração de *virtualhosts*

Virtualhosts, ou servidores virtuais, podem ser utilizados nos seguintes casos comuns:

- Hospedar múltiplos *sites* diferentes em um mesmo endereço IP;
- Hospedar múltiplos *sites*, cada um com seu IP específico.

Destes, o primeiro cenário é o mais usual, e o que será abordado nesta atividade.

No servidor web Apache instalado em nosso servidor Debian, os arquivos de configuração de todos os *sites* devem ser colocados no pasta `/etc/apache2/sites-available`. Esses *sites* podem estar ativos ou inativos:

- Para ativar um *site*, basta criar um *link* simbólico do arquivo original para a pasta `/etc/apache2/sites-enabled` e recarregar o servidor Apache. Esse *link* pode ser criado manualmente, ou através do comando `a2ensite` ("*Apache 2 enable site*").
- Para desabilitar um *site*, toma-se o caminho oposto: apague o *link* simbólico da pasta `/etc/apache2/sites-enabled`, ou use o comando `a2dissite` ("*Apache 2 disable site*").

Relembrando a sessão 7 — DNS e NFS, criamos duas entradas `CNAME` apontando para a máquina *Server_Linux*, quais sejam:

```
# cat /etc/bind/db.empresa.com.br | grep 'CNAME *servidor'
www       IN    CNAME         servidor
meusite   IN    CNAME         servidor
```

1. Crie dois *virtualhosts* na máquina *Server_Linux*, um respondendo requisições enviadas para `www.empresa.com.br` e outro para `meusite.empresa.com.br`.
2. Crie pastas específicas para cada *virtualhost* dentro do diretório `/var/www`.
3. Crie arquivos `index.html` na raiz dessas pastas que identifiquem cada um dos *virtualhosts*.

4. Acesse os nomes de domínio a partir de um navegador (seja na máquina física, *Client_Linux* ou *Win7-padrao*) e verifique que suas configurações surtiram efeito.

3) Configuração de criptografia SSL

O protocolo HTTP não possui nenhum recurso de criptografia e, por consequência, todo o tráfego de rede gerado entre cliente e servidor poderia ser visualizado por um atacante. Para aumentar a segurança de aplicações web, é interessante habilitar o suporte a conexões cifradas através do *Secure Sockets Layer* (SSL).

1. Habilite o módulo SSL do Apache através do comando `a2enmod` ("*Apache 2 enable module*").
2. Crie um certificado auto-assinado RSA de 4096 bits para o *virtualhost* `meusite.empresa.com.br`, com validade de um ano. Armazene a chave pública na pasta `/etc/ssl/certs`, e a chave privada em `/etc/ssl/private`. Tenha atenção às permissões de arquivo e usuário/grupo dono.
3. Configure o *virtualhost* `meusite.empresa.com.br` para utilizar o protocolo HTTPS em qualquer conexão. Redirecione qualquer conexão sem criptografia direcionada à porta 80/HTTP para a porta 443/HTTPS.
4. Acesse o domínio `meusite.empresa.com.br` a partir de um navegador (seja na máquina física, *Client_Linux* ou *Win7-padrao*) e verifique que suas configurações surtiram efeito.

4) Autenticação e acesso a conteúdo restrito usando LDAP

Autenticação de usuários, especialmente em áreas sensíveis de um *site*, é integral à configuração de segurança de servidores web. Em particular, estamos interessados em habilitar autenticação para uma área restrita do *virtualhost* `meusite.empresa.com.br`.

1. Habilite o módulo de autenticação LDAP do Apache, `authnz_ldap`, através do comando `a2enmod`.
2. Crie uma pasta `/restrito` dentro da raiz do *virtualhost*. Dentro dessa pasta, crie um arquivo `index.html` que possa ser usado para testar a configuração.
3. Configure o *virtualhost* para requerer autenticação quando um usuário tentar acessar a URL `meusite.empresa.com.br/restrito`. Exija que o cliente forneça uma combinação de usuário/senha válida e existente na base LDAP local.
4. Acesse a URL `meusite.empresa.com.br/restrito` a partir de um navegador (seja na máquina física, *Client_Linux* ou *Win7-padrao*) e verifique que suas configurações surtiram efeito.

5) Habilitando páginas pessoais de usuários

O módulo `userdir` do Apache permite a um usuário publicar seu próprio *site*, localizado dentro da sua pasta pessoal. Ele procura uma pasta com nome `public_html` dentro do diretório *home* do usuário e, caso existente, serve o conteúdo dessa pasta via HTTP.

1. Habilite o módulo páginas pessoais do Apache, `userdir`, através do comando `a2enmod`.
2. Crie a pasta `public_html` dentro do diretório *home* do usuário `aluno` e insira dentro dela um

arquivo `index.html` que permita testar a configuração.

3. Configure o sistema para que todos os usuários criados futuramente já tenham a pasta `public_html` criada automaticamente eu seus diretórios *home*.
4. Teste o acesso à página pessoal do usuário `aluno` a partir de um navegador (seja na máquina física, *Client_Linux* ou *Win7-padrao*), verificando que suas configurações surtiram efeito.

Correio Eletrônico — SMTP

As atividades desta sessão serão realizadas na máquina virtual *Server_Linux*.

Neste capítulo iremos realizar a configuração da primeira parte de um serviço de correio eletrônico: o envio e recebimento de emails entre domínios através do protocolo *Simple Mail Transfer Protocol* (SMTP). Iremos instalar e configurar o Postfix, uma dos servidores SMTP *open source* mais populares do mundo. Juntamente com o Postfix iremos instalar também o Cyrus SASL, um programa que provê módulos de autenticação plugáveis para verificarmos usuários e senhas via acesso cifrado, com criptografia TLS.

1) Instalação do servidor SMTP Postfix

Antes de instalar o Postfix, temos que corrigir alguns aspectos da nossa instalação atual. Como você se recorda da sessão 7 — DNS e NFS, configuramos a máquina *Server_Linux* com o nome de domínio `servidor.empresa.com.br`, no IP 192.168.0.10. Da mesma forma, inserimos uma entrada fictícia no DNS para uma máquina `email.empresa.com.br` no IP 192.168.0.15, que não existe em nossa topologia de rede.

Já que vamos instalar o Posfix + Cyrus na máquina *Server_Linux*, temos que apontar o nome `email.empresa.com.br` para o IP 192.168.0.10.

Contudo, não podemos tomar o caminho mais fácil, que seria criar um registro de *alias* `CNAME` do nome `email.empresa.com.br` para o nome `servidor.empresa.com.br` — a RFC 2181, seção 10.3 (https://tools.ietf.org/html/rfc2181) proíbe uso de `CNAME` para apontamentos `MX`, exigindo que esses apontamentos sejam feitos diretamente por registros `A`.

Isso exige uma série de alterações ao registro direto do domínio `empresa.com.br`, no arquivo `/etc/bind/db.empresa.com.br`, que fica como se segue:

```
$TTL 86400 ; (1 day)
$ORIGIN empresa.com.br.
@         IN    SOA           email.empresa.com.br. admin.empresa.com.br. (
                2018081200    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            email.empresa.com.br.

@         IN    MX      10    email.empresa.com.br.

email     IN    A             192.168.0.10
cliente   IN    A             192.168.0.20
windows   IN    A             192.168.0.25

meusite   IN    CNAME         email
pop       IN    CNAME         email
servidor  IN    CNAME         email
smtp      IN    CNAME         email
www       IN    CNAME         email
```

Da mesma forma, surge um problema também na resolução de registros reversos do domínio. Não é recomendado que haja múltiplos apontamentos `PTR` para o mesmo endereço IP, sob pena de obter respostas diferentes em duas *queries* DNS distintas. Vamos alterar o registro reverso no arquivo `/etc/bind/db.0.168.192`, deixando-o assim:

```
$TTL 86400 ; (1 day)
$ORIGIN 0.168.192.in-addr.arpa.
@         IN    SOA           email.empresa.com.br. admin.empresa.com.br. (
                2018081200    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            email.empresa.com.br.

@         IN    MX      10    email.empresa.com.br.

10        IN    PTR           email.empresa.com.br.
20        IN    PTR           cliente.empresa.com.br.
25        IN    PTR           windows.empresa.com.br.
```

Agora, vamos testar. Reinicie o serviço `bind` e verifique se o DNS que responde pelo domínio

empresa.com.br é, de fato, a máquina email.empresa.com.br:

```
# systemctl restart bind9.service

# dig -t NS empresa.com.br

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> -t NS empresa.com.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35860
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;empresa.com.br.                        IN      NS

;; ANSWER SECTION:
empresa.com.br.         86400   IN      NS      email.empresa.com.br.

;; ADDITIONAL SECTION:
email.empresa.com.br.   86400   IN      A       192.168.0.10

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Aug 12 14:50:45 -03 2018
;; MSG SIZE  rcvd: 79
```

De igual forma, verifique o registro reverso do IP 192.168.0.10, que deve retornar o nome email.empresa.com.br. Finalmente, o nome servidor.empresa.com.br torna-se agora um *alias* do CNAME email.empresa.com.br.

```
# nslookup 192.168.0.10
Server:         127.0.0.1
Address:        127.0.0.1#53

10.0.168.192.in-addr.arpa       name = email.empresa.com.br.

# nslookup servidor.empresa.com.br
Server:         127.0.0.1
Address:        127.0.0.1#53

servidor.empresa.com.br canonical name = email.empresa.com.br.
Name:   email.empresa.com.br
Address: 192.168.0.10
```

Ainda falta alterar os registros locais de nomes, nos arquivos /etc/hostname, /etc/mailname e /etc/hosts. Altere-os como mostrado a seguir:

```
# cat /etc/hostname
email

# cat /etc/mailname
email.empresa.com.br

# cat /etc/hosts
127.0.0.1                       localhost
127.0.1.1    email.empresa.com.br email
192.168.0.10 email.empresa.com.br email

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Finalmente, reinicie a máquina *Server_Linux*. No próximo login, o nome mostrado pelo *prompt* do shell deve ser `USERNAME@email:~$`.

```
# reboot

(...)

$ ssh aluno@192.168.0.10
You have new mail.
Last login: Sun Aug 12 18:00:53 2018 from 192.168.0.254
aluno@email:~$
```

Isso feito, podemos começar a atividade. Instale o Postfix + Cyrus SASL na máquina *Server_Linux* (pacotes `postfix`, `sasl2-bin` e `mailutils`). Em seguida, reconfigure o Postfix (comando `dpkg-reconfigure postfix`) de acordo com as informações da tabela abaixo:

Tabela 8. Configurações do Postfix

Parâmetro	**Valor**
Tipo geral de configuração de e-mail	Site da internet
Nome de e-mail do sistema	email.empresa.com.br
Destinatário das mensagens para root e postmaster	Em branco
Outros destinos para os quais deve aceitar mensagens	email.empresa.com.br, localhost.empresa.com.br, empresa.com.br, localhost
Forçar atualizações síncronas na fila de mensagem	Não
Redes locais	127.0.0.0/8, 192.168.1.0/24, 172.16.0.0/24, [::ffff:127.0.0.0]/104, [::1]/128

Parâmetro	Valor
Usar procmail para entrega local	Sim
Limite de tamanho da caixa postal	0
Caractere de extensão de endereço local	+
Protocolos de internet para usar	Todos

Crie um par de chaves RSA de 4096 bits e validade de dois anos para permitir conexões TLS ao seu servidor, com chave pública em `/etc/ssl/certs/smtpd.crt` e chave privada em `/etc/ssl/private/smtpd.key`. Feito isso, configure o Postfix, editando o arquivo `/etc/postfix/main.cf`, e:

- Habilite criptografia TLS em conexões oriundas dos clientes, de forma opcional;
- Use as chaves assimétricas criadas acima para implementar a cifragem TLS;
- Habilite autenticação SASL dos tipos PLAIN e LOGIN, comunicando-se com o *daemon* `saslauthd` do Cyrus — deve-se consultar a base de usuários locais via PAM para autenticação.

Atente-se para o fato de que, por padrão, o Postfix opera dentro de um ambiente `chroot`. Será necessário editar opções padrão do `saslauthd` no arquivo `/etc/default/saslauthd` para adaptar-se a esse cenário. Mais além, adicione o usuário do `postfix` ao grupo `sasl` para permitir comunicação entre os dois *daemons*.

Ao final do processo, use o comando `telnet` para testar a configuração realizada, logando no servidor SMTP com usuário `aluno` e senha `rnpesr` pelo método PLAIN.

2) Envio e recebimento de mensagens por *telnet*

Vamos agora testar o envio de mensagens usando o comando `telnet`, diretamente a partir do servidor SMTP. Este teste visa averiguar o funcionamento do servidor de e-mail sem a influência de configurações de clientes de e-mail (*Mail User Agents* — MUA).

Conecte-se ao servidor SMTP por `telnet` com um usuário qualquer existente na base local de usuários ou LDAP e envie email para outro usuário usando os comandos `MAIL` e `RCPT TO` do SMTP. Logue na conta do destinatário e verifique que a mensagem foi recebida.

3) Análise do log de envio

Envie uma nova mensagem de email usando o `telnet`, e monitore ao mesmo tempo o arquivo `/var/log/mail.log` por alterações. Responda, apontando a excerto do log que identifica a informação:

- Qual é o IP de origem da conexão SMTP?
- Qual o nome do usuário que efetuou login?
- Qual o endereço do destinatário da mensagem?
- Qual o método de entrega da mensagem para a caixa do usuário?

Correio Eletrônico — POP/IMAP

As atividades 1, 2 e 3 desta sessão serão realizadas na máquina virtual *Server_Linux*. A atividade 4 será realizada na máquina *Win7-padrao*.

Iremos continuar a configuração da sessão anterior, instalando e configurando o MDA (*Mail Delivery Agent*) Courier.

1) Configuração de entrega *Maildir*

No momento, o Postfix está configurado para entregar mensagens no estilo `mbox`, em que todas as mensagens ficam em um único arquivo no diretório *home* do usuário. A modalidade de entrega `Maildir`, mais moderna, é preferível porque coloca cada mensagem dentro de um arquivo próprio, e as indexa permitindo controle de duplicidade, tempos de expiração e facilita procedimentos de busca. Além disso, o formato `Maildir` é mais performático que o `mbox`.

Crie, dentro da pasta de cada usuário existente no servidor, um diretório de nome `Maildir` com as seguintes sub-pastas: `new`, `cur`, `tmp`, `.Drafts`, `.Spam` e `.Trash` (observe o caractere "." na frente das últimas três pastas, indicando que são ocultas). Ajuste a permissão do diretório `Maildir` para `700`. A seguir, faça com que todos os usuários criados futuramente já tenham essa estrutura de diretórios criada em suas pastas *home* automaticamente.

Depois, altere o estilo de entrega do Postfix de `mbox` para `Maildir`. Finalmente, envie uma mensagem para um usuário e teste se sua configuração surtiu efeito.

2) Configuração do MDA Courier POP/IMAP

Os protocolos *Post Office Protocol* (POP) e *Internet Message Access Protocol* (IMAP) são utilizados pelos clientes de email (MUAs) para recuperar mensagens armazenadas no servidor de e-mail. Nesta atividade iremos configurar os servidores POP e IMAP, e testá-los usando o comando `telnet`.

Instale o Courier-POP e Courier-IMAP, pacotes `courier-imap-ssl`, `courier-pop-ssl`, `libsasl2-modules-ldap` e `gamin`. Passe a opção `--no-install-recommends` para o `apt-get` para que não sejam instalados alguns pacotes adicionais desnecessários à configuração que será feita. Ao ser perguntado se deseja "Criar diretórios para administração via web", responda negativamente.

Teste a conexão com os servidores POP e IMAP. Em caso de sucesso, autentique-se em ambos usando o comando `telnet`.

3) Configuração de autenticação do POP/IMAP em LDAP

Altere as configurações do Cyrus-SASL para permitir autenticação a partir do diretório LDAP, em lugar do PAM. Você deve alterar os arquivos `/etc/default/saslauthd` e `/etc/postfix/sasl/smtpd.conf`. Além disso, será necessário criar um novo arquivo, `/etc/saslauthd.conf`, para especificar a base de pesquisa e filtros de busca na base LDAP.

Teste o funcionamento da configuração usando o comando `testsaslauthd`. Lembre-se que o Postfix está operando em `chroot`, e por conseguinte a localização do *socket* do `saslauthd` deve ser informada manualmente.

Em caso de dúvidas, consulte http://www.postfix.org/SASL_README.html . Tenha especial atenção à configuração do *plugin* `ldapdb`.

4) Utilização de clientes POP/IMAP

Os programas clientes de e-mail (MUA) utilizam-se dos protocolos POP ou IMAP para recuperar mensagens no servidor de e-mail. Nesta atividade iremos configurar um cliente para o recebimento de mensagens usando esses protocolos.

1. Instale o cliente de e-mail *Mozilla Thunderbird* na máquina *Win7-padrao*. Inicie o programa e crie uma nova conta de e-mail para o usuário `aluno`. Na tela inicial, informe:

   Tabela 9. Opções para criação de conta de e-mail existente

Parâmetro	Valor
Seu nome	aluno
Endereço de e-mail	aluno@empresa.com.br
Senha	rnpesr

2. Agora, clique em "Continuar". O *Thunderbird* irá tentar buscar configuração automática dos servidores, sem sucesso. Clique então em "Config. manual", e informe:

   Tabela 10. Configurações avançadas para criação de e-mail

Tipo	Protocolo	Nome do servidor	Porta	SSL	Autenticação
Recebimento	IMAP	email.empresa.com.br	143	Nenhuma	Senha normal
Envio	SMTP	email.empresa.com.br	25	STARTTLS	Senha normal

3. Na parte de baixo, em "Nome de usuário", troque o valor padrão `aluno@empresa.com.br` para `aluno` apenas. Garanta que ambos os campos "Recebimento" e "Envio" estão corretos. Finalmente, clique em "Concluído".
4. O *Thunderbird* irá avisar que o recebimento de e-mails (via IMAP) não está usando criptografia. Marque a caixa "Eu entendo os riscos" e depois clique em "Concluído".
5. Terminado esse passo, crie uma nova conta de e-mail, com as mesmas configurações explicadas acima, para um outro usuário do servidor.
6. Finalmente, teste o envio e recebimento de e-mails entre os dois usuários e verifique que o serviço está funcionando como esperado.

Proxy Squid

Nesta sessão iremos instalar e configurar o Squid, uma solução de *proxy* web que provê funcionalidades de *cache* e redirecionamento. O Squid pode ser utilizado para diversos fins: acelerar o acesso web a partir da realização de *cache* de páginas acessadas com frequência, realizar *cache* de requisições web, DNS outros tipos de consulta para um grupo de usuários, e filtragem de acesso por domínio, URL e análise de conteúdo de páginas. Normalmente configura-se o Squid para trabalhar com os protocolos HTTP e FTP, mas também é possível filtrar requisições HTTPS através de inspeção SSL/TLS.

1) Instalação e configuração inicial do servidor *proxy* Squid

Esta configuração será realizada na máquina virtual *Server_Linux*.

Instale e configure o servidor *proxy* Squid na máquina *Server_Linux*, pacotes `squid3` e `sarg`. Configurações:

- Autorizar conexões vindas de ambas as redes internas, 192.168.0.0/24 e 172.16.0.0/24.
- Recusar demais conexões.
- Diretório de *cache* de páginas em `/var/spool/squid3`
- Log de acessos em `/var/log/squid3/access.log`
- Log geral do *proxy* em `/var/log/squid3/cache.log`
- Porta de acesso 3128/TCP.

2) Configuração do navegador cliente do *proxy*

Esta configuração será realizada na máquina virtual *Win7-padrao*.

Vamos testar a configuração realizada. Acesse a máquina *Win7-padrao* e configure o *proxy* do sistema para o IP da máquina *Server_Linux*. A seguir, acesse um website na porta 80/HTTP (sugestão: http://www.openbsd.org), teste se houve sucesso na conexão, e verifique se o log de acessos do Squid fez o *cache* das páginas solicitadas pelo usuário.

3) Configuração de controles de acesso

Esta configuração será realizada nas máquinas virtuais *Server_Linux* e *Win7-padrao*.

Vamos agora implementar controles de acesso ao servidor *proxy* usando ACLs (*Access Control Lists*). Para testar as configurações, evite usar websites HTTPS, pois o Squid está configurado para HTTP apenas; além disso, o navegador Internet Explorer da máquina *Win7-padrao* está bastante desatualizado. O website http://www.openbsd.org é um bom alvo para testes.

Implemente os seguintes controles:

a. Bloqueio via endereço físico (MAC) — `acl` com palavra-chave `arp`.

b. Bloqueio via endereço IP de origem — `acl` com palavra-chave `src`.

c. Bloqueio pela hora de acesso — `acl` com palavra-chave `time`. Utilize os comandos `date -s` e `hwclock --systohc` para ajustar o relógio do servidor para um horário proibido e testar sua configuração.

d. Bloqueio por expressão regular de extensão de arquivo — `acl` com palavra-chave `urlpath_regex`. Faça com que o acesso a qualquer arquivo com as extensões `.avi`, `.mp3` ou `.pdf` seja bloqueado. Use a pesquisa `site:ftp.openbsd.org filetype:pdf` no Google para encontrar um arquivo que se encaixe no bloqueio configurado.

e. Bloqueio por expressão regular de palavra em URL — `acl` com palavra-chave `urlpath_regex`. Faça com que qualquer URL que contenha as palavras `crypto`, `playboy`, `sexo`, `torrent` e `virus` seja bloqueada. Acesse a URL http://www.openbsd.org/crypto.html para testar a configuração.

f. Bloqueio por domínio de destino — `acl` com palavra-chave `dstdomain`. Faça com que qualquer acesso aos domínios `facebook.com`, `instagram.com`, `twitter.com` e `whatsapp.com` seja negado. Acesse a URL http://web.whatsapp.com para testar sua configuração.

3) Configuração do SARG

Esta configuração será realizada na máquina virtual *Server_Linux*.

Vamos agora configurar o *Squid Analysis Report Generator*, ou simplesmente SARG. O SARG é um gerador de relatórios de acesso do Squid, que analisa os arquivos de log deste para produzir informações relevantes para o administrador de sistemas.

Já instalamos o pacote do SARG na atividade 1 desta sessão. Configure-o da seguinte forma:

- Analisar log do Squid em `/var/log/squid3/access.log`.
- Produzir relatórios no diretório `/var/www/meusite/squid-reports`.
- Não resolver endereços IP para nomes.
- Usar formato de data no padrão europeu (mesmo utilizado no Brasil).
- Produzir relatórios no *charset* UTF-8.

Uma vez configurado o programa, rode o comando `sarg` como root e acesse a URL `https://meusite.empresa.com.br/squid-reports/` para visualizar os resultados.

4) *Proxy* transparente

Esta configuração será realizada nas máquinas virtuais *Server_Linux* e *Win7-padrao*.

Pode não ser interessante ter que configurar cada estação cliente para que utilize expressamente o *proxy*. É possível configurar o firewall da rede para redirecionar conexões às portas 80/HTTP e

443/HTTPS de forma automática para o *proxy*, sem editar as configurações de qualquer cliente — esse tipo de cenário é denominado *proxy* transparente.

Edite o firewall `iptables` da máquina *Server_Linux* para que os pacotes passantes com destino à porta 80/HTTP de um servidor externo sejam redirecionados para o Squid local, operando na porta 3128/TCP.

Use o pacote `iptables-persistent` para tornar suas configurações permanentes mesmo após o `reboot` da máquina. Na instalação do pacote, quando perguntado, responda:

Tabela 11. Configurações do iptables-persistent

Pergunta	Resposta
Salvar as regras IPv4 atuais?	Sim
Salvar as regras IPv6 atuais?	Sim

Não se esqueça de configurar o Squid em modo transparente. Finalmente, limpe as configurações de *proxy* da máquina *Win7-padrao*, e verifique que a *cache* e bloqueios do Squid permanecem operacionais.

Observe que todas as configurações desta sessão foram feitas para um *proxy* HTTP apenas. Embora funcional, muito sites hoje em dia utilizam HTTPS exclusivamente, o que torna nossa implantação apenas parcialmente útil.

O módulo *Peek and Splice* do Squid (https://wiki.squid-cache.org/Features/SslPeekAndSplice), disponível a partir da versão 3.5, permite a configuração de *proxy* para o protocolo HTTPS. O Squid, nesse caso, atua como uma espécie de *man-in-the-middle* entre a máquina cliente e o servidor remoto, forjando certificados para manter duas conexões criptografadas simultaneamente:

```
Cliente ⇐ ⇒ Squid ⇐ ⇒ Servidor Remoto
```

Assim, os dados passam em claro por dentro do próprio *proxy*.

A configuração desse módulo extrapola o escopo desta sessão, mas deixamos aqui nossa recomendação do mesmo para leitura futura.