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

Sessão 0 — 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.

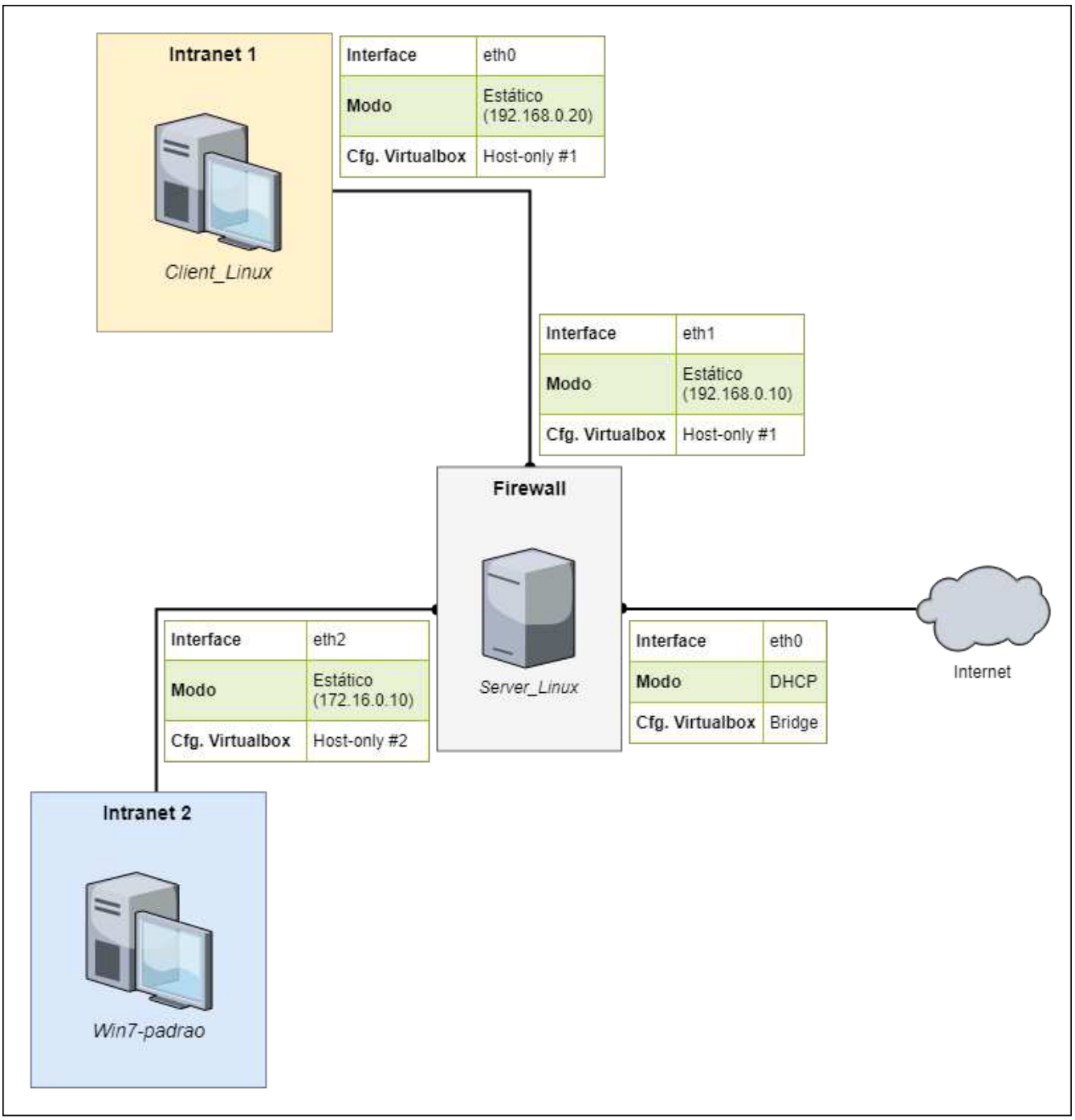

Figura 1: Topologia de rede do curso

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:

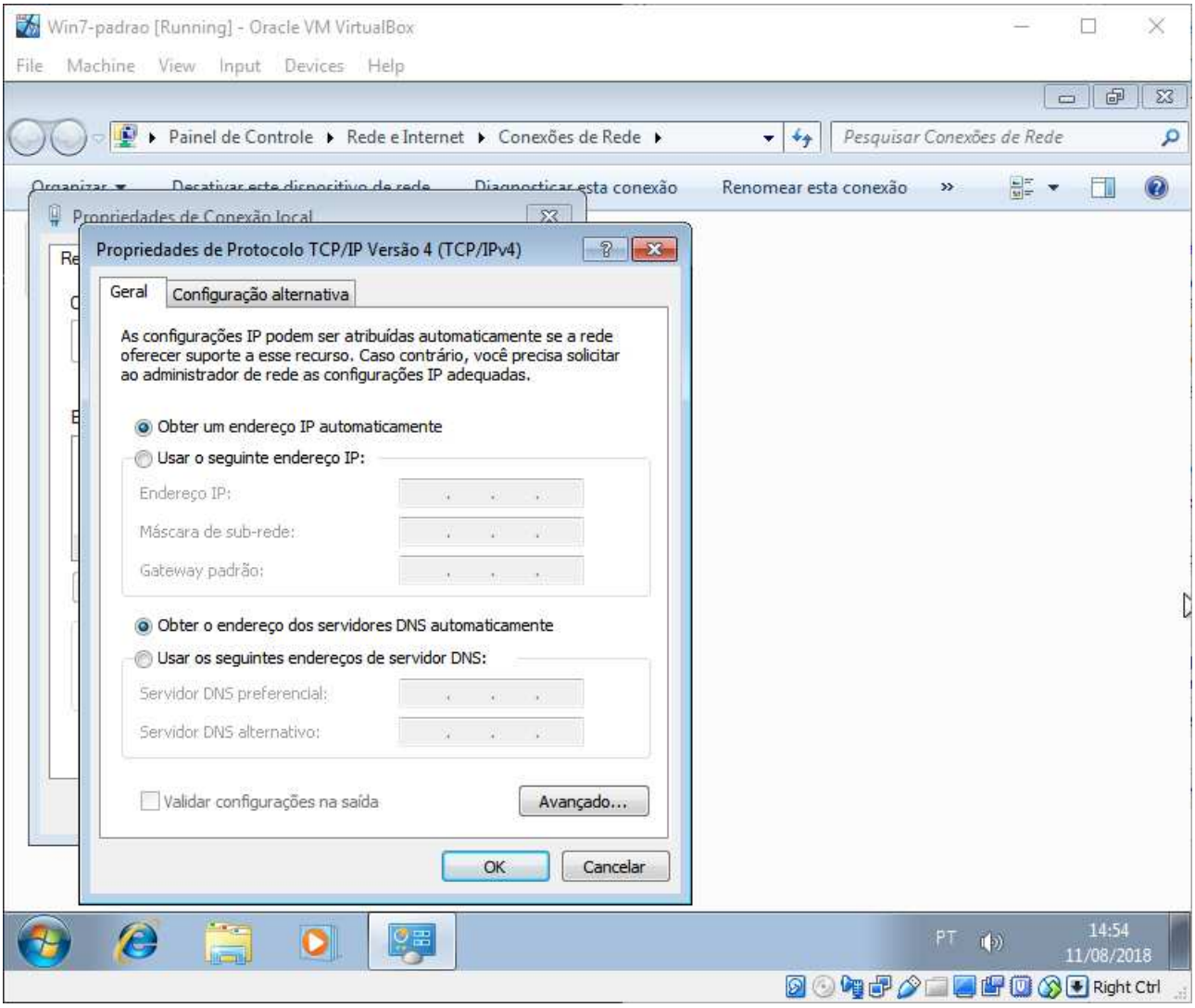

Figura 2: Configuração de rede da máquina Win7-padrao

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
```

Sessão 1 — 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*?

   ```
   $ ls -lha / | grep 'tmp$'
   drwxrwxrwt  7 root root 4,0K Ago  7 01:01 tmp
   ```

   O sticky bit está definido: `t`.

2. Considerando que há permissão de escrita no diretório para todos, o que o impediria de remover um arquivo de outra pessoa?

   ```
   $ rm -f /tmp/file_root
   rm: não foi possível remover “/tmp/file_root”: Operação não permitida
   ```

   Com o sticky bit definido somente o dono de um arquivo pode removê-lo.

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*?

   ```
   $ ls -lha /home/ | egrep ' aluno$'
   drwxr-xr-x  2 aluno  aluno  4,0K Ago  7 01:45 aluno
   ```

   Como visto acima, `2`. Esse número não é fixo, mas depende do conteúdo do diretório. Um diretório recém criado, que não tenha nenhum conteúdo possui dois *links* (um referente ao próprio diretório e outro referente à entrada especial ".").

2. Crie o arquivo `arqses1ex3` no seu diretório *home*. Utilize o comando *touch*.

```
$ touch ~/arqses1ex3
$ ls /home/aluno
arqses1ex3
```

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.

```
$ mytemp=$(mktemp) && ls -lha ~/arqses1ex3 | tee nlinks && awk '{print $2}' nlinks
> $mytemp && mv $mytemp nlinks
-rw-r--r-- 1 aluno aluno 0 Ago  7 01:52 /home/aluno/arqses1ex3
$ cat nlinks
1
```

   O arquivo `arqses1ex3` possui apenas um link.

4. Verifique se mudou o número de *links* do seu diretório *home*.

```
$ ls -lha /home/ | egrep ' aluno$'
drwxr-xr-x  2 aluno  aluno  4,0K Ago  7 02:05 aluno
```

   O número de *links* continuou o mesmo.

5. Crie um diretório com o nome de `dirses1ex3`, também no seu diretório *home*.

```
$ mkdir /home/aluno/dirses1ex3
$ ls ~
arqses1ex3  dirses1ex3  nlinks
```

6. Mais uma vez, verifique o número de *links* do seu diretório *home*. Ele mudou? Você saberia dizer por quê?

```
$ ls -lha /home/ | egrep ' aluno$'
drwxr-xr-x  3 aluno  aluno  4,0K Ago  7 02:11 aluno
```

   O número de *links* aumentou em uma unidade, por conta de entrada especial ".." presente no diretório `/home/aluno/dirses1ex3`, que aponta para o diretório `/home/aluno`.

7. Qual o número de links do diretório `dirses1ex3`?

```
$ ls -lha ~ | egrep ' dirses1ex3$'
drwxr-xr-x 2 aluno aluno 4,0K Ago  7 02:11 dirses1ex3
```

   Como visto acima, 2.

8. Verifique qual opção deve ser passada ao comando `ls` para que ele liste as informações do diretório `dirses1ex3` e não o seu conteúdo.

```
$ ls -dl ~/dirses1ex3/
drwxr-xr-x 2 aluno aluno 4096 Ago  7 02:11 /home/aluno/dirses1ex3/
```

   Devem ser passadas as opções `-d` e `-l`.

9. Você saberia explicar por que o número de *links* do diretório `dirses1ex3` é maior que um?

   Os dois *links* são relativos ao próprio diretório. Um aponta o caminho direto `/home/aluno` → `/home/aluno/dirses1ex3` e o outro corresponde à entrada especial ".", presente no próprio diretório `/home/aluno/dirses1ex3`.

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`.

```
$ ln /home/aluno/arqses1ex3 /home/aluno/hosts.hard
$ ls ~
arqses1ex3  dirses1ex3  hosts.hard  nlinks
```

2. Verifique agora o número de links do arquivo `arqses1ex3` e compare com aquele obtido na atividade 2. Explique a diferença.

```
$ ls -lha /home/aluno/arqses1ex3 | awk '{print $2}'
2
$ cat nlinks
1
```

   O número de *links* foi aumentado de 1 para 2 devido à criação do *link* `hosts.hard`.

3. Crie um *symbolic link* para o arquivo `arqses1ex3`, que deverá se chamar `hosts.symbolic`.

```
$ ln -s /home/aluno/arqses1ex3 /home/aluno/hosts.symbolic
$ ls
arqses1ex3  dirses1ex3  hosts.hard  hosts.symbolic  nlinks
```

4. O número de *links* do arquivo `arqses1ex3` aumentou?

```
$ ls -lha /home/aluno/arqses1ex3
-rw-r--r-- 2 aluno aluno 0 Ago  7 01:52 /home/aluno/arqses1ex3
```

   Não, não aumentou.

5. Caso não tenha aumentado, por que isso aconteceu, considerando que foi criado um *link* para ele?

   Porque o *symbolic link* aponta para outro *inode*.

6. Qual o tamanho do arquivo *hosts.symbolic*?

```
$ du -sb ~/hosts.symbolic
22      /home/aluno/hosts.symbolic
```

   Como mostrado acima, 22 bytes.

7. Você percebe alguma correlação entre o tamanho e o arquivo para o qual ele aponta?

```
$ ls -d /home/aluno/arqses1ex3 | tr -d '\n' | wc -c
22
```

   Esse tamanho representa o número de caracteres presentes no *path* completo do arquivo original linkado, sendo cada caractere representado por 1 byte.

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*?

   Pelo *hard link* conseguiríamos acessar o conteúdo do arquivo normalmente. Já pelo *symbolic link* não conseguiríamos acessar o conteúdo do arquivo, uma vez que o mesmo é somente uma referência para o arquivo original.

2. Depois de responder a essas questões, remova o arquivo criado (`arqses1ex3`) e verifique se as suas respostas estão corretas.

```
$ rm arqses1ex3

$ ls -l hosts.hard
-rw-r--r-- 1 aluno aluno 0 Ago  7 01:52 hosts.hard
$ ls -l hosts.symbolic
lrwxrwxrwx 1 aluno aluno 22 Ago  7 02:38 hosts.symbolic -> /home/aluno/arqses1ex3

$ cat hosts.hard
$ cat hosts.symbolic
cat: hosts.symbolic: Arquivo ou diretório não encontrado
```

As respostas acima 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.

```
$ touch ~/arqses1ex6
$ ln ~/arqses1ex6 /tmp/link-arqses1ex6
ln: failed to create hard link “/tmp/link-arqses1ex6” => “/home/aluno/arqses1ex6”:
Link entre dispositivos inválido

$ df -h | sed -n '1!p' | egrep -v '^tmpfs |^udev ' | awk '{printf "%s\t mounted on:
%s\n", $6, $1}'
/        mounted on: /dev/sda1
/tmp     mounted on: /dev/sda6
```

Não foi possível criar o *hard link*, porque o diretório `/tmp` está em outra partição.

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`.

```
$ ln -s /usr/bin /home/aluno/link-bin ; cd link-bin
$ pwd
/home/aluno/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`?

```
$ cd ..
$ pwd
/home/aluno
```

   Porque o *link* simbólico é apenas uma referência para o diretório.

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.

```
$ chmod 555 /home/aluno
$ ls -ld /home/aluno
dr-xr-xr-x 3 aluno aluno 4096 Ago  7 03:38 /home/aluno
```

2. Verifique as permissões associadas ao arquivo `arqses1ex6`. Você tem permissão para escrever nesse arquivo? O grupo tem?

```
$ ls -lha ~/arqses1ex6
-rw-r--r-- 1 aluno aluno 0 Ago  7 02:55 /home/aluno/arqses1ex6
```

   Somente o dono do arquivo tem permissão para escrever no mesmo.

3. Tente remover o arquivo `arqses1ex6`. Você conseguiu? Em caso negativo, você sabe explicar o motivo?

```
$ rm ~/arqses1ex6
rm: não foi possível remover “/home/aluno/arqses1ex6”: Permissão negada
```

   Não, porque o diretório `/home/aluno` está sem permissão de escrita para o dono.

4. Modifique as permissões do arquivo `arqses1ex6` de forma a retirar a permissão de escrita para o dono e colocá-la para o grupo.

```
$ chmod 464 ~/arqses1ex6
$ ls -ld ~/arqses1ex6
-r--rw-r-- 1 aluno aluno 0 Ago  7 02:55 /home/aluno/arqses1ex6
```

5. Com o uso de redirecionamento, tente copiar o conteúdo do seu diretório *home* para dentro do arquivo `arqses1ex6`.

```
$ ls -lha /home/aluno > /home/aluno/arqses1ex6
-bash: /home/aluno/arqses1ex6: Permissão negada
```

   Apresentou erro de permissão de gravação no diretório por parte do dono.

6. Torne a colocar a permissão para escrita no seu diretório *home* para o dono.

```
$ chmod 755 /home/aluno
$ ls -ld ~
drwxr-xr-x 3 aluno aluno 4096 Ago  7 03:38 /home/aluno
```

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.

```
$ umask 000 ; touch arq1ses1ex9 ; mkdir dir1ses1ex9
$ umask 002 ; touch arq2ses1ex9 ; mkdir dir2ses1ex9
$ umask 003 ; touch arq3ses1ex9 ; mkdir dir3ses1ex9
$ umask 023 ; touch arq4ses1ex9 ; mkdir dir4ses1ex9
$ umask 222 ; touch arq5ses1ex9 ; mkdir dir5ses1ex9
$ umask 022 ; touch arq6ses1ex9 ; mkdir dir6ses1ex9

$ ls -lha /home/aluno | egrep 'arq[1-6]ses1ex9|dir[1-6]ses1ex9'
-rw-rw-rw- 1 aluno aluno    0 Ago  7 03:50 arq1ses1ex9
-rw-rw-r-- 1 aluno aluno    0 Ago  7 03:50 arq2ses1ex9
-rw-rw-r-- 1 aluno aluno    0 Ago  7 03:50 arq3ses1ex9
-rw-r--r-- 1 aluno aluno    0 Ago  7 03:52 arq4ses1ex9
-r--r--r-- 1 aluno aluno    0 Ago  7 03:52 arq5ses1ex9
-rw-r--r-- 1 aluno aluno    0 Ago  7 03:52 arq6ses1ex9
drwxrwxrwx 2 aluno aluno 4,0K Ago  7 03:50 dir1ses1ex9
drwxrwxr-x 2 aluno aluno 4,0K Ago  7 03:50 dir2ses1ex9
drwxrwxr-- 2 aluno aluno 4,0K Ago  7 03:50 dir3ses1ex9
drwxr-xr-- 2 aluno aluno 4,0K Ago  7 03:52 dir4ses1ex9
dr-xr-xr-x 2 aluno aluno 4,0K Ago  7 03:52 dir5ses1ex9
drwxr-xr-x 2 aluno aluno 4,0K Ago  7 03:52 dir6ses1ex9
```

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?

   O comando `umask` trabalha de forma diferente com arquivos e diretórios. Por motivos de segurança um um novo arquivo nunca recebe a permissão de execução quando da sua criação.

Sessão 2 — 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.

   ```
   $ ls -ld /etc/gshadow /etc/shadow
   -rw-r----- 1 root shadow  666 Ago  5 16:52 /etc/gshadow
   -rw-r----- 1 root shadow 1125 Ago  5 16:51 /etc/shadow
   ```

   O aluno deve verificar se os arquivos `/etc/shadow` e `/etc/gshadow` existem.

2. Crie uma conta para você no sistema, seguindo os passos descritos na aula teórica e no material didático.
   - Editar o arquivo `/etc/group` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha ("x");
     - GID;
     - Membros do grupo.

     ```
     marcelo:x:1001:
     ```

   - Editar o arquivo `/etc/gshadow` e inserir uma nova linha com os parâmetros relativos ao grupo do novo usuário:
     - Nome do grupo;
     - Senha criptografada do grupo ("!");
     - Administradores do grupo;
     - Membros do grupo.

     ```
     marcelo:!::
     ```

   - Editar o arquivo `/etc/passwd` e inserir uma nova linha com os parâmetros relativos à conta do novo usuário:

    - Nome do usuário;
    - Senha ("x");
    - UID;
    - GID;
    - GECOS: campo com comentários informativos do usuário;
    - Diretório *home*;
    - Shell de login.

    ```
    marcelo:x:1001:1001:,,,:/home/marcelo:/bin/bash
    ```

- Editar o arquivo `/etc/shadow` e inserir uma nova linha os parâmetros relativos à conta do novo usuário:
    - Nome do usuário;
    - Senha criptografada: inserir valor "*", que será alterado a seguir;
    - *last_change*: número de dias desde a última alteração de senha;
    - *minimum*: número mínimo de dias até que senha possa ser alterada novamente;
    - *maximum*: número máximo de dias até que a senha deva ser alterada;
    - *warning*: número de dias para aviso de expiração de senha;
    - *inactive*: número de dias após expiração em que a senha será aceita;
    - *expire*: data para expiração da senha.

    ```
    marcelo:*:16846:0:99999:7:::
    ```

- Definir uma senha para a nova conta, utilizando o comando `passwd`:

    ```
    # passwd marcelo
    ```

- Copiar os arquivos de inicialização contidos no diretório `/etc/skel` para o diretório *home* do usuário.

    ```
    # cp -r /etc/skel /home/marcelo
    ```

- Alterar o usuário e grupo donos dos arquivos na pasta *home* do novo usuário:

    ```
    # chown -R marcelo.marcelo /home/marcelo
    ```

- Configurar a *quota* de disco para o usuário, se o sistema utilizar *quotas*.
- Testar se a conta foi criada corretamente, fazendo login no sistema e verificando se o

diretório corrente é o diretório *home* do usuário, definido no arquivo `/etc/passwd`.

- O *script shell* abaixo mostra uma maneira como os comandos executados manualmente nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

usage() {
  echo "  Usage: $0 -u USER -p PASSWORD"
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":u:p:" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    p)
      pass=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }
[ -z $pass ] && { echo "  [*] No password?"; usage; }

if egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User exists!"
  exit 1
fi

lastgid=$( getent group | grep -v 'nogroup' | cut -d':' -f3 | sort -n | tail -n1
)
((lastgid++))

echo "$user:x:$lastgid:" >> /etc/group
echo "$user:!::" >> /etc/gshadow

lastuid=$( getent passwd | grep -v 'nobody' | cut -d':' -f3 | sort -n | tail -n1
)
((lastuid++))
```

```
echo "$user:x:$lastuid:$lastgid:,,,:/home/$user:/bin/bash" >> /etc/passwd

salt="$( cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 8 | head -n 1 )"
hpass="$( mkpasswd -m sha-512 -S $salt -s <<< $pass )"
echo "$user:$hpass:16842:0:99999:7:::" >> /etc/shadow

cp -r /etc/skel /home/$user
chown -R ${user}.${user} /home/$user
```

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`.

```
# useradd instrutor -d /instrutor -m -e 2018-08-07
# passwd instrutor
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

Ao usar o comando `useradd`, o shell escolhido pelo sistema é o `/bin/sh`, por padrão. Para alterar o shell do usuário, pode-se editar o arquivo `/etc/passwd` diretamente, ou executar o comando `chsh`, mostrado abaixo:

```
# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/sh

# chsh instrutor
Mudando o shell de login para instrutor
Informe o novo valor ou pressione ENTER para aceitar o padrão
        Shell de Login [/bin/sh]: /bin/bash

# getent passwd | grep '^instrutor:'
instrutor:x:1002:1002::/home/instrutor:/bin/bash
```

4. O comando `useradd` não é uma boa opção para informar a senha do usuário. Por quê?

   Porque a senha criptografada deve ser digitada diretamente na linha de comando, podendo ser lida posteriormente via logs ou histórico do shell.

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.

O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

IFS=':'
useradd="$( which useradd )"
groupadd="$( which groupadd )"


usage() {
  echo "  Usage: $0 -f NEWUSERS_FILE"
  echo "  File syntax: username:password:uid:gid:gecos:homedir:shell"
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":f:" opt; do
  case "$opt" in
    f)
      file=${OPTARG}
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $file ] && { echo "  [*] No file?"; usage; }

while read username password uid gid gecos homedir shell; do
  if egrep "^${username}:" /etc/passwd &> /dev/null; then
    echo "  [*] User $username already exists, skipping..."
  elif getent passwd | cut -d':' -f3 | grep "$uid" &> /dev/null; then
    echo "  [*] UID $uid already exists, skipping..."
  elif getent group | cut -d':' -f3 | grep "$gid" &> /dev/null; then
    echo "  [*] GID $gid already exists, skipping..."
  else
    hpass="$( mkpasswd -m sha-512 -s <<< $pass )"
    $groupadd $username -g $gid
    $useradd $username -p $( mkpasswd -m sha-512 -s <<< $password)  -u $uid -g $gid
-c "$gecos" -d $homedir -s $shell
    cp -r /etc/skel $homedir
    chown -R $username:$username $homedir
  fi
done < "$file"
```

Um arquivo de entrada com sintaxe válida para o *script* acima seria como se segue:

```
usuario1:rnpesr:1101:1101::/home/usuario1:/bin/bash
usuario2:rnpesr:1102:1102::/home/usuario2:/bin/bash
usuario3:rnpesr:1103:1103::/home/usuario3:/bin/bash
```

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.

```
$ ssh instrutor@localhost
instrutor@localhost's password:

$ id
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor)
$ pwd
/instrutor
$ ls -la
total 8
drwxr-xr-x  2 instrutor instrutor 4096 Ago  7 14:42 .
drwxr-xr-x 23 root      root      4096 Ago  7 14:42 ..
```

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.

```
# chfn -w 6198765432 instrutor
# finger -l instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/sh
Office Phone: 619-876-5432
Last login Tue Aug  7 14:44 (-03) on pts/1 from localhost
No mail.
No Plan.
```

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`.

```
# addgroup grupoteste
Adicionando grupo 'grupoteste' (GID 1003) ...
Concluído.
```

2. Liste o arquivo `/etc/group` e anote o `GID` que foi atribuído ao grupo criado.

```
# getent group | egrep '^grupoteste:' | cut -d':' -f3
1003
```

3. Aproveite para observar, no arquivo `/etc/group`, quais são os outros grupos existentes no sistema. Qual o grupo associado ao usuário `root`?

```
# getent group | grep root
root:x:0:
```

   O grupo `root`, que é o grupo primário do superusuário do sistema.

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`.

```
# usermod -g grupoteste instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1003:,,6198765432,:/instrutor:/bin/sh
```

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?

```
$ gpasswd -a instrutor grupoteste
gpasswd : Permissão negada.
```

   Não, porque somente o usuário `root` pode cadastrar administradores em um grupo. Os comandos para viabilizar essa tarefa seriam:

```
# gpasswd -A aluno grupoteste
# logout

$ whoami
aluno
$ gpasswd -a instrutor grupoteste
Adicionando usuário instrutor ao grupo grupoteste
```

6. Altere novamente o grupo primário do usuário `instrutor` para o grupo `instrutor`.

```
# usermod -g instrutor instrutor
# getent passwd | egrep '^instrutor:'
instrutor:x:1002:1002:,,6198765432,:/instrutor:/bin/sh
```

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.

   Inserir após o último caractere ":" na linha referente ao grupo `grupoteste`, o *username* do usuário `instrutor`.

```
# getent group | egrep '^grupoteste:'
grupoteste:x:1003:instrutor
# groups instrutor
instrutor : instrutor grupoteste
```

2. Agora, utilize um comando apropriado para inserir nesse mesmo grupo o usuário criado para você no primeiro item da atividade 1.

```
# groups marcelo
marcelo : marcelo

# usermod -a -G grupoteste marcelo
# groups marcelo
marcelo : marcelo grupoteste
```

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.

```
# passwd -l instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
Permission denied, please try again.
```

2. Agora desbloqueie a conta e faça o teste de acesso para verificar se sua alteração surtiu efeito.

```
# passwd -u instrutor
passwd: informação de expiração de senha alterada.

# ssh instrutor@localhost
instrutor@localhost's password:
$ pwd
/instrutor
```

Também pode-se utilizar o comando `# usermod -U USERNAME` para atingir o mesmo objetivo.

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.

   Em ordem, deve-se executar as atividades espelho das que foram feitas anteriormente, quais sejam:

   - Remover entradas referente à conta nos arquivos:
     - `/etc/group`
     - `/etc/gshadow`
     - `/etc/passwd`
     - `/etc/shadow`
   - Remover o diretório *home* do usuário;
   - Remover as configurações de *quota*, caso tenham sido configuradas anteriormente.
   - O *script shell* abaixo mostra uma maneira como os comandos executados manualmente nesta atividade poderiam ser automatizados por um administrador de sistemas:

```
#!/bin/bash

BACKUP_DIR="/root/user_backups"

usage() {
  echo "  Usage: $0 -u USER [-b]"
  echo "  Use [-b] to backup user dir to /root before deletion."
  exit 1
}


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

backup=false
while getopts ":u:b" opt; do
  case "$opt" in
    u)
      user=${OPTARG}
      ;;
    b)
      backup=true
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $user ] && { echo "  [*] No user?"; usage; }

if ! egrep "^${user}:" /etc/passwd &> /dev/null; then
  echo "  [*] User does not exist!"
  exit 1
fi

homedir=$( getent passwd | egrep "^$user:" | cut -d':' -f6 )

if $backup; then
  [ ! -d $BACKUP_DIR ] && mkdir $BACKUP_DIR
  tar czf $BACKUP_DIR/${user}.tar.gz $homedir
fi
rm -rf /home/$user

sed -i "/^$user:/d" /etc/group
sed -i "/^$user:/d" /etc/gshadow
sed -i "/^$user:/d" /etc/passwd
sed -i "/^$user:/d" /etc/shadow
```

```
# remove user from secondary groups
sed -r -i "s/,?${user},?/,/ ; s/:,/:/ ; s/,$//" /etc/group
```

2. Certifique-se de que esse usuário foi realmente excluído do sistema, utilizando um dos comandos que fornecem informações sobre os usuários.

```
# finger marcelo
finger: marcelo: no such user.
```

3. Faça um backup de seus dados de modo que o instrutor possa ter sobre eles o mesmo tipo de acesso que você.

   O *script* apontado no primeiro item desta atividade já faz o backup de arquivos (via opção `-b`). Caso o usuário tenha sido removido sem que seu *home* tenha sido apagado (por exemplo, via comando `userdel`), pode-se fazer o backup dos dados da seguinte forma:

```
# tar czf /instrutor/marcelo.tar.gz /home/marcelo && rm -rf /home/marcelo
tar: Removendo `/' inicial dos nomes dos membros
# ls /instrutor/
marcelo.tar.gz
```

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.

```
$ id instrutor
uid=1002(instrutor) gid=1002(instrutor) grupos=1002(instrutor),1003(grupoteste)

$ finger instrutor
Login: instrutor                        Name:
Directory: /instrutor                   Shell: /bin/sh
Office Phone: 619-876-5432
Last login Tue Aug  7 15:45 (-03) on pts/1 from localhost
No mail.
No Plan.
```

   O comando `id` mostra os grupos do usuário e seu UID enquanto o comando `finger` mostra informações como: diretório *home*, shell, *username*, GECOS, terminal utilizado pelo usuário, etc.

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.

```
# userdel -r instrutor
# getent passwd | egrep '^instrutor:'
# getent group  | egrep ',?instrutor,?'
#
```

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.

```
# chgrp grupoteste /etc/passwd
# ls -ld /etc/passwd
-rw-r--r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

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.

```
# chmod 664 /etc/passwd
# ls -ld /etc/passwd
-rw-rw-r-- 1 root grupoteste 1612 Ago  7 16:12 /etc/passwd
```

   Alternativamente, pode-se usar também o comando `# chmod g+w /etc/passwd` para atingir o mesmo objetivo.

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?

- Crie o grupo controladoria:

```
# addgroup controladoria
Adicionando grupo 'controladoria' (GID 1002) ...
Concluído.
```

- Crie a conta do novo funcionário e defina o grupo `controladoria` como seu grupo primário:

```
# useradd -m -g controladoria funcionario
# ls -lha /home/ | egrep ' funcionario$'
drwxr-xr-x  2 funcionario controladoria 4,0K Ago  7 16:22 funcionario
```

- Crie o diretório `/home/controladoria`:

```
# mkdir /home/controladoria
# chgrp controladoria /home/controladoria
# chmod g+w /home/controladoria/
# ls -lha /home/ | egrep ' controladoria$'
drwxrwxr-x  2 root        controladoria 4,0K Ago  7 16:24 controladoria
```

- Habilite o *sticky bit* para o diretório `/home/controladoria`, de forma que todos os membros do grupo `controladoria` possam criar arquivos ali, mas apenas o dono de cada arquivo possa apagá-los:

```
# chmod +t /home/controladoria/
# ls -lha /home/ | egrep ' controladoria$'
drwxrwxr-t  2 root        controladoria 4,0K Ago  7 16:24 controladoria
```

- Mova os arquivos do antigo funcionário para o diretório `/home/controladoria`:

```
# cp -a /home/antigo_funcionario /home/controladoria
# ls /home/controladoria
antigo_funcionario
```

  Redefina as permissões dos arquivos do antigo funcionário:

```
# chown -R root.controladoria /home/controladoria
```

- Remova a conta do antigo funcionário:

```
# userdel -r antigo_funcionario
```

- Oriente o novo funcionário para que ele só armazene os arquivos relacionados ao setor de controladoria no diretório `/home/controladoria`, e seus arquivos pessoais em `/home/funcionario`.

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
```

Sessão 3 — 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.

```
# ps aux | sed -n '1!p' | wc -l
71
```

2. Faça um *script* que liste o número de processo que cada usuário está executando.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

users=( $( ps aux | awk '{ if (NR>1) print $1 }' | sort | uniq ) )

for (( i=0; i<${#users[@]}; i++ )); do
  nproc=$( ps aux | grep "${users[$i]}" | wc -l )
  echo "User ${users[$i]} has $nproc active processes"
done
```

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?

```
$ echo -e "PID: $$\nPPID: $PPID"
PID: 1016
PPID: 1015
```

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.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

pinit=( $( ps -eo ppid,comm | egrep -e "^ *1 " | sort | uniq | awk {'print $2'} ) )
pinit_count=${#pinit[@]}

echo "$pinit_count processes started by init (1):"

for (( i=0; i<$pinit_count; i++ )); do
  echo "  ${pinit[$i]}"
done
```

3) Estados dos processos

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

```
$ ps aux | awk '{print $8}' | sort | uniq -c | sort -n | tac
     24 S
     23 S<
     16 Ss
      4 S+
      1 STAT
      1 Ssl
      1 Ss+
      1 SN
      1 R+
      1 D+
```

O estado mais frequente é *sleep*, porque apenas um processo pode estar sendo executado pela CPU em um dado momento.

4) Alternando a execução de processos

1. Execute o comando `$ sleep 1000` diretamente do terminal.

```
$ sleep 1000
```

2. Pare o processo e mantenha-o em memória.

   Basta digitar a combinação de teclas `CTRL + Z`.

```
$ sleep 1000
^Z
[1]+  Parado
```

3. Liste os processos parados.

```
$ jobs
[1]+  Parado                 sleep 1000
```

4. Coloque-o em *background*.

```
$ bg
[1]+ sleep 1000 &
$ jobs
[1]+  Executando             sleep 1000 &
```

5. Verifique se o comando sleep 1000 está rodando.

```
$ ps ax | egrep 'sleep 1000$'
 2178 pts/0    S      0:00 sleep 1000
```

6. É possível cancelar a execução desse comando quando ele está rodando em *background*? Caso seja possível, faça-o.

```
$ kill 2178
$ ps ax | egrep 'sleep 1000$'
[1]+  Terminado              sleep 1000
```

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`.

   Na primeira console, execute:

```
$ passwd
Mudando senha para aluno.
Senha UNIX (atual):
```

   Antes de digitar a senha, abra uma segunda console como `root` e execute:

```
# ps -eo user,ruser,comm | egrep '^USER | passwd$'
USER     RUSER    COMMAND
root     aluno    passwd

# which passwd
/usr/bin/passwd
# ls -lh /usr/bin/passwd
-rwsr-xr-x 1 root root 53K Mai 17  2017 /usr/bin/passwd
```

Os valores são diferentes porque o binário `passwd` possui o bit *SUID* ativado. O `RUID` (*real uid*) é do usuário que está executando o comando e o EUID (*effective uid*) é o do usuário `root`, que é o dono do arquivo.

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
   ```

   Basta executar o comando `# ps lax` e buscar o processo relevante, verificando o valor da quinta coluna. Em uma única linha e de forma mais específica, podemos fazer:

   ```
   # ps lax | egrep ' sleep 1000$' | awk '{print $5}'
   2289 5
   ```

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?

   ```
   # nice -n -5 sleep 1000 &
   [2] 2312
   # ps lax | egrep ' sleep 1000$' | awk '{print $3, $5}'
   2289 5
   2312 15
   ```

   A prioridade diminuiu, porque quanto maior o valor na coluna `PRI`, menor a prioridade do processo.

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?

```
$ crontab -l
no crontab for aluno
```

2. Que opção deve ser usada para editar o seu arquivo de `crontab`?

```
$ crontab -e
no crontab for aluno - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
No modification made
```

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.

   Utilize o comando `$ crontab -e` para editar o `crontab` e inserir a linha:

   ```
   30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null
   ```

2. Como verificar se a configuração foi feita corretamente?

   ```
   $ crontab -l | egrep -v '^#'
   30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null
   ```

3. Qual o requisito fundamental para garantir que a ação programada será executada?

   O daemon do `cron` deve estar em execução e a sintaxe do `crontab`, incluindo a linha de comando utilizada, deve estar correta.

4. Há como confirmar se a mensagem foi efetivamente enviada, sem consultar o destinatário?

   Verifique no arquivo `/var/log/syslog` se a tarefa foi executada no horário correto com sucesso. Você deve ver uma entrada do tipo:

   ```
   /var/log/syslog:Aug  7 17:40:01 cliente CRON[2524]: (aluno) CMD (COMMAND)
   ```

   Dependendo da distribuição Linux em uso, as mensagens relativas ao `cron` podem estar em `/var/log/syslog`, `/var/log/cron.log`, `/var/log/daemon.log` ou outros arquivos. Verifique na documentação do fabricante/mantenedor.

5. Dê dois exemplos de utilização desse mecanismo para apoiar atividades do administrador de sistemas.

   Podemos, por exemplo, utilizar o `cron` para agendamento de backups e limpeza de diretórios temporários.

6. Faça um script que liste os arquivos sem dono do sistema e envie a lista por e-mail ao usuário root.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto, com a característica adicional de guardar os logs enviados por e-mail em um diretório dentro do *home* do `root`:

```
#!/bin/bash

LOGDIR="/root/nouser_logs"

[ ! -d $LOGDIR ] && mkdir $LOGDIR

curlog="$LOGDIR/nouser_$( date +%Y%m%d ).log"
find / -nouser -print > $curlog
mail -s "Files without ownership for $( date )" root < $curlog
```

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.

   Logado como usuário `root`, digite o comando `# crontab -e` para editar o `crontab` e insira a linha a seguir:

```
30 22 * * 1-5 /root/scripts/find_nouser.sh
```

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?

```
$ crontab -l | egrep -v '^#'
30 20 * * * mail fulano@dominio -s "Mensagem de teste" < /dev/null

$ crontab -r
$ crontab -l
no crontab for aluno
```

10) Entendendo o comando exec

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

```
# whoami
root
# exec ls -l /mnt/
total 0

$ whoami
aluno
```

   O shell corrente foi finalizado. Sempre que um comando é executado, um novo processo é criado. Já quando um comando é executado como argumento do comando `exec`, a imagem do

shell corrente é substituída pela do processo invocado, e quando esse processo encerra sua execução já não há mais shell de retorno.

Sessão 4 — 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?

   ```
   $ cat /etc/fstab | grep -v '^#' | awk '{print $3}' | sort | uniq
   ext4
   swap
   udf,iso9660
   ```

   Alternativamente, verifique no arquivo `/etc/fstab` o campo *type* de cada partição.

2. Qual partição ocupa maior espaço em disco?

   ```
   $ df -m | awk 'NR>1' | awk '{print $2,$1}' | sort -n | tac | head -n1
   29910 /dev/sda1
   ```

   Alternativamente, verifique com o comando `df -h` a partição que possui o maior número de bytes em uso, na coluna "*Used*".

3. Qual é o *device* correspondente à partição raiz?

   ```
   $ df -h | egrep ' /$' | awk '{print $1}'
   /dev/sda1
   ```

   Alternativamente, verifique através do comando `df -h` a linha que possui no campo "*Mounted on*" o caractere `/` e em seguida, nesta mesma linha, verificar o *device* correspondente no campo "*Filesystem*".

4. Os discos do computador que você está utilizando são do tipo `IDE` ou `SCSI`?

```
$ dmesg | egrep 'Attached.*disk'
[   10.310957] sd 1:0:0:0: [sdb] Attached SCSI disk
[   10.358641] sd 0:0:0:0: [sda] Attached SCSI disk
```

Alternativamente, verifique através do comando `df -h`, o campo "*Filesystem*". Discos `IDE` são representados pelos dispositivos `/dev/hda`, `/dev/hdb`, `/dev/hdc`, etc. Discos `SCSI` são representados pelos dispositivos `/dev/sda`, `/dev/sdb`, `/dev/sdc`, etc.

5. A que partição pertence o arquivo `/etc/passwd`?

```
$ df -T /etc/passwd | sed -n '1!p' | awk '{print $1}'
/dev/sda1
```

Alternativamente, verifique através do comando `df` em qual partição se encontra o diretório `/etc`.

6. Você faria alguma crítica em relação ao particionamento do disco do computador que você está utilizando? Como você o reparticionaria?

O aluno deve avaliar o esquema de particionamento adotado e responder à pergunta levando em conta as vantagens obtidas com o particionamento, como isolamento de falhas, ganho de performance, etc.

2) Determinando o espaço utilizado por um diretório

1. Que subdiretório do diretório `/var` ocupa maior espaço em disco?

```
# du -sm /var/* | sort -n | tac | head -n1
97      /var/lib
```

Alternativamente, verifique através do comando `du -mcs /var/*` qual diretório 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%).

O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

parts=( $( df -h | egrep -e "^/dev" | awk {'print $6'} ) )
partusage=( $( df -h | egrep -e "^/dev" | awk {'print $5'} | tr -d % ) )
out="$( mktemp )"

for (( i=0; i<${#parts[@]}; i++ )); do
  if [ ${partusage[$i]} -gt 90 ]; then
    echo -e "Filesystem ${parts[$i]} over ${partusage[$i]}% capacity." >> $out
  fi
done

if  [ -e $out ]; then
  mail -s "Filesystem capacity report" root@localhost < $out
  rm -f $out
fi
```

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.
   - Primeiro, vamos verificar quais discos foram conectados ao sistema durante o *boot*:

     ```
     # dmesg | egrep 'Attached.*disk'
     [   10.310957] sd 1:0:0:0: [sdb] Attached SCSI disk
     [   10.358641] sd 0:0:0:0: [sda] Attached SCSI disk
     ```

   - Vamos checar o estado de uso desses discos, começando pelo `/dev/sda`:

```
# fdisk -l /dev/sda

Disco /dev/sda: 40 GiB, 42949672960 bytes, 83886080 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0x27232fb6

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 62500863 62498816 29,8G 83 Linux
/dev/sda2       62502910 83884031 21381122 10,2G  5 Extended
/dev/sda5       62502912 66406399  3903488  1,9G 82 Linux swap / Solaris
/dev/sda6       66408448 83884031 17475584  8,3G 83 Linux
```

- O disco `/dev/sda` já está sendo utilizado, e aparentemente está cheio. Vamos então verificar o dispositivo `/dev/sdb`:

```
# fdisk -l /dev/sdb

Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
```

- Perfeito, parece estar vazio. Vamos formatá-lo e criar uma única partição Linux ocupando a totalidade do espaço livre:

```
# fdisk /dev/sdb

Bem-vindo ao fdisk (util-linux 2.25.2).
As alterações permanecerão apenas na memória, até que você decida gravá-las.
Tenha cuidado antes de usar o comando de gravação.

A unidade não contém uma tabela de partição conhecida.
Created a new DOS disklabel with disk identifier 0x4fa0acac.

Comando (m para ajuda): o
Created a new DOS disklabel with disk identifier 0xb33d8f79.

Comando (m para ajuda): n
Tipo da partição
   p   primária (0 primárias, 0 estendidas, 4 livre)
   e   estendida (recipiente para partições lógicas)
Selecione (padrão p):

Usando resposta padrão p.
Número da partição (1-4, padrão 1):
Primeiro setor (2048-16777215, padrão 2048):
Último setor, +setores ou +tamanho{K,M,G,T,P} (2048-16777215, padrão 16777215):

Criada uma nova partição 1 do tipo "Linux" e de tamanho 8 GiB.

Comando (m para ajuda): t
Selecionou a partição 1
Código hexadecimal (digite L para listar todos os códigos): 83
O tipo da partição "Linux" foi alterado para "Linux".

Comando (m para ajuda): w
A tabela de partição foi alterada.
Chamando ioctl() para reler tabela de partição.
Sincronizando discos.
```

- Finalmente, vamos verificar se o procedimento produziu o resultado esperado:

```
# fdisk -l /dev/sdb

Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 setores
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0xb33d8f79

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 16777215 16775168   8G 83 Linux
```

2. Formate a partição com o sistema de arquivos `ext4`.

```
# mkfs.ext4 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2096896 4k blocks and 524288 inodes
Filesystem UUID: 2464c725-9356-4abb-8a9f-a2de3d64e7ac
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. Crie um *mount point* chamado `/dados` e monte nele a nova partição.

```
# mkdir /dados
# mount -t ext4 /dev/sdb1 /dados
# mount | egrep '^/dev/sdb1'
/dev/sdb1 on /dados type ext4 (rw,relatime,data=ordered)
```

4. Qual a quantidade de espaço em disco que foi reservada para armazenar os dados dos *inodes*? E da partição em si?

   Para calcular o espaço solicitado, o primeiro passo é descobrir quantos *inodes* foram criados, e qual o tamanho de cada um deles:

```
$ sudo tune2fs -l /dev/sdb1 | egrep -i 'inode count|inode size'
Inode count:              524288
Inode size:               256
```

   Feito isso, basta multiplicar os dois valores e, opcionalmente, mostrar o resultado em um formato mais legível, já que o `tune2fs` mostra o tamanho dos *inodes* em bytes:

```
# s=( $(tune2fs -l /dev/sdb1 | egrep -i 'inode count|inode size' | awk '{print
$3}') ); echo "$(( ${s[0]} * ${s[1]} / 1048576 )) MB"
128 MB
```

5. Cheque a partição criada com o comando apropriado. Que tipos de checagens foram realizados?

```
# umount /dev/sdb1
# e2fsck /dev/sdb1 -fv
e2fsck 1.42.12 (29-Aug-2014)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information

          11 inodes used (0.00%, out of 524288)
           0 non-contiguous files (0.0%)
           0 non-contiguous directories (0.0%)
             # of inodes with ind/dind/tind blocks: 0/0/0
             Extent depth histogram: 3
       70287 blocks used (3.35%, out of 2096896)
           0 bad blocks
           1 large file

           0 regular files
           2 directories
           0 character device files
           0 block device files
           0 fifos
           0 links
           0 symbolic links (0 fast symbolic links)
           0 sockets
------------
           2 files
```

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.

   Deve-se inserir a linha abaixo ao final do arquivo `/etc/fstab`.

   ```
   /dev/sdb1   /dados/   ext4   defaults,errors=remount-ro   0   2
   ```

   Feito isso, reinicie o sistema e verifique a montagem do *filesystem*.

Atualmente, é muito comum sistemas Linux indicarem os *filesystems* no arquivo `/etc/fstab` através de seu UUID (*Universally Unique Identifier*), em lugar de nome de dispositivo, já que a ordem em que os discos são detectados pelo kernel não é determinística — em uma instância de *boot* um disco pode ser detectado como `/dev/sda`, e na próxima, como `/dev/sdb`. Para identificar a partição que acabamos de criar através do seu UUID, siga os passos abaixo:

```
# ls -l /dev/disk/by-uuid/ | egrep 'sdb1$' | awk '{print $9}'
2464c725-9356-4abb-8a9f-a2de3d64e7ac

# uuid="$(ls -l /dev/disk/by-uuid/ | egrep 'sdb1$' | awk '{print
$9}')" ; echo "UUID=$uuid   /dados   ext4   defaults,errors=remount-
ro   0   2" >> /etc/fstab

# egrep ' /dados ' /etc/fstab
UUID=2464c725-9356-4abb-8a9f-a2de3d64e7ac   /dados   ext4
defaults,errors=remount-ro   0   2
```

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.

   Verifique se o pacote `quota` está instalado no sistema com o comando `dpkg -l | grep quota`. Caso não esteja, instale-o usando o `apt-get`:

   ```
   # dpkg -l | grep ' quota '
   # apt-get -y install quota quotatool
   ```

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.

   Insira no arquivo `/etc/fstab` o suporte à *quota* de disco na partição raiz com as opções apropriadas:

   ```
   # grep ' / ' /etc/fstab | grep -v '^#'
   UUID=6d035549-c33d-4f72-a751-1e7ddc602dbe   /   ext4   errors=remount-
   ro,usrquota,grpquota   0   1
   ```

   Feito isso, reinicie o sistema e verifique se o suporte a *quotas* foi habilitado através do comando

`mount`:

```
# mount | egrep '^/dev/sda1'
/dev/sda1 on / type ext4 (rw,relatime,quota,usrquota,grpquota,errors=remount-
ro,data=ordered)
```

3. Crie uma conta de usuário para teste e configure o limite desse novo usuário para 200 MB, utilizando o comando `edquota`.

   Primeiro, vamos criar o usuário. Em seguida, editar seu arquivo de *quota*:

   ```
   # useradd -m pedro
   # edquota -u pedro
   ```

   O comando `edquota` irá invocar um editor (indicado pela variável de ambiente `$EDITOR`) para que as *quotas* sejam ajustadas. Vamos editar os campos *soft* e *hard* da seção *block* do arquivo — note que os valores devem ser informados em `kBytes`. Pode-se, opcionalmente, também setar um limite para *inodes* que o usuário pode criar.

   ```
   Disk quotas for user pedro (uid 1005):
     Filesystem                   blocks       soft       hard     inodes     soft
   hard
     /dev/sda1                        16     100000     200000          4        0
   0
   ```

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?

   ```
   # su - pedro

   $ quota -u
   Disk quotas for user pedro (uid 1005):
        Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
         /dev/sda1      16  100000  200000              4       0       0
   ```

   Na listagem acima, pode-se observar que o usuário `pedro` está utilizando 16 kB de espaço em disco, com um *soft limit* de 100 MB e um *hard limit* de 200 MB.

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?

```
$ pwd
/home/pedro

$ quota -u
Disk quotas for user pedro (uid 1005):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda1      16  100000  200000              4       0       0

$ cp /boot/vmlinuz-3.16.0-6-amd64 ~
$ ls
vmlinuz-3.16.0-6-amd64
$ quota -u
Disk quotas for user pedro (uid 1005):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda1    3116  100000  200000              5       0       0

$ ln -s /boot/vmlinuz-3.16.0-6-amd64 ~/kernel-link
$ ls
kernel-link  vmlinuz-3.16.0-6-amd64
$ quota -u
Disk quotas for user pedro (uid 1005):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda1    3116  100000  200000              6       0       0
```

A forma de contabilização é diferente: o tamanho do link simbólico corresponde apenas ao tamanho em bytes do *path* completo até o arquivo apontado; já o arquivo criado com o comando `cp` possui o mesmo tamanho do arquivo original.

6. Como determinar se o sistema de *quotas* está habilitado na inicialização do sistema? E, se não estiver como habilitá-lo?

   Em sistemas com o sistema de *init* `systemd`, como é o caso do Debian e da maioria das distribuições Linux atuais, podemos usar o comando `# systemctl is-enabled` para determinar o estado de um *daemon* durante a inicialização do sistema:

   ```
   # systemctl is-enabled quota
   enabled
   ```

   Para desabilitar um serviço, basta usar a palavra-chave `disable`. Ao contrário, para habilitá-lo, utilize `enable`:

```
# systemctl disable quota
Synchronizing state for quota.service with sysvinit using update-rc.d...
Executing /usr/sbin/update-rc.d quota defaults
Executing /usr/sbin/update-rc.d quota disable
insserv: warning: current start runlevel(s) (empty) of script `quota' overrides LSB
defaults (S).
insserv: warning: current stop runlevel(s) (0 6 S) of script `quota' overrides LSB
defaults (0 6).
# systemctl is-enabled quota
disabled

# systemctl enable quota
Synchronizing state for quota.service with sysvinit using update-rc.d...
Executing /usr/sbin/update-rc.d quota defaults
insserv: warning: current start runlevel(s) (empty) of script `quota' overrides LSB
defaults (S).
insserv: warning: current stop runlevel(s) (0 6 S) of script `quota' overrides LSB
defaults (0 6).
Executing /usr/sbin/update-rc.d quota enable
# systemctl is-enabled quota
enabled
```

7. Teste a efetividade do sistema de *quotas*:

```
# su - pedro

$ quota -u
Disk quotas for user pedro (uid 1005):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/sda1      20  100000  200000              5       0       0

$ du -sk /boot/vmlinuz-3.16.0-6-amd64
3100    /boot/vmlinuz-3.16.0-6-amd64

$ for i in {1..1000}; do cp /boot/vmlinuz-3.16.0-6-amd64 ~/kernel-$i; done
sda1: warning, user block quota exceeded.
sda1: write failed, user block limit reached.
cp: erro escrevendo “/home/pedro/kernel-65”: Disk quota exceeded
```

   Através do comando acima, o usuário `pedro` conseguiu copiar para seu diretório *home* a imagem do kernel Linux, copiada do `/boot` e com tamanho de 3100 kB, por 64 vezes até que o *hard limit* de *quota* fosse ativado, e novas cópias fossem desabilitadas.

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*.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

for user in $( getent shadow | awk -F: '$2 != "*" && $2 !~ /^!/ { print $1 }' ); do
  edquota -u ${user} -p $1
done
```

Note, no entanto, que apesar de o *script* acima ser minimamente funcional, há alguns parâmetros importantes que não sendo testados no momento:

- O usuário passado como parâmetro para o *script* existe?
- Está sendo removido o usuário `root` da lista de usuários para aplicação de `quota`?
- Está sendo removido o próprio usuário passado como parâmetro da lista de usuários para aplicação de `quota`?

A resposta para todos esses itens, evidentemente, é não. Poderíamos estender o script para fazer essas funções, mas no intuito de mostrar uma abordagem diferente para o problema, veja abaixo uma solução equivalente, mais completa, usando a linguagem Python:

```
#!/usr/bin/python

import os, sys, subprocess, pwd, spwd

if os.geteuid() != 0:
  exit('  Not root?')

if len(sys.argv) <= 1:
  exit('  Usage: ' + sys.argv[0] + ' TEMPLATE_USER')

try:
  pwd.getpwnam(sys.argv[1])
except KeyError:
  exit('No such \'' + sys.argv[1] + '\' user')

qusers = []

for user in pwd.getpwall():
  if user[0] == 'root' or user[0] == sys.argv[1]:
    continue

  phash = spwd.getspnam(user[0]).sp_pwd

  if phash != '*' and not phash.startswith('!'):
    qusers.append(user[0])

for user in qusers:
  subprocess.call(['edquota', '-u', user, '-p', sys.argv[1]])
```

O que você achou da solução acima? Mais fácil, mais difícil ou apenas diferente? Lembre-se, ao atuar como um administrador de redes e sistemas não se deve ficar preso a um único tipo de ferramenta ou solução, mas sim utilizar a melhor alternativa possível para resolver o problema.

Sessão 5 — 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`.

```
# echo "kern.*        -/var/log/kernel.log" >> /etc/rsyslog.conf
# systemctl restart rsyslog.service

# cat /var/log/kernel.log
cat: /var/log/kernel.log: Arquivo ou diretório não encontrado
```

   Mesmo após reiniciar o *daemon* `rsyslog`, o arquivo não será criado de imediato. Para testar o funcionamento da diretiva, precisamos gerar alguma mensagem para a *facility* apropriada:

```
# modprobe lp
# cat /var/log/kernel.log
Aug  9 11:15:45 cliente kernel: [  447.128333] lp: driver loaded but no devices
found
```

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`.

```
# useradd -m aluno2
# passwd aluno2
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

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`.

No servidor `ssh`, execute:

```
# tail -f -n0 /var/log/auth.log
```

De outra máquina, faça login via `ssh` com a conta criada anteriormente:

```
$ ssh aluno2@192.168.0.20
aluno2@192.168.0.20's password:

aluno2@cliente:~$
```

Monitore o que aconteceu no arquivo `/var/log/auth.log`:

```
# tail -f -n0 /var/log/auth.log
Aug  9 11:26:24 cliente sshd[1050]: Accepted password for aluno2 from 192.168.0.254
port 50325 ssh2
Aug  9 11:26:24 cliente sshd[1050]: pam_unix(sshd:session): session opened for user
aluno2 by (uid=0)
```

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.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while read -r line; do
  s=( $( echo $line ) )
  echo -e "Host ${s[1]}: ${s[0]} failed logins"
done < <( grep "(sshd.auth): authentication failure.*rhost=" /var/log/auth.log |
awk '{print $14}' | cut -d'=' -f2 | sort -n | uniq -c )
```

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.

```
$ last | head -n2
aluno2   pts/1        192.168.0.254     Thu Aug  9 11:26 - 11:27  (00:01)
aluno    pts/0        192.168.0.254     Thu Aug  9 11:10   still logged in
```

2. Como você poderia verificar as contas existentes em seu computador que nunca efetuaram login?

```
$ lastlog | grep '**Nunca logou**' | sort
avahi-autoipd                                **Nunca logou**
backup                                       **Nunca logou**
bin                                          **Nunca logou**
daemon                                       **Nunca logou**
Debian-exim                                  **Nunca logou**
funcionario                                  **Nunca logou**
games                                        **Nunca logou**
gnats                                        **Nunca logou**
irc                                          **Nunca logou**
list                                         **Nunca logou**
lp                                           **Nunca logou**
mail                                         **Nunca logou**
man                                          **Nunca logou**
marcelo                                      **Nunca logou**
messagebus                                   **Nunca logou**
news                                         **Nunca logou**
nobody                                       **Nunca logou**
pedro                                        **Nunca logou**
proxy                                        **Nunca logou**
sshd                                         **Nunca logou**
statd                                        **Nunca logou**
sync                                         **Nunca logou**
sys                                          **Nunca logou**
systemd-bus-proxy                             **Nunca logou**
systemd-network                              **Nunca logou**
systemd-resolve                              **Nunca logou**
systemd-timesync                             **Nunca logou**
uucp                                         **Nunca logou**
www-data                                     **Nunca logou**
```

3. Qual a maneira mais fácil de identificar um login remoto efetuado em seu computador?

   Através do comando `last`. A terceira coluna mostra o *host* de origem do login, seja ele local ou remoto:

```
$ last | head -n20 | grep -v '^reboot'
aluno2   pts/1        192.168.0.254     Thu Aug  9 11:26 - 11:27  (00:01)
aluno    pts/0        192.168.0.254     Thu Aug  9 11:10   still logged in
root     tty1                          Thu Aug  9 03:25 - down   (00:00)
aluno    pts/0        192.168.0.254     Thu Aug  9 02:32 - 03:25  (00:53)
aluno    pts/0        192.168.0.254     Thu Aug  9 02:25 - down   (00:05)
aluno    pts/0        192.168.0.254     Thu Aug  9 01:47 - down   (00:37)
root     tty1                          Wed Aug  8 19:05 - down   (00:00)
aluno    pts/0        192.168.0.254     Wed Aug  8 18:19 - 19:05  (00:46)
root     tty1                          Tue Aug  7 18:18 - down   (00:00)
aluno    pts/0        192.168.0.254     Tue Aug  7 17:56 - 18:17  (00:21)
aluno    pts/1        192.168.0.254     Tue Aug  7 17:07 - 17:15  (00:07)
instruto pts/1        localhost         Tue Aug  7 15:45 - 16:01  (00:15)
instruto pts/1        localhost         Tue Aug  7 14:44 - 14:46  (00:01)
instruto pts/1        localhost         Tue Aug  7 14:42 - 14:42  (00:00)
instruto pts/1        localhost         Tue Aug  7 14:39 - 14:39  (00:00)
```

4. Faça um *script* que mostre o tempo total que cada usuário ficou logado no sistema utilizando as informações obtidas com o comando `last`.

   O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash

users=( $( last -w | egrep '(tty|pts)' | awk '{print $1}' | sort | uniq ) )

for user in "${users[@]}"; do
  times=( $( last -w | egrep "^$user " | egrep '(tty|pts)' | egrep -v 'still logged
in *$' | sed 's/  *$//' | awk -F '[ :()]' '{printf "%s:%s\n", $(NF-2), $(NF-1)}' )
)

  h=0
  m=0
  for time in "${times[@]}" ; do
    s=( $( echo $time | tr ':' ' ' ) )
    ((h+=${s[0]}))
    ((m+=${s[1]}))
  done

  mh=$(($m/60))
  mr=$(($m%60))
  ((h+=$mh))

  echo "User \"$user\" logged time: $h hours, $mr minutes"
done
```

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.

   Na máquina *Server_Linux*, edite o arquivo `/etc/rsyslog.conf` e descomente as linhas que se seguem. Em seguida, reinicie o serviço do `rsyslog`.

   ```
   # grep -A1 'imudp' /etc/rsyslog.conf
   $ModLoad imudp
   $UDPServerRun 514

   # systemctl restart rsyslog.service
   ```

   Na máquina *Client_Linux*, configure o envio de logs para o servidor remoto editando o arquivo `/etc/rsyslog.conf` e inserindo a linha que se segue ao final do arquivo, substituindo o endereço IP `192.168.0.10` pelo IP da máquina *Server_Linux*. Em seguida, reinicie o serviço do `rsyslog`.

   ```
   # tail -n1 /etc/rsyslog.conf
   *.*                     @192.168.0.10

   # systemctl restart rsyslog.service
   ```

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*.

   Tendo em vista que o evento gerado na máquina *Client_Linux* será de login, o registro deverá ser enviado para o arquivo onde eventos de autenticação são enviados, na *facility* `authpriv`:

   ```
   # grep '^auth,authpriv' /etc/rsyslog.conf
   auth,authpriv.*                 /var/log/auth.log
   ```

   Sabendo que o arquivo a ser monitorado é o `/var/log/auth.log`, usaremos o comando `tail` para fazê-lo:

   ```
   # tail -f -n0 /var/log/auth.log
   ```

   Após gerar um evento de login via `ssh` na máquina *Client_Linux*, imediamente a mesma mensagem aparece replicada nos logs da máquina *Server_Linux*:

```
# tail -f -n0 /var/log/auth.log
Aug  9 15:18:07 cliente sshd[3285]: Accepted password for aluno from 192.168.0.254
port 50854 ssh2
Aug  9 15:18:07 cliente sshd[3285]: pam_unix(sshd:session): session opened for user
aluno by (uid=0)
```

Evidentemente, é muito confuso ter todas as mensagens de log de uma máquina remota sendo colocadas nos mesmos arquivos que registram os eventos do servidor local. Para tratar esses logs com mais clareza, é interessante separar os logs de cada *host* remoto em seus próprios arquivos e pastas para facilitar o processamento e entendimento. A seguinte configuração pode ser útil para atingir esse objetivo.

Primeiro, note que o `rsyslog` inclui arquivos customizados pelo usuário terminados com a extensão `.conf` no diretório `/etc/rsyslog.d`:

```
# grep '^\$IncludeConfig' /etc/rsyslog.conf
$IncludeConfig /etc/rsyslog.d/*.conf
```

Vamos criar um arquivo novo nessa pasta, `/etc/rsyslog.d/client_linux.conf`, indicando um arquivo específico para envio dos logs da máquina *Client_Linux*, e evitar a escrita desses registros em qualquer outro arquivo local (palavra-chave `stop`). Feito isso, basta reiniciar o *daemon* `rsyslog`. Veja abaixo o conteúdo desse arquivo:

```
if $fromhost-ip == '192.168.0.25'     then /var/log/client_linux.log
& stop
```

Pronto! Agora, novos eventos gerados pela máquina *Client_Linux* serão enviados exclusivamente para o arquivo `/var/log/client_linux.log`, sem se misturar com os eventos locais do servidor de logs.

```
# tail -f -n0 /var/log/client_linux.log
Aug  9 15:34:33 cliente sshd[3340]: Accepted password for aluno from
192.168.0.254 port 50902 ssh2
Aug  9 15:34:33 cliente sshd[3340]: pam_unix(sshd:session): session
opened for user aluno by (uid=0)
```

3. Cite três vantagens obtidas com o uso de um servidor de logs.
    - Facilita o gerenciamento dos arquivos de log, já que estão centralizados em um único servidor.
    - Aumenta a segurança no armazenamento dos arquivos de log, pois o servidor pode estar em outra rede, com regras diferenciadas, dificultando o acesso de possíveis invasores.

- Facilita o backup dos arquivos de log.

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`.

```
# tail -n1 /etc/rsyslog.conf
cron.*                  /var/log/cron.log

# systemctl restart rsyslog.service
```

2. Utilize o comando `logger` para testar se a alteração feita no passo anterior produziu o efeito esperado.

```
# logger -p cron.info "teste"

# tail /var/log/cron.log
Aug  9 15:52:26 servidor aluno: teste
```

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.

   No arquivo `/etc/logrotate.conf` estão as configurações globais para o rotacionamento dos arquivos de log. Ao configurar o rotacionamento de um arquivo ou um grupo de logs podemos editar diretamente esse arquivo ou, opcionalmente, incluir novas configurações dentro do diretório `/etc/logrotate.d`.

```
# grep '^include' /etc/logrotate.conf
include /etc/logrotate.d

# ls /etc/logrotate.d/
apt  aptitude  dpkg  exim4-base  exim4-paniclog  iptraf  rsyslog
```

   Vamos criar um arquivo `/etc/logrotate.d/cron` para configurar os aspectos de rotacionamento de logs desse arquivo de acordo com os parâmetro especificados no exercício, com o seguinte conteúdo:

```
/var/log/cron.log
{
  rotate 2
  monthly
  missingok
  notifempty
  delaycompress
  compress
  create 640 root adm
  postrotate
    systemctl reload cron.service > /dev/null
  endscript
}
```

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`.

   Primeiro, vamos instalar o pacote:

   ```
   # apt-get install logwatch
   ```

   A seguir, vamos copiar o modelo do arquivo de configuração em `/usr/share/logwatch/default.conf/logwatch.conf` para o diretório `/etc/logwatch/conf`:

   ```
   # cp /usr/share/logwatch/default.conf/logwatch.conf /etc/logwatch/conf/
   ```

   Edite o arquivo para que o período e opções de envio fiquem de acordo com o solicitado pela atividade. Abaixo mostramos o conteúdo do arquivo `/etc/logwatch/conf/logwatch.conf`, excluídas linhas de comentário:

   ```
   LogDir = /var/log
   TmpDir = /var/cache/logwatch
   Output = mail
   Format = text
   Encode = none
   MailTo = root
   MailFrom = Logwatch
   Range = All
   Detail = Low
   Service = All
   mailer = "/usr/sbin/sendmail -t"
   ```

Lembre-se de criar o diretório `/var/cache/logwatch`, que ainda não existe:

```
# mkdir /var/cache/logwatch
```

Finalmente, observe que por padrão o Debian já habilita a execução diária do `logwatch` através de um *script* instalado pelo próprio pacote no diretório `/etc/cron.daily`:

```
# ls /etc/cron.daily/ | grep 'logwatch'
00logwatch

# cat /etc/cron.daily/00logwatch | grep -v '^#' | sed '/^$/d'
test -x /usr/share/logwatch/scripts/logwatch.pl || exit 0
/usr/sbin/logwatch --output mail
```

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.

   Primeiro, vamos instalar o `swatch` via `apt-get`:

   ```
   # apt-get install swatch
   ```

   A configuração do `swatch` é um tanto quanto arcana, mas a página de manual do programa (`$ man 1p swatch`) nos dá algum direcionamento através da seção *CONFIGURATION EXAMPLE*. Um dos requisitos é criar um arquivo de configuração com a expressão regular que casa com o erro de autenticação do daemon do `sshd`. Primeiro, precisamos conhecer o formato da mensagem:

   ```
   Aug  9 16:39:56 servidor sshd[4113]: Failed password for aluno from 192.168.0.254
   port 51230 ssh2
   ```

   Outro ponto de atenção é a tentativa de `su` para o usuário `root` com falha, possivelmente por senha incorreta. Vamos verificar o formato da mensagem de log:

   ```
   Aug  9 16:46:29 servidor su[4175]: FAILED su for root by aluno
   ```

   Sabendo os formatos objetivados, vamos agora elaborar expressões regulares que casem com os padrões acima, extraiam informação relevante, e executem uma ação apropriada — enviar e-mail de notificação ao usuário `root` em caso de violação desses padrões. Abaixo mostramos o conteúdo do arquivo `/etc/swatch.conf`:

```
watchfor /^(.*sshd\[[0-9]*\]\: Failed password for [A-Za-z0-9]* from ([0-9:.]*).*)/
  exec "echo '$1' | mail root -s '[swatch][ssh]:\ $2' "
  echo

watchfor /^(.*su\[[0-9]*\]\: FAILED su for root by ([A-Za-z0-9]*))/
  exec "echo '$1' | mail root -s '[swatch][su]:\ $2' "
  echo
```

Vamos rodar o `swatch` manualmente e testar se os padrões estão sendo capturados. Serão realizadas duas ações de violação — um login `ssh` com senha incorreta e uma tentativa de `su` para `root` com senha incorreta.

```
# swatch --tail-file=/var/log/auth.log --config-file=/etc/swatch.conf --pid
-file=/var/run/swatch.pid

*** swatch version 3.2.3 (pid:5011) started at Qui Ago  9 17:29:51 -03 2018

Aug  9 17:32:35 servidor sshd[5093]: Failed password for aluno from 192.168.0.254
port 51460 ssh2
Aug  9 17:32:43 servidor su[5117]: FAILED su for root by aluno
```

Aparentemente, tudo funcionou. Vamos verificar se os e-mails estão sendo de fato enviados:

```
$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/aluno": 2 messages 2 new
>N  1 root@servidor.emp  Thu Aug 09 17:32   16/705   [swatch][ssh]: 192.168.0.254
 N  2 root@servidor.emp  Thu Aug 09 17:32   16/663   [swatch][su]: aluno
& 1
Message 1:
From root@servidor.empresa.com.br Thu Aug 09 17:32:35 2018
Envelope-to: root@servidor.empresa.com.br
Delivery-date: Thu, 09 Aug 2018 17:32:35 -0300
To: root@servidor.empresa.com.br
Subject: [swatch][ssh]: 192.168.0.254
From: root <root@servidor.empresa.com.br>
Date: Thu, 09 Aug 2018 17:32:35 -0300

Aug  9 17:32:35 servidor sshd[5093]: Failed password for aluno from 192.168.0.254
port 51460 ssh2

& 2
Message 2:
From root@servidor.empresa.com.br Thu Aug 09 17:32:43 2018
Envelope-to: root@servidor.empresa.com.br
Delivery-date: Thu, 09 Aug 2018 17:32:43 -0300
To: root@servidor.empresa.com.br
Subject: [swatch][su]: aluno
From: root <root@servidor.empresa.com.br>
Date: Thu, 09 Aug 2018 17:32:43 -0300

Aug  9 17:32:43 servidor su[5117]: FAILED su for root by aluno
```

Excelente! Para que o `swatch` não tenha que ser iniciado manualmente, e continue operando mesmo após o reinício do sistema, é necessário que ele possua um *initscript* correspondente. Infelizmente, a versão instalada pelo apt-get não disponibiliza tal facilidade nem em formato legado (no diretório `/etc/init.d`) nem em arquivo de serviço para o `systemd` (que ficam no diretório `/etc/systemd/system`).

Felizmente, é relativamente fácil criar um arquivo de serviço para o `systemd` manualmente. Abaixo mostramos o conteúdo do arquivo `/etc/systemd/system/swatch.service`:

```
[Unit]
Description=Swatch Log Monitoring Daemon
After=syslog.target network.target auditd.service sshd.service

[Service]
ExecStart=/usr/bin/swatch --config-file=/etc/swatch.conf --tail-file
=/var/log/auth.log --pid-file=/var/run/swatch.pid --daemon
ExecStop=/bin/kill -s KILL $(cat /var/run/swatch.pid)
Type=forking
PIDFile=/var/run/swatch.pid

[Install]
WantedBy=multi-user.target
```

Uma vez criado, deve-se instruir o `systemd` a carregar o arquivo:

```
# systemctl daemon-reload
```

Pronto! Agora é possível habilitar/desabilitar o `swatch` durante o *boot* do sistema, e iniciar/parar/reiniciar e verificar o estado do serviço normalmente:

```
# systemctl enable swatch.service
Created symlink from /etc/systemd/system/multi-user.target.wants/swatch.service to
/etc/systemd/system/swatch.service.

# systemctl is-enabled swatch.service
enabled

# systemctl start swatch.service

# systemctl status swatch.service
● swatch.service - Swatch Log Monitoring Daemon
   Loaded: loaded (/etc/systemd/system/swatch.service; enabled)
   Active: active (running) since Qui 2018-08-09 17:37:57 -03; 4s ago
  Process: 5216 ExecStart=/usr/bin/swatch --config-file=/etc/swatch.conf --tail
-file=/var/log/auth.log --pid-file=/var/run/swatch.pid --daemon (code=exited,
status=0/SUCCESS)
 Main PID: 5218 (/usr/bin/swatch)
   CGroup: /system.slice/swatch.service
           ├──5218 /usr/bin/swatch --config-file=/etc/swatch.conf --tail
-file=/var/log/auth...
           └──5219 /usr/bin/tail -n 0 -F /var/log/auth.log

Ago 09 17:37:57 servidor systemd[1]: Starting Swatch Log Monitoring Daemon...
Ago 09 17:37:57 servidor systemd[1]: PID file /var/run/swatch.pid not readable
(yet?) a...rt.
Ago 09 17:37:57 servidor systemd[1]: Started Swatch Log Monitoring Daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

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.).

   Primeiro, vamos instalar o `logcheck` via `apt-get`:

   ```
   # apt-get install logcheck
   ```

   O `logcheck` já vem com envio de e-mails habilitado por padrão, então a única configuração necessária é alterar a periodicidade de envio de relatórios. O arquivo `/etc/cron.d/logcheck` vem configurado para envios de hora em hora. Edite a linha:

   ```
   2 * * * *       logcheck    if [ -x /usr/sbin/logcheck ]; then nice -n10
   /usr/sbin/logcheck; fi
   ```

   Alterando-a para:

   ```
   0,30 * * * *       logcheck    if [ -x /usr/sbin/logcheck ]; then nice -n10
   /usr/sbin/logcheck; fi
   ```

O `logcheck` fará um *scan* dos logs de sistema e enviará por e-mail linhas consideradas "interessantes" — note que o programa envia apenas os registros ocorridos desde a sua última execução.

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.
    - Desabilitar o serviço `sshd` no servidor de logs, permitindo acesso somente pela console.
    - Configurar o firewall de *host* para permitir apenas tráfego de pacotes UDP na porta 514.
    - Utilizar uma rede isolada para a troca de mensagens de log.
    - Desinstalar todos os serviços que não estão sendo utilizados ou são desnecessários à função do servidor.
    - Manter o sistema operacional rigorosamente atualizado.

Sessão 6 — 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`.

```
# apt-get install 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:

```
123456
1234
rnpesr
senha
abacate
```

3. Rode o *password cracker* com o comando `# john -wordlist=/root/dicionario.txt -rules /etc/shadow`.

```
# john -wordlist=/root/dicionario.txt -rules /etc/shadow
Created directory: /root/.john
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (aluno2)
senha            (marcelo)
abacate          (aluno3)
rnpesr           (root)
rnpesr           (aluno)
5g 0:00:00:01 100% 3.676g/s 70.58p/s 352.9c/s 352.9C/s 123456..Abacate9
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

4. Veja o resultado da verificação com o comando `# john -show /etc/shadow`.

```
# john -show /etc/shadow
root:rnpesr:16842:0:99999:7:::
aluno:rnpesr:16842:0:99999:7:::
aluno2:123456:17752:0:99999:7:::
marcelo:senha:17752:0:99999:7:::
aluno3:abacate:17752:0:99999:7:::

5 password hashes cracked, 0 left
```

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.

```
# groupadd corp
# usermod -a -G corp aluno
# groups aluno
aluno : aluno cdrom floppy sudo audio dip video plugdev netdev bluetooth corp
```

2. Entre no sistema a partir da sua conta e crie um diretório chamado `dir_corp`.

```
$ mkdir dir_corp
$ ls
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`.

```
$ chgrp corp ~/dir_corp/
$ chmod 2755 ~/dir_corp/
$ ls -ld dir_corp/
drwxr-sr-x 2 aluno corp 4096 Ago  9 19:15 dir_corp/
```

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`.

```
$ pwd
/home/aluno
$ touch arq1
$ touch dir_corp/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?

```
$ ls -ld arq1
-rw-r--r-- 1 aluno aluno 0 Ago  9 19:19 arq1
$ ls -ld dir_corp/arq2
-rw-r--r-- 1 aluno corp 0 Ago  9 19:19 dir_corp/arq2
```

   O arquivo criado no diretório `/home/aluno/dir_corp/` possui o mesmo grupo dono de seu diretório-pai, pois o mesmo está com o bit SGID definido — isso faz com que qualquer arquivo criado dentro dele tenha o mesmo grupo dono que o próprio diretório, independente do usuário que o tenha criado. Já o arquivo criado no diretório `/home/aluno/` tem o mesmo grupo primário do usuário que o criou, já que este diretório não tem o bit SGID definido.

6. Quais as vantagens desse esquema?

   Esse recurso é útil em diretórios compartilhados, nos quais diversos usuários criam arquivos que precisam ter permissão de escrita e/ou leitura para todos os usuários do grupo do diretório.

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.

   Diversos comandos podem ser utilizados para verificar o uso dos recursos computacionais, dentre os quais podemos destacar: `df`, `du`, `ps`, `top`, `htop`, `free`, `vmstat`, `iostat`, `lsof`, etc.

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?

   A instalação e configuração de *quotas* de disco para usuários é uma excelente maneira de implementar controles nesse sentido.

Sessão 7 — 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.

1. Instale os seguintes aplicativos:

```
# apt-get install bind9 bind9utils
```

2. Ajuste os arquivos de configuração da seguinte forma:
   - `/etc/bind/named.conf.options` — opções do servidor `bind`:

```
options {
  directory "/var/cache/bind";

  forwarders {
    8.8.8.8;
    8.8.4.4;
  };

  dnssec-validation auto;
  auth-nxdomain no;

  allow-transfer { none; };
  allow-query { internals; };
  allow-recursion { internals; };

  listen-on { 127.0.0.1; 192.168.0.10; };
  listen-on-v6 { none; };

  version none;
};
```

- `/etc/bind/named.conf.local` — configurações locais do servidor `bind`:

```
acl internals { 127.0.0.0/8; 192.168.0.0/24; };

zone "empresa.com.br" {
  type master;
  file "/etc/bind/db.empresa.com.br";
};

zone "0.168.192.in-addr.arpa" {
  type master;
  file "/etc/bind/db.0.168.192";
};

include "/etc/bind/zones.rfc1918";
```

- `/etc/bind/db.empresa.com.br` — arquivo de zona do domínio `empresa.com.br`:

```
$TTL 86400 ; (1 day)
$ORIGIN empresa.com.br.
@         IN    SOA           servidor.empresa.com.br. admin.empresa.com.br. (
                2018080900    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            servidor.empresa.com.br.

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

servidor  IN    A             192.168.0.10
email     IN    A             192.168.0.15
cliente   IN    A             192.168.0.20
windows   IN    A             192.168.0.25

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

- `/etc/bind/db.0.168.192` — arquivo de resolução reversa do domínio `empresa.com.br`:

```
$TTL 86400 ; (1 day)
$ORIGIN 0.168.192.in-addr.arpa.
@         IN    SOA           servidor.empresa.com.br. admin.empresa.com.br. (
                2018080900    ;Serial (YYYYMMDDnn)
                14400         ;Refresh (4 hours)
                1800          ;Retry (30 minutes)
                1209600       ;Expire (2 weeks)
                3600          ;Negative Cache TTL (1 hour)
)

@         IN    NS            servidor.empresa.com.br.

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

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

- `/etc/resolv.conf` — configuração de resolução de nomes para o *Server_Linux*:

```
domain empresa.com.br
search empresa.com.br
nameserver 127.0.0.1
```

3. Como a interface de rede `eth0` da máquina *Server_Linux* está configurada para obter endereço via DHCP, o *daemon* `dhclient` irá sobrescrever as alterações que fizemos ao arquivo `/etc/resolv.conf` no próximo *reboot*. Para prevenir isso, podemos ativar o atributo `immutable` do arquivo, impedindo sua alteração:

```
# chattr +i /etc/resolv.conf
```

4. Reinicie o `bind` e verifique por possíveis erros:

```
# systemctl restart bind9.service

# systemctl status bind9.service
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled)
  Drop-In: /run/systemd/generator/bind9.service.d
           └─50-insserv.conf-$named.conf
   Active: active (running) since Qui 2018-08-09 21:23:48 -03; 5s ago
     Docs: man:named(8)
  Process: 14402 ExecStop=/usr/sbin/rndc stop (code=exited, status=0/SUCCESS)
 Main PID: 14406 (named)
   CGroup: /system.slice/bind9.service
           └─14406 /usr/sbin/named -f -u bind

Ago 09 21:23:48 servidor named[14406]: zone 22.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone 16.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone 27.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone 127.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone 26.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone 25.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: zone localhost/IN: loaded serial 2
Ago 09 21:23:48 servidor named[14406]: zone 28.172.in-addr.arpa/IN: loaded serial 1
Ago 09 21:23:48 servidor named[14406]: all zones loaded
Ago 09 21:23:48 servidor named[14406]: running
```

5. Teste os registros com o uso das ferramentas `nslookup` e `dig`:

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

Name:   servidor.empresa.com.br
Address: 192.168.0.10

# dig -x 192.168.0.25 +noquestion

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> -x 192.168.0.25 +noquestion
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5625
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; ANSWER SECTION:
25.0.168.192.in-addr.arpa. 86400 IN     PTR     windows.empresa.com.br.

;; AUTHORITY SECTION:
0.168.192.in-addr.arpa. 86400   IN      NS      servidor.empresa.com.br.

;; ADDITIONAL SECTION:
servidor.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: Thu Aug 09 21:26:41 -03 2018
;; MSG SIZE  rcvd: 129
```

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`.

1. Antes de mais nada, configure o *Server_Linux* para permitir transferência de zona a partir do servidor secundário *Client_Linux*:

```
# egrep '^ *allow-transfer' /etc/bind/named.conf.options
  allow-transfer { 192.168.0.20; };

# systemctl restart bind9.service
```

2. A seguir, instale o servidor DNS `bind` na máquina *Client_Linux*:

```
# apt-get install bind9 bind9utils
```

3. Ajuste os arquivos de configuração da seguinte forma:
   - `/etc/bind/named.conf.options` — opções do servidor `bind`:

```
options {
  directory "/var/cache/bind";

  forwarders {
    192.168.0.10;
  };

  dnssec-validation auto;
  auth-nxdomain no;

  allow-transfer { none; };
  allow-query { internals; };
  allow-recursion { internals; };

  listen-on { 127.0.0.1; 192.168.0.20; };
  listen-on-v6 { none; };

  version none;
};
```

   - `/etc/bind/named.conf.local` — configurações locais do servidor `bind`:

```
acl internals { 127.0.0.0/8; 192.168.0.0/24; };

zone "empresa.com.br" {
  type slave;
  file "/etc/bind/sec/db.empresa.com.br";
  masters { 192.168.0.10; };
};

zone "0.168.192.in-addr.arpa" {
  type slave;
  file "/etc/bind/sec/db.0.168.192";
  masters { 192.168.0.10; };
};

include "/etc/bind/zones.rfc1918";
```

- `/etc/resolv.conf` — configuração de resolução de nomes para o *Client_Linux*:

```
domain empresa.com.br
search empresa.com.br
nameserver 127.0.0.1
```

4. Observe que iremos escrever os arquivos transferidos no diretório `/etc/bind/sec`, que ainda não existe. Vamos criá-lo e atribuir permissionamento:

```
# mkdir /etc/bind/sec
# chown bind.root /etc/bind/sec
```

5. Reinicie o `bind` e verifique se os arquivos de zona foram transferidos corretamente:

```
# systemctl restart bind9.service

# systemctl status bind9.service -l
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled)
  Drop-In: /run/systemd/generator/bind9.service.d
           └──50-insserv.conf-$named.conf
   Active: active (running) since Qui 2018-08-09 21:41:27 -03; 2s ago
     Docs: man:named(8)
  Process: 5549 ExecStop=/usr/sbin/rndc stop (code=exited, status=0/SUCCESS)
 Main PID: 5553 (named)
   CGroup: /system.slice/bind9.service
           └──5553 /usr/sbin/named -f -u bind

Ago 09 21:41:27 cliente named[5553]: all zones loaded
Ago 09 21:41:27 cliente named[5553]: running
Ago 09 21:41:27 cliente named[5553]: zone empresa.com.br/IN: Transfer started.
Ago 09 21:41:27 cliente named[5553]: transfer of 'empresa.com.br/IN' from
192.168.0.10#53: connected using 192.168.0.20#48366
Ago 09 21:41:27 cliente named[5553]: zone empresa.com.br/IN: transferred serial
2018080900
Ago 09 21:41:27 cliente named[5553]: transfer of 'empresa.com.br/IN' from
192.168.0.10#53: Transfer completed: 1 messages, 12 records, 312 bytes, 0.001 secs
(312000 bytes/sec)
Ago 09 21:41:28 cliente named[5553]: zone 0.168.192.in-addr.arpa/IN: Transfer
started.
Ago 09 21:41:28 cliente named[5553]: transfer of '0.168.192.in-addr.arpa/IN' from
192.168.0.10#53: connected using 192.168.0.20#35160
Ago 09 21:41:28 cliente named[5553]: zone 0.168.192.in-addr.arpa/IN: transferred
serial 2018080900
Ago 09 21:41:28 cliente named[5553]: transfer of '0.168.192.in-addr.arpa/IN' from
192.168.0.10#53: Transfer completed: 1 messages, 8 records, 261 bytes, 0.001 secs
(261000 bytes/sec)

# ls -lh /etc/bind/sec/
total 8,0K
-rw-r--r-- 1 bind bind 569 Ago  9 21:41 db.0.168.192
-rw-r--r-- 1 bind bind 720 Ago  9 21:41 db.empresa.com.br
```

6. Finalmente, teste a resolução de nomes no servidor secundário:

```
# nslookup pop.empresa.com.br 192.168.0.20
Server:         192.168.0.20
Address:        192.168.0.20#53

pop.empresa.com.br      canonical name = email.empresa.com.br.
Name:   email.empresa.com.br
Address: 192.168.0.15

# dig -x 192.168.0.10 +noquestion @192.168.0.20

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> -x 192.168.0.10 +noquestion @192.168.0.20
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30045
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; ANSWER SECTION:
10.0.168.192.in-addr.arpa. 86400 IN     PTR     servidor.empresa.com.br.

;; AUTHORITY SECTION:
0.168.192.in-addr.arpa. 86400   IN      NS      servidor.empresa.com.br.

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

;; Query time: 2 msec
;; SERVER: 192.168.0.20#53(192.168.0.20)
;; WHEN: Thu Aug 09 21:52:36 -03 2018
;; MSG SIZE  rcvd: 121
```

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).

1. Instale os pacotes abaixo no servidor:

```
# apt-get nfs-kernel-server
```

2. Crie o diretório a ser exportado:

```
# mkdir /dados
```

3. Edite o arquivo `/etc/exports` para configurar o compartilhamento da pasta:

```
/dados  192.168.0.20(rw,no_root_squash,async,no_subtree_check)
```

4. Finalmente, exporte o diretório e reinicie o serviço NFS. Teste se o *mapping* de pasta está correto.

```
# exportfs -a

# systemctl restart nfs-kernel-server.service
# systemctl restart nfs-common.service

# showmount -e
Export list for servidor:
/dados 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.

1. Crie o diretório de montagem:

```
# mkdir /mnt/remoto
```

2. Execute a montagem temporária do diretório NFS e verifique seu funcionamento:

```
# mount -t nfs 192.168.0.10:/dados /mnt/remoto/

# mount | grep '^192.168.0.10:/dados '
192.168.0.10:/dados on /mnt/remoto type nfs4
(rw,relatime,vers=4.0,rsize=65536,wsize=65536,namlen=255,hard,proto=tcp,timeo=600,r
etrans=2,sec=sys,clientaddr=192.168.0.20,local_lock=none,addr=192.168.0.10)
```

3. Adicione a linha a seguir ao arquivo `/etc/fstab` para que a montagem se torne automática após o *boot*. Reinicie a máquina e verifique seu funcionamento.

```
# tail -n1 /etc/fstab
192.168.0.10:/dados   /mnt/remoto   nfs   defaults   0   0
```

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.

1. No servidor, execute:

```
# hostname
servidor

# touch /dados/teste
# ls /dados/
teste
```

2. No cliente, verifique e edite o arquivo:

```
# hostname
cliente

# ls /mnt/remoto/
teste
# echo "$( date )" >> /mnt/remoto/teste
```

3. De volta ao servidor, cheque se o arquivo foi editado com sucesso:

```
# hostname
servidor

# cat /dados/teste
Sex Ago 10 14:35:52 -03 2018
```

Sessão 8 — 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`.

1. Instale o OpenLDAP e programas auxiliares que serão utilizados:

```
# apt-get install slapd ldap-utils migrationtools attr libpam-ldap libnss-ldap nscd
```

2. Reconfigure o pacote `slapd`, respondendo as perguntas de acordo com o exposto na tabela acima.

```
# dpkg-reconfigure slapd
```

3. Inicie o *daemon* `slapd` e verifique seu estado.

```
# systemctl start slapd

# systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access
Protocol)
   Loaded: loaded (/etc/init.d/slapd)
   Active: active (running) since Sex 2018-08-10 15:06:46 -03; 3s ago
  Process: 5095 ExecStop=/etc/init.d/slapd stop (code=exited, status=0/SUCCESS)
  Process: 6128 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/slapd.service
           └─6133 /usr/sbin/slapd -h ldap:/// ldapi:/// -g openldap -u openldap
-F /etc/ld...

Ago 10 15:06:46 servidor systemd[1]: Starting LSB: OpenLDAP standalone server
(Lightwei......
Ago 10 15:06:46 servidor slapd[6132]: @(#) $OpenLDAP: slapd  (Jun 14 2018 21:56:48)
$
                                                buildd@x86-csail-01:/build/openldap-
Yko3W...apd
Ago 10 15:06:46 servidor slapd[6133]: slapd starting
Ago 10 15:06:46 servidor systemd[1]: Started LSB: OpenLDAP standalone server
(Lightweig...l).
Ago 10 15:06:46 servidor slapd[6128]: Starting OpenLDAP: slapd.
Hint: Some lines were ellipsized, use -l to show in full.
```

4. Edite o arquivo `/etc/ldap/ldap.conf` com os valores apropriados:

```
# grep '^BASE\|^URI' /etc/ldap/ldap.conf
BASE    dc=empresa,dc=com,dc=br
URI     ldap://127.0.0.1
```

5. Finalmente, consulte a raiz da *search base* do diretório para verificar seu funcionamento.

```
# ldapsearch -x -b 'dc=empresa,dc=com,dc=br' -s base '(ObjectClass=*)'
# extended LDIF
#
# LDAPv3
# base <dc=empresa,dc=com,dc=br> with scope baseObject
# filter: (ObjectClass=*)
# requesting: ALL
#

# empresa.com.br
dn: dc=empresa,dc=com,dc=br
objectClass: top
objectClass: dcObject
objectClass: organization
o: Empresa
dc: empresa

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

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.

```
# grep '^\$DEFAULT_' /etc/migrationtools/migrate_common.ph
$DEFAULT_MAIL_DOMAIN = "empresa.com.br";
$DEFAULT_BASE = "dc=empresa,dc=com,dc=br";
```

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*.

```
# cd /usr/share/migrationtools
# ./migrate_base.pl > /root/base.ldif
# ./migrate_passwd.pl /etc/passwd /root/passwd.ldif
# ./migrate_group.pl /etc/group /root/group.ldif
```

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.

```
# sed -i '/dn: dc=com,dc=br/,/^$/d' /root/base.ldif
# sed -i '/dn: dc=empresa,dc=com,dc=br/,/^$/d' /root/base.ldif

# head -n1 /root/base.ldif
dn: ou=Networks,dc=empresa,dc=com,dc=br
```

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.

```
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/base.ldif
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/passwd.ldif
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/group.ldif
```

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.

```
# ldapsearch -x -b 'dc=empresa,dc=com,dc=br' '(&(cn=*)(objectClass=posixGroup))'
# extended LDIF
#
# LDAPv3
# base <dc=empresa,dc=com,dc=br> with scope subtree
# filter: (&(cn=*)(objectClass=posixGroup))
# requesting: ALL
#

# root, Group, empresa.com.br
dn: cn=root,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: root
gidNumber: 0

(...)

# openldap, Group, empresa.com.br
dn: cn=openldap,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: openldap
gidNumber: 117

# search result
search: 2
result: 0 Success

# numResponses: 58
# numEntries: 57
```

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`

Parâmetro	Valor
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.

1. Instale os *plugins* LDAP para as bibliotes PAM e NSS, bem como programas auxiliares que serão utilizados:

   ```
   # apt-get install ldap-utils libpam-ldap libnss-ldap nscd
   ```

2. Verifique se os arquivos das bibliotecas PAM e NSS foram configurados automaticamente de forma correta pelo gerenciador de pacotes:

   - `/etc/libnss-ldap.conf`:

     ```
     # cat /etc/libnss-ldap.conf | grep -v '^#' | sed '/^$/d'
     base dc=empresa,dc=com,dc=br
     uri ldap://192.168.0.10
     ldap_version 3
     rootbinddn cn=admin,dc=empresa,dc=com,dc=br
     ```

   - `/etc/libnss-ldap.secret`:

     ```
     # cat /etc/libnss-ldap.secret
     rnpesr
     ```

   - `/etc/pam_ldap.conf`:

```
# cat /etc/pam_ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://192.168.0.10
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
pam_password crypt
```

- `/etc/pam_ldap.secret`:

```
# cat /etc/pam_ldap.secret
rnpesr
```

3. Insira as informações sobre o servidor LDAP no arquivo `/etc/ldap/ldap.conf`:

```
# cat /etc/ldap/ldap.conf | grep -v '^#' | sed '/^$/d'
BASE    dc=empresa,dc=com,dc=br
URI     ldap://192.168.0.10
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

4. Configure o `nsswitch` para consultar as bases LDAP em adição às bases locais de usuários e senhas. Se desejar que as bases do LDAP tenham preferência, coloque a palavra-chave `ldap` à frente da palavra `compat`.

```
# cat /etc/nsswitch.conf | grep -v '^#'

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

5. Para que diretórios *home* sejam criados de forma automática se usuários LDAP inexistentes na máquina local logarem no sistema, insira a linha a seguir ao final do arquivo `/etc/pam.d/common-password`:

```
# tail -n1 /etc/pam.d/common-session
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022
```

6. Adiantando-se ao problema futuro que em que o *daemon* `nscd` irá apontar a inexistência do arquivo `/etc/netgroup`, crie-o, vazio:

```
# touch /etc/netgroup
```

7. Finalmente, reinicie o *daemon* `nscd` e verifique se os usuários e grupos remotos do servidor LDAP estão sendo utilizados pelos subsistemas de autenticação local:

```
# systemctl restart nscd

# grep '^openldap:' /etc/passwd
# getent passwd | grep '^openldap:'
openldap:x:111:117:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false

# grep '^openldap:' /etc/group
# getent group | grep '^openldap:'
openldap:x:117:
```

   Observe, acima, que tanto o usuário quanto o grupo `openldap` estão disponíveis na máquina local, muito embora não existam nos arquivos `/etc/passwd` e `/etc/group`. Eles estão sendo obtidos, remotamente, no servidor LDAP *Server_Linux*.

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.

1. Essencialmente, basta repetir os passos do exercício anterior, desta vez na máquina *Server_Linux*:

```
# hostname
servidor

# cat /etc/libnss-ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://127.0.0.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
```

```
# cat /etc/libnss-ldap.secret
rnpesr

# cat /etc/pam_ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://127.0.0.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
pam_password crypt

# cat /etc/pam_ldap.secret
rnpesr

# cat /etc/ldap/ldap.conf | grep -v '^#' | sed '/^$/d'
BASE    dc=empresa,dc=com,dc=br
URI     ldap://127.0.0.1
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt

# cat /etc/nsswitch.conf | grep -v '^#'

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis

# tail -n1 /etc/pam.d/common-session
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022

# ls -ld /etc/netgroup
-rw-r--r-- 1 root root 0 Ago 12 17:20 /etc/netgroup

# systemctl restart nscd
```

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.

1. Primeiro, vamos verificar o formato da entrada de diretório LDAP para um usuário existente:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=aluno)'
dn: uid=aluno,ou=People,dc=empresa,dc=com,dc=br
uid: aluno
cn: aluno
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/aluno
gecos: aluno,,,
```

2. Vamos adicionar o usuário `esr`. Para isso, basta processar a saída do comando acima, substituir com os dados do novo usuário, e enviar como entrada para o comando `ldapadd`, como se segue:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=aluno)' | sed 's/aluno/esr/
; s/1000/5000/' | ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br'
Enter LDAP Password:
adding new entry "uid=esr,ou=People,dc=empresa,dc=com,dc=br"

# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=esr)'
dn: uid=esr,ou=People,dc=empresa,dc=com,dc=br
uid: esr
cn: esr
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 5000
gidNumber: 5000
homeDirectory: /home/esr
gecos: esr,,,
```

3. Excelente! Vamos fazer o mesmo para o grupo:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br'
'(&(cn=aluno)(objectClass=posixGroup))' | sed 's/aluno/esr/ ; s/1000/5000/' |
ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br'
Enter LDAP Password:
adding new entry "cn=esr,ou=Group,dc=empresa,dc=com,dc=br"

# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br'
'(&(cn=esr)(objectClass=posixGroup))'
dn: cn=esr,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: esr
gidNumber: 5000
```

4. Ainda falta configurar a senha do novo usuário. Vamos fazer isso através do comando `ldappasswd`:

```
# ldappasswd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' -S
"uid=esr,ou=People,dc=empresa,dc=com,dc=br"
New password:
Re-enter new password:
Enter LDAP Password:

# ldapsearch -x -LLL -W -D 'cn=admin,dc=empresa,dc=com,dc=br' -b
'dc=empresa,dc=com,dc=br' '(uid=esr)' userPassword
Enter LDAP Password:
dn: uid=esr,ou=People,dc=empresa,dc=com,dc=br
userPassword:: e1NTSEF9ZW1YWkFBQVNWWEh1a2kwQmVRbzdMdkNzSGp0cm9VOEc=
```

5. De volta à máquina *Client_Linux*, vamos verificar se o novo usuário está sendo importado corretamente:

```
# hostname
cliente

# getent passwd | grep '^esr:'
esr:x:5000:5000:esr,,,:/home/esr:/bin/bash
# getent group | grep '^esr:'
esr:*:5000:
```

6. Agora, basta fazer login com o usuário e testar se a criação automática de diretório *home* está funcionando:

```
$ ssh esr@192.168.0.20
esr@192.168.0.20's password:
Creating directory '/home/esr'.

$ hostname
cliente

$ whoami
esr

$ pwd
/home/esr
```

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.

1. O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash


tldap_user() {
  qlu=$( ldapsearch -x -LLL -b "dc=empresa,dc=com,dc=br" "(uid=${1})" | grep
"^uid:" | awk '{print $2}' )
  [ ! -z $qlu ] && return 1 || return 0
}


# $1 ldap_admin, $2 ldap_password, $3: user, $4: pass
r_adduser() {
  if ! tldap_user $3; then
    echo "  [*] LDAP user exists!"
    exit 1
  fi

  lastuid=$( ldapsearch -x -LLL
'(&(objectClass=posixAccount)(uid=*)(!(uid=nobody)))' uidNumber | grep
'^uidNumber:' | awk '{print $2}' | sort -n | tail -n1 )
```

```
  lastgid=$( ldapsearch -x -LLL '(&(objectClass=posixGroup)(cn=*)(!(cn=nogroup)))'
gidNumber | grep  '^gidNumber:' | awk '{print $2}' | sort -n | tail -n1 )

  ((lastuid++))
  ((lastgid++))

  ldapadd -x -D $1 -w $2 << EOF
dn: uid=$3,ou=People,dc=empresa,dc=com,dc=br
uid: $3
cn: $3
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: $lastuid
gidNumber: $lastgid
homeDirectory: /home/$3
gecos: $3,,,

EOF

  ldapadd -x -D $1 -w $2 << EOF
dn: cn=$3,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: $3
gidNumber: $lastgid

EOF

  ldappasswd -x -D $1 -w $2 -s $4 "uid=$3,ou=People,dc=empresa,dc=com,dc=br"
}


# $1 ldap_admin, $2 ldap_password, $3: user
r_deluser() {
  if tldap_user $3; then
    echo "  [*] LDAP user does not exist!"
    exit 1
  fi

  ldapdelete -x -D $1 -w $2 "uid=$3,ou=People,dc=empresa,dc=com,dc=br"
  ldapdelete -x -D $1 -w $2 "cn=$3,ou=Group,dc=empresa,dc=com,dc=br"
}


usage() {
  echo "  Usage: $0 -l LDAP_ADMIN -w LDAP_PASSWD -u USER [-a|-d] [-p PASSWD]"
```

```
  exit 1
}


# - - - main() - - -


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":adu:p:l:w:" opt; do
  case "$opt" in
    l)
      ladmin=${OPTARG}
      ;;
    w)
      lpass=${OPTARG}
      ;;
    u)
      user=${OPTARG}
      ;;
    p)
      pass=${OPTARG}
      ;;
    a)
      uadd=1
      ;;
    d)
      udel=1
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $ladmin ] && { echo "  [*] No LDAP admin?"; usage; }
[ -z $lpass ] && { echo "  [*] No LDAP password?"; usage; }
[ -z $user ] && { echo "  [*] No user?"; usage; }

if [ -z $uadd ] && [ -z $udel ]; then
  echo "  [*] Choose '-a' (add) or '-d' (delete)."
  usage
elif (($uadd)) && (($udel)); then
  echo "  [*] Do not use '-a' (add) and '-d' (delete) simultaneously."
  usage
fi

if (($uadd)) && [ -z $pass ]; then
```

```
  echo "  [*] '-p' (password) mandatory with '-a' (add)."
  usage
fi

(($uadd)) && r_adduser $ladmin $lpass $user $pass
(($udel)) && r_deluser $ladmin $lpass $user
```

Observe que apesar de o *script* ser relativamente complexo, ele ainda não está completo — falta tratar a adição de usuários a grupos secundários no LDAP, bem como sua remoção desses grupos quando de sua deleção.

Sessão 9 — 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`?

1. Instale o servidor DHCP:

```
# apt-get install isc-dhcp-server
```

2. Edite o arquivo de configuração `/etc/dhcp/dhcpd.conf` de acordo com as especificações da atividade:

```
authoritative;
ddns-update-style none;
log-facility local7;

default-lease-time 43200;
max-lease-time 86400;

option domain-name "empresa.com.br";
option domain-search "empresa.com.br";
option domain-name-servers 192.168.0.10, 192.168.0.20;

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.200 192.168.0.250;
  option routers 192.168.0.10;
}
```

3. Para garantir que o servidor DHCP irá escutar por requisições exclusivamente na interface `eth1`, edite o parâmetro `INTERFACES` no arquivo `/etc/default/isc-dhcp-server` como se segue:

```
# cat /etc/default/isc-dhcp-server | grep '^INTERFACES='
INTERFACES="eth1"
```

4. Reinicie o servidor DHCP e verifique que o arquivo `/var/lib/dhcp/dhcpd.leases` está vazio:

```
# systemctl restart isc-dhcp-server.service

# cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.3.1
```

5. Acesse a máquina *Client_Linux* e configure-a para obter configurações de rede via DHCP. Comente a configuração de rede antiga, possibilitando *rollback* rápido caso a atividade não funcione como esperado:

```
# hostname
cliente

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

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
# iface eth0 inet static
#        address 192.168.0.20
#        netmask 255.255.255.0
#        gateway 192.168.0.10
```

6. Reinicie a máquina *Client_Linux* e verifique se a configuração foi propagada corretamente:

```
# hostname
cliente

# ip a s eth0 | grep '^ *inet '
    inet 192.168.0.200/24 brd 192.168.0.255 scope global eth0

# ip r s
default via 192.168.0.10 dev eth0
169.254.0.0/16 dev eth0  scope link  metric 1000
192.168.0.0/24 dev eth0  proto kernel  scope link  src 192.168.0.200

# cat /etc/resolv.conf
domain empresa.com.br
search empresa.com.br.
nameserver 192.168.0.10
nameserver 192.168.0.20
```

7. De volta à máquina *Server_Linux*, verifique o conteúdo do arquivo `/var/lib/dhcp/dhcpd.leases`:

```
# hostname
servidor

# cat /var/lib/dhcp/dhcpd.leases | grep -v '^#\|^$'
lease 192.168.0.200 {
  starts 6 2018/08/11 20:25:01;
  ends 0 2018/08/12 08:25:01;
  tstp 0 2018/08/12 08:25:01;
  cltt 6 2018/08/11 20:25:01;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:e3:16:71;
  client-hostname "cliente";
}
```

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.

1. Edite o arquivo `/etc/dhcp/dhcpd.conf`, inserindo o excerto a seguir ao final do arquivo. A seguir, reinicie o servidor DHCP.

```
# cat dhcpd.conf | awk '/^host Client_Linux/,/^$/'
host Client_Linux {
  option host-name "cliente.empresa.com.br";
  hardware ethernet 08:00:27:e3:16:71;
  fixed-address 192.168.0.20;
}

# systemctl restart isc-dhcp-server.service
```

2. Reinicie a máquina e/ou as interfaces de rede da máquina *Client_Linux* e verifique se a configuração foi propagada corretamente:

```
# hostname
cliente

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

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?

1. Expanda o arquivo de configuração `/etc/dhcp/dhcpd.conf`, incluindo os novos requisitos:

```
authoritative;
ddns-update-style none;
log-facility local7;

default-lease-time 43200;
max-lease-time 86400;

option domain-name "empresa.com.br";
option domain-search "empresa.com.br";
option domain-name-servers 192.168.0.10, 192.168.0.20;

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.200 192.168.0.250;
  option routers 192.168.0.10;
}

subnet 172.16.0.0 netmask 255.255.255.0 {
  range 172.16.0.50 172.16.0.80;
  option routers 172.16.0.10;
}

host Client_Linux {
  option host-name "cliente.empresa.com.br";
  hardware ethernet 08:00:27:e3:16:71;
  fixed-address 192.168.0.20;
}
```

2. Inclua a nova interface de rede na lista de interfaces em que o servidor DHCP irá escutar por requisições:

```
# cat /etc/default/isc-dhcp-server | grep '^INTERFACES='
INTERFACES="eth1 eth2"
```

3. Edite a configuração do `bind` para atender a requisições de resolução de nomes oriundas da rede 172.16.0.0/24:

```
# cat /etc/bind/named.conf.local | grep '^ *acl '
acl internals { 127.0.0.0/8; 192.168.0.0/24; 172.16.0.0/24; };
```

4. Reinicie ambos os serviços de rede:

```
# systemctl restart bind9.service
# systemctl restart isc-dhcp-server.service
```

5. Verifique se a máquina *Win7-padrao* recebeu um endereço IP dentro da faixa esperada:

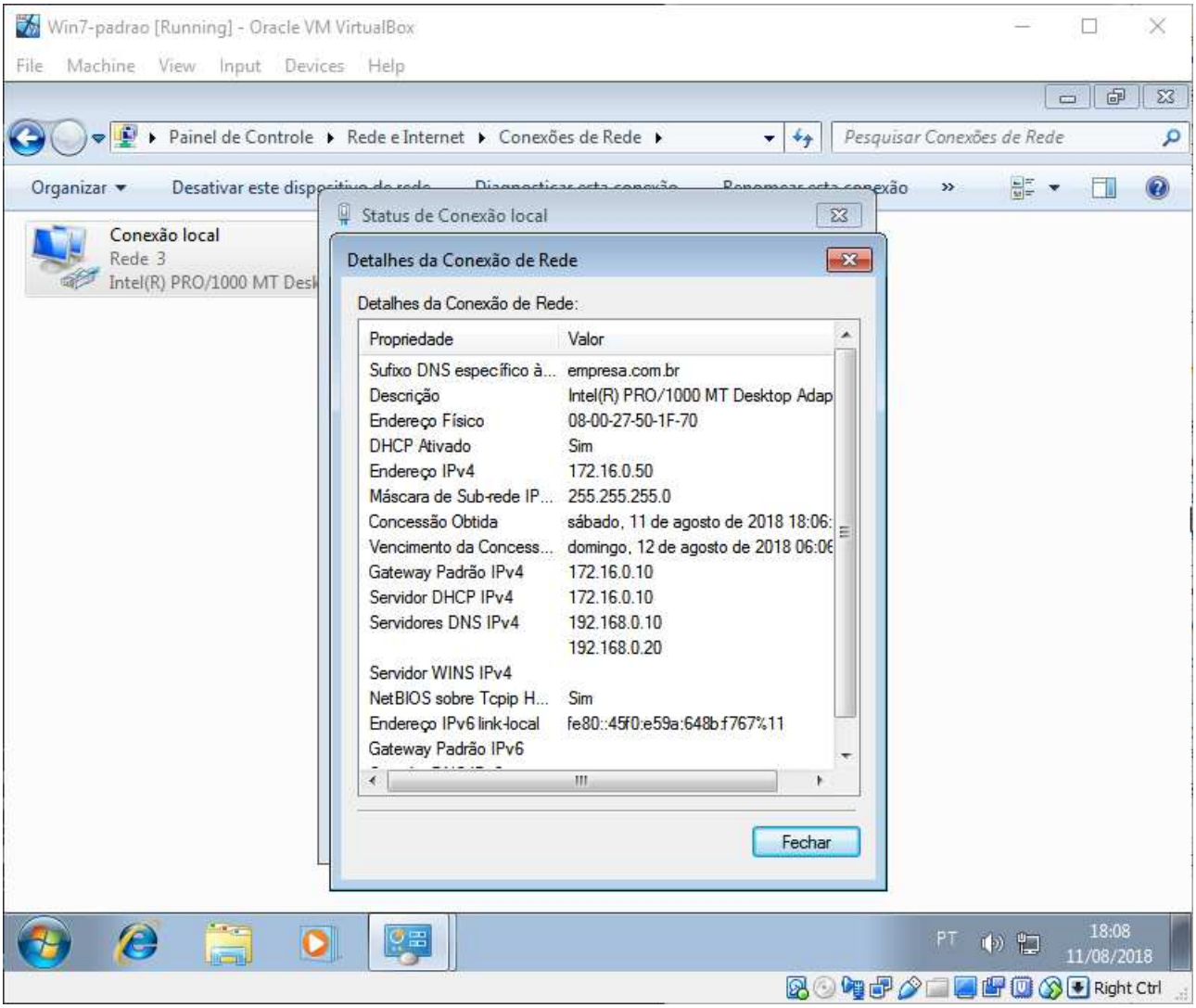

Figura 3: IP recebido via DHCP pelo Windows 7

6. Cheque se a resolução de nomes está operacional:

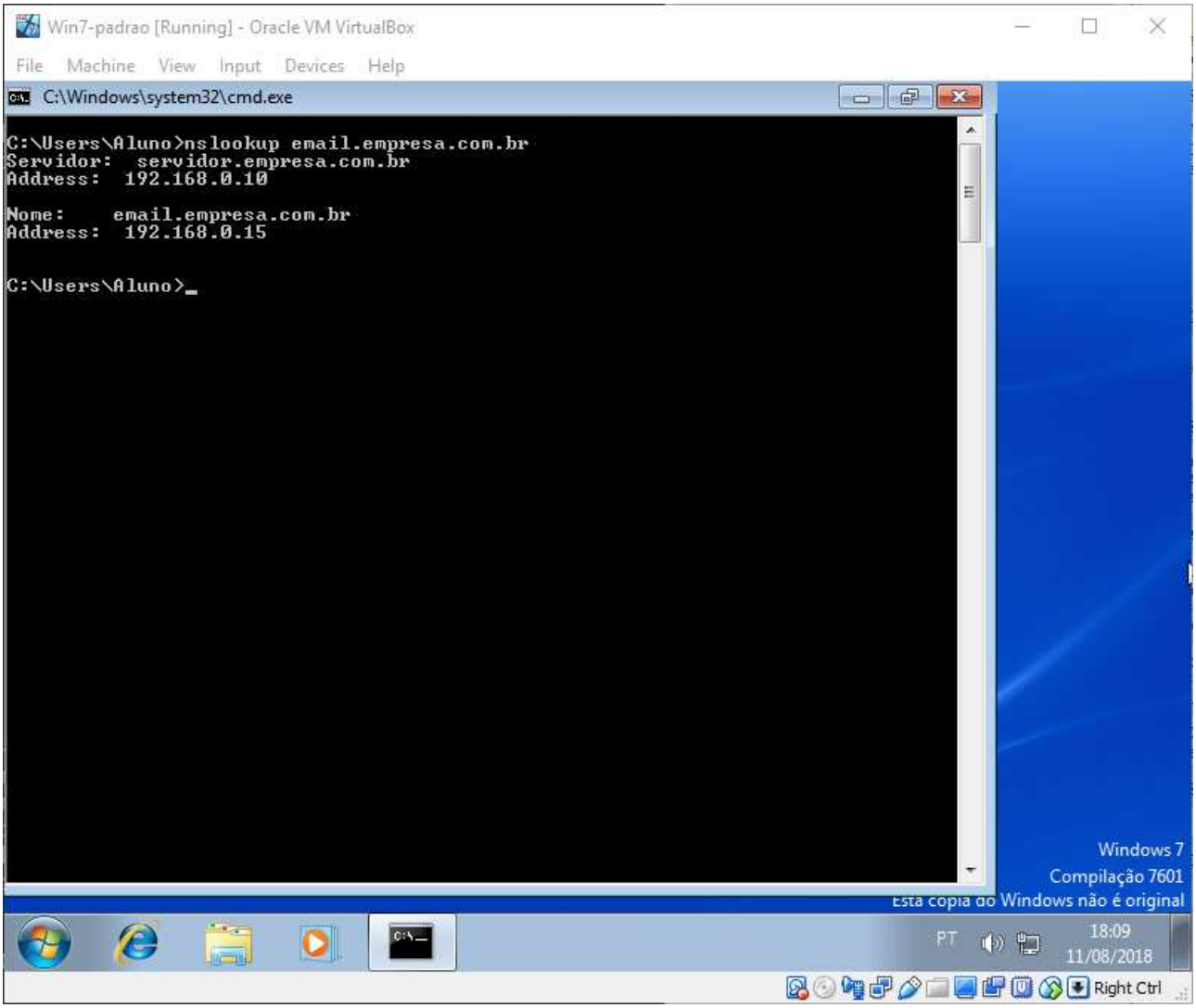

Figura 4: Resolução de nomes no Windows 7

7. Finalmente, tente navegar na internet. Na foto abaixo, acessamos o website https://esr.rnp.br/ :

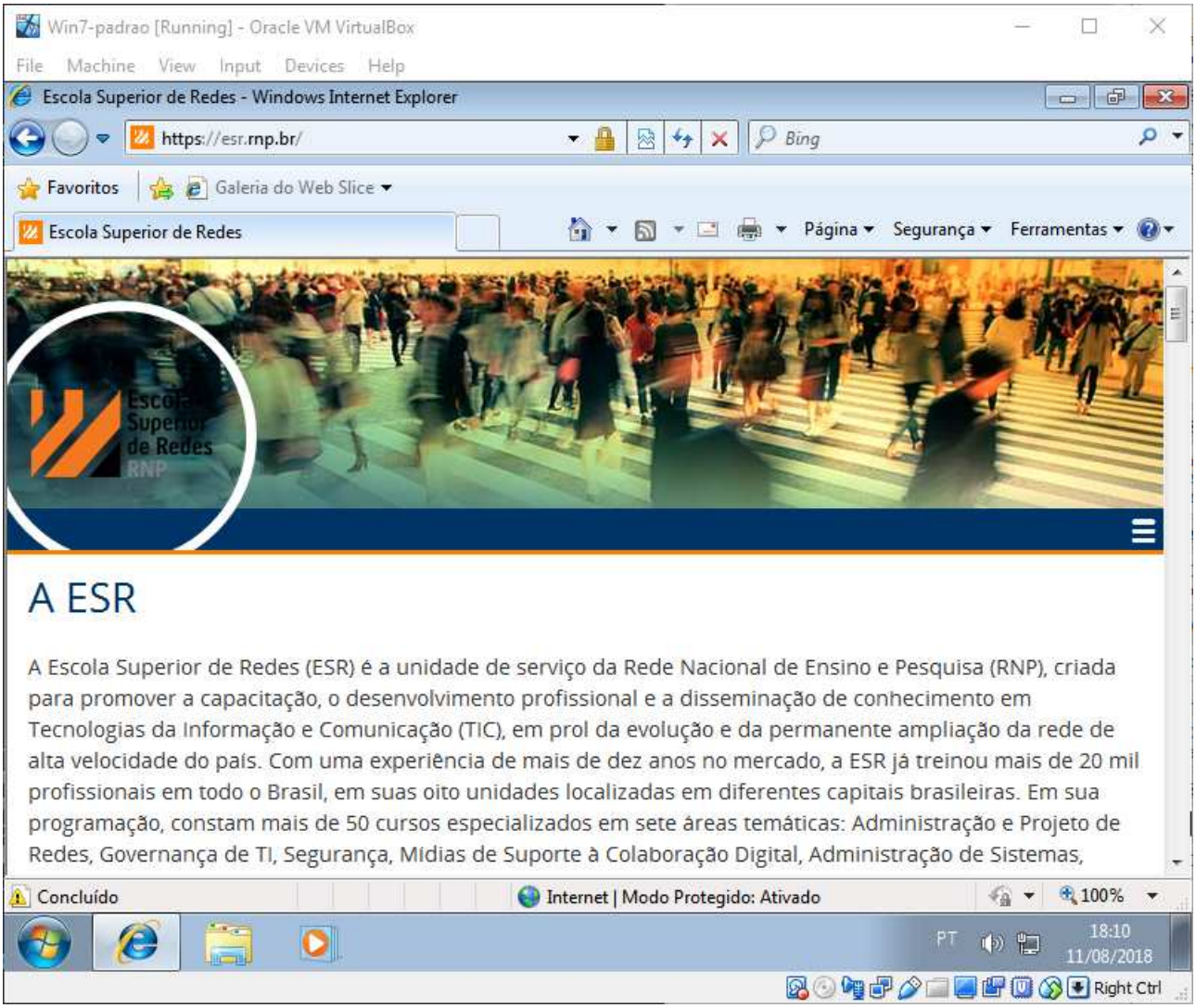

Figura 5: Navegação no Windows 7

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.

1. Instale o servidor FTP:

```
# apt-get install vsftpd
```

2. Edite o arquivo de configuração `/etc/vsftpd.conf` como se segue:

```
allow_writeable_chroot=YES
anonymous_enable=YES
chroot_local_user=YES
connect_from_port_20=YES
dirmessage_enable=YES
ftpd_banner=Servidor FTP SEG12
listen_ipv6=NO
listen=YES
local_enable=YES
local_umask=022
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
secure_chroot_dir=/var/run/vsftpd/empty
ssl_enable=NO
use_localtime=YES
write_enable=YES
xferlog_enable=YES
```

3. A diretiva `pam_service_name=vsftpd` irá processar o arquivo `/etc/pam.d/vsftpd` durante tentativas de login via FTP. A última linha desse arquivo exige que o shell do usuário conste no arquivo `/etc/shells`, que não é o caso do shell inválido `/bin/false`. Para solucionar essa questão, comente a última linha do arquivo `/etc/pam.d/vsftpd`:

```
# cat /etc/pam.d/vsftpd | tail -n1
#auth   required        pam_shells.so
```

4. Isso resolvido, crie o usuário `ftpuser` sem shell válido, e defina sua senha:

```
# useradd -m -s /bin/false ftpuser
# passwd ftpuser
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha atualizada com sucesso
```

5. Na máquina *Client_Linux*, crie um arquivo de teste para ser enviado por *upload* para o servidor FTP. A seguir, logue no servidor e envie o arquivo:

```
$ hostname
cliente

$ echo "client_linux : $( date )" > test

$ ftp 192.168.0.10
Connected to 192.168.0.10.
220 Servidor FTP SEG12
Name (192.168.0.10:aluno): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.

ftp> put test
local: test remote: test
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
45 bytes sent in 0.00 secs (1.0729 MB/s)
```

6. De volta à máquina *Server_Linux*, verifique que o arquivo foi enviado com sucesso.

```
# hostname
servidor

# cat /home/ftpuser/test
client_linux : Sáb Ago 11 18:55:30 -03 2018
```

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`.

1. Como pode ser visto abaixo, o servidor `ssh` já se encontra instalado na máquina *Server_Linux*:

```
# dpkg -l | grep '^[i ]*openssh-server '
ii  openssh-server                 1:6.7p1-5+deb8u4                    amd64
secure shell (SSH) server, for secure access from remote machines
```

2. Basta, então, acessá-la remotamente e executar o comando solicitado.

```
$ ssh aluno@192.168.0.10
The authenticity of host '192.168.0.10 (192.168.0.10)' can't be established.
ECDSA key fingerprint is 6f:65:6b:5b:8c:21:b7:00:17:e0:a9:f8:67:a4:e4:ea.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.10' (ECDSA) to the list of known hosts.
aluno@192.168.0.10's password:
No mail.
Last login: Sat Aug 11 19:05:14 2018 from cliente.empresa.com.br

$ hostname
servidor
```

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.

1. Primeiro, vamos gerar a chave RSA. Deixe o `passphrase` vazio para que não seja necessário digitar senha toda vez que for utilizar a chave.

```
$ hostname
cliente

$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aluno/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aluno/.ssh/id_rsa.
Your public key has been saved in /home/aluno/.ssh/id_rsa.pub.
The key fingerprint is:
10:5e:12:7d:a1:90:ab:48:46:d2:4a:91:e6:41:70:e9 aluno@cliente
The key's randomart image is:
+---[RSA 4096]----+
|+++.  =+. ..     |
|.*+  ..=...      |
|+=.   o...       |
|..E   ..         |
| o . .  S        |
|  . .            |
|                 |
|                 |
|                 |
+-----------------+
```

2. A seguir, copie a chave para o diretório `.ssh` do usuário `aluno`, na máquina *Server_Linux*:

```
$ ssh-copy-id aluno@192.168.0.10
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
aluno@192.168.0.10's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'aluno@192.168.0.10'"
and check to make sure that only the key(s) you wanted were added.
```

3. Finalmente, basta logar utilizando a chave privada RSA. O sistema remoto não irá solicitar senha.

```
$ ssh aluno@192.168.0.10
You have new mail.
Last login: Sat Aug 11 19:06:02 2018 from cliente.empresa.com.br

$ hostname
servidor

$ whoami
aluno
```

4. A chave pública RSA foi armazenada no arquivo `/home/aluno/.ssh/id_rsa.pub`, na pasta *home* do usuário `aluno` dentro da máquina *Server_Linux*. Vamos exibir seu conteúdo:

```
$ hostname
servidor

$ ls ~/.ssh
authorized_keys  id_rsa  id_rsa.pub  known_hosts

$ cat ~/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCvtAjoHfRfhxUDd67eZhncv8n034RXM0ZZUyNiDYvId27q8MKerFH
ZAnCMxf0Sm+2MMqfNZxcvH7EiF28VE3ikaMnqfi6xj8Nhqp+kzEXAQNLfuVGBjnmrz7EUOVtG2YvUMrkTqU
ibAOFPCKrlkhyJg06tmkJVhJuKB7jzOmOFTrWeInCkPukv4lmi4JaEuLA5He9Qepg9WYduH0Gydb6D5nDkc
HVt0zl5YT21imXOQFIMIHpfquKs6pc7kUFl/JiHHwAfJ+wkawyamTyKDSKbvwclzZvxeFpYBZ5VcwLy52bz
dmsFakU8cIU1nr+6sdvOuejy8kodfKIrE2zmQ4ZL aluno@cliente
```

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.

1. Primeiro, na máquina *Client_Linux*, vamos criar um arquivo contendo o *hostname* local e data corrente.

```
$ hostname
cliente

$ echo "$( hostname ) , $( date )" > scpfile.txt

$ cat scpfile.txt
cliente , Dom Ago 12 00:22:55 -03 2018
```

2. Agora, vamos copiar o arquivo usando `scp`:

```
$ scp /home/aluno/scpfile.txt aluno@192.168.0.10:~
scpfile.txt                                                   100%   39     0.0KB/s
00:00
```

3. Basta logar via `ssh` na máquina remota e exibir o conteúdo do arquivo para verificar o funcionamento do processo:

```
$ ssh aluno@192.168.0.10 'hostname ; cat ~/scpfile.txt'
servidor
cliente , Dom Ago 12 00:22:55 -03 2018
```

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.

1. Primeiro, na máquina *Client_Linux*, vamos criar um arquivo contendo o *hostname* local e data corrente.

```
$ hostname
cliente

$ echo "$( hostname ) , $( date )" > sftpfile.txt

$ cat sftpfile.txt
cliente , Dom Ago 12 00:26:25 -03 2018
```

2. Agora, vamos copiar o arquivo usando `scp`:

```
$ sftp aluno@192.168.0.10
Connected to 192.168.0.10.
sftp> pwd
Remote working directory: /home/aluno
sftp> put sftpfile.txt
Uploading sftpfile.txt to /home/aluno/sftpfile.txt
sftpfile.txt                                                  100%   39     0.0KB/s
00:00
```

3. Basta logar via `ssh` na máquina remota e exibir o conteúdo do arquivo para verificar o funcionamento do processo:

```
$ ssh aluno@192.168.0.10 'hostname ; cat ~/sftpfile.txt'
servidor
cliente , Dom Ago 12 00:26:25 -03 2018
```

Sessão 10 — 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*).

1. Instale o servidor web Apache:

```
# apt-get install apache2
```

2. Vamos testar o funcionamento acessando o IP do servidor *Server_Linux* através de um navegador instalado na máquina *Win7-padrao*:

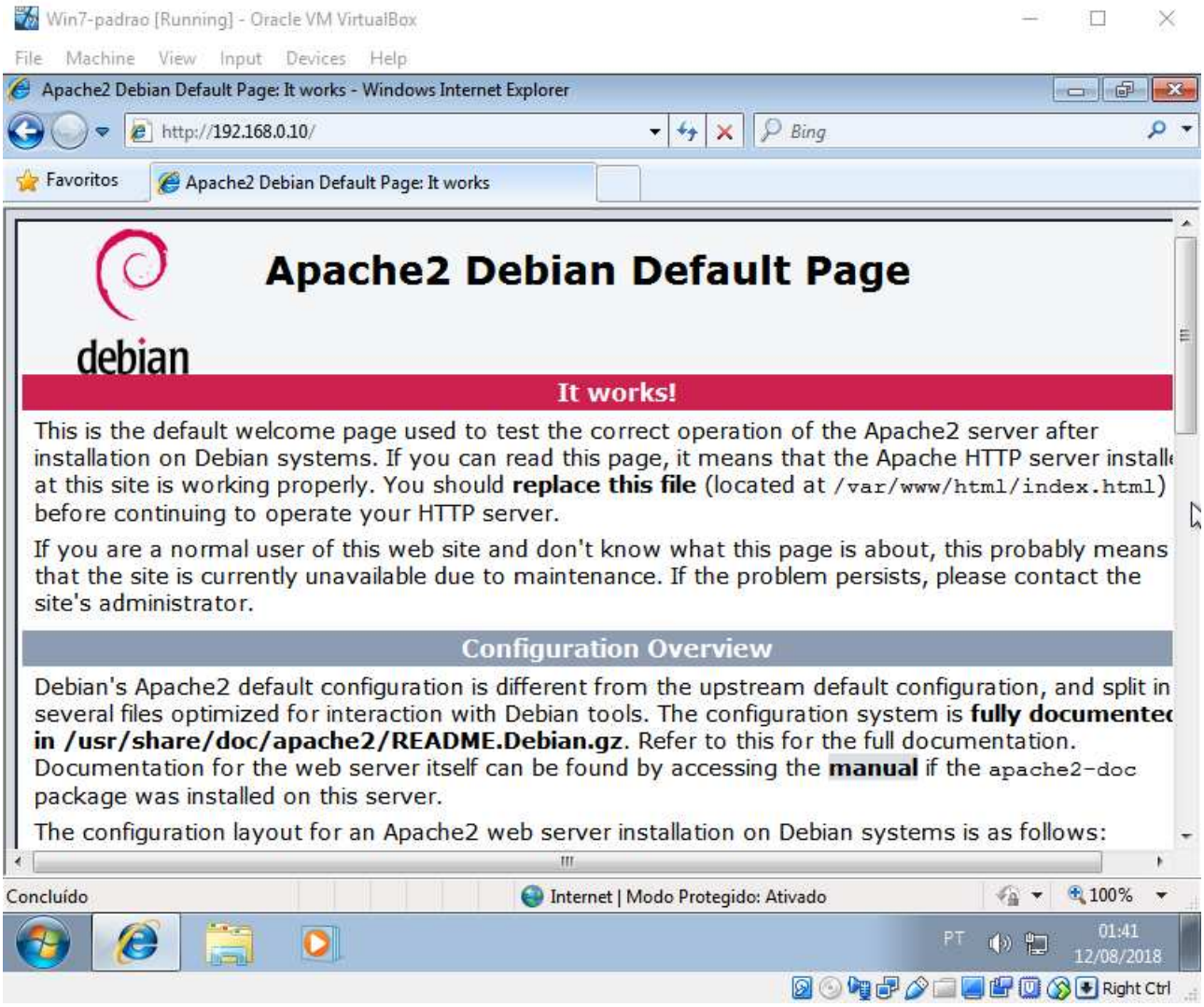

Figura 6: Apache instalado com sucesso

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.

Siga os passos abaixo:

1. Crie o arquivo de *virtualhost* `/etc/apache2/sites-available/www.conf` para o domínio `www.empresa.com.br`, como se segue:

   ```
   <VirtualHost *:80>
     ServerAdmin webmaster@empresa.com.br
     ServerName www.empresa.com.br
     DocumentRoot /var/www/www

     ErrorLog ${APACHE_LOG_DIR}/www-error.log
     CustomLog ${APACHE_LOG_DIR}/www-access.log combined
   </VirtualHost>
   ```

2. Faça o mesmo para o domínio `meusite.empresa.com.br`, editando o arquivo `/etc/apache2/sites-available/meusite.conf`:

```
<VirtualHost *:80>
  ServerAdmin webmaster@empresa.com.br
  ServerName meusite.empresa.com.br
  DocumentRoot /var/www/meusite

  ErrorLog ${APACHE_LOG_DIR}/meusite-error.log
  CustomLog ${APACHE_LOG_DIR}/meusite-access.log combined
</VirtualHost>
```

3. Crie a pasta e arquivo `index.html` para o *virtualhost* `www.empresa.com.br`:

```
# mkdir /var/www/www

# cat /var/www/www/index.html
<html>
  <head>
    <title>Test</title>
</head>
  <body>
    Welcome to www.empresa.com.br
  </body>
</html>
```

4. Faça o mesmo para o domínio `meusite.empresa.com.br`:

```
# mkdir /var/www/meusite

# cat /var/www/meusite/index.html
<html>
  <head>
    <title>Test</title>
</head>
  <body>
    Welcome to meusite.empresa.com.br
  </body>
</html>
```

5. Habilite ambos os *virtualhosts* e recarregue a configuração do Apache:

```
# a2ensite www
Enabling site www.
To activate the new configuration, you need to run:
  service apache2 reload

# a2ensite meusite.conf
Enabling site meusite.
To activate the new configuration, you need to run:
  service apache2 reload

# systemctl reload apache2
```

6. Vamos testar o funcionamento do *virtualhost* `www.empresa.com.br` através de um navegador instalado na máquina *Win7-padrao*:

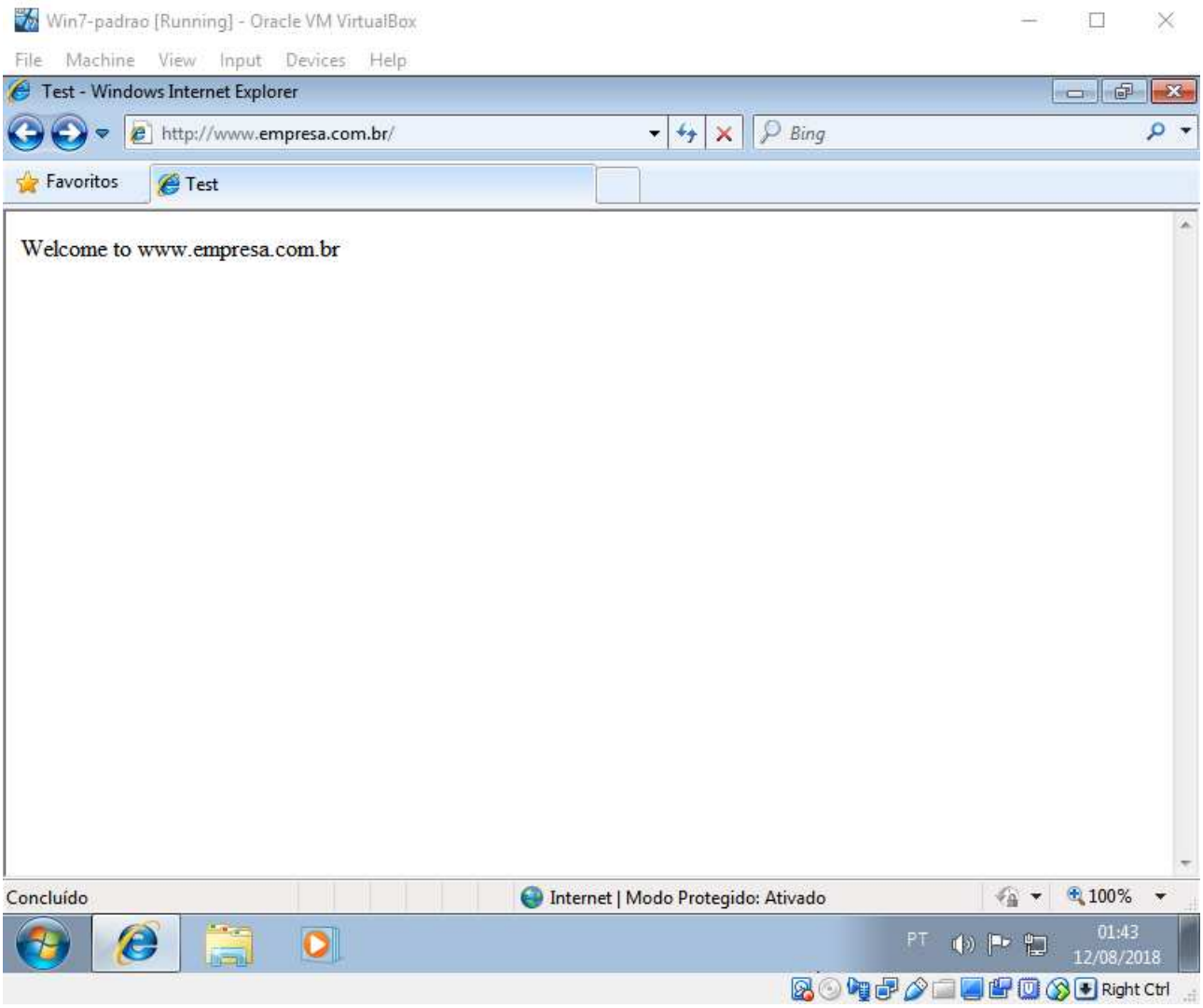

Figura 7: Virtualhost www.empresa.com.br acessível

7. E, novamente, repetiremos o teste para o *virtualhost* `meusite.empresa.com.br`:

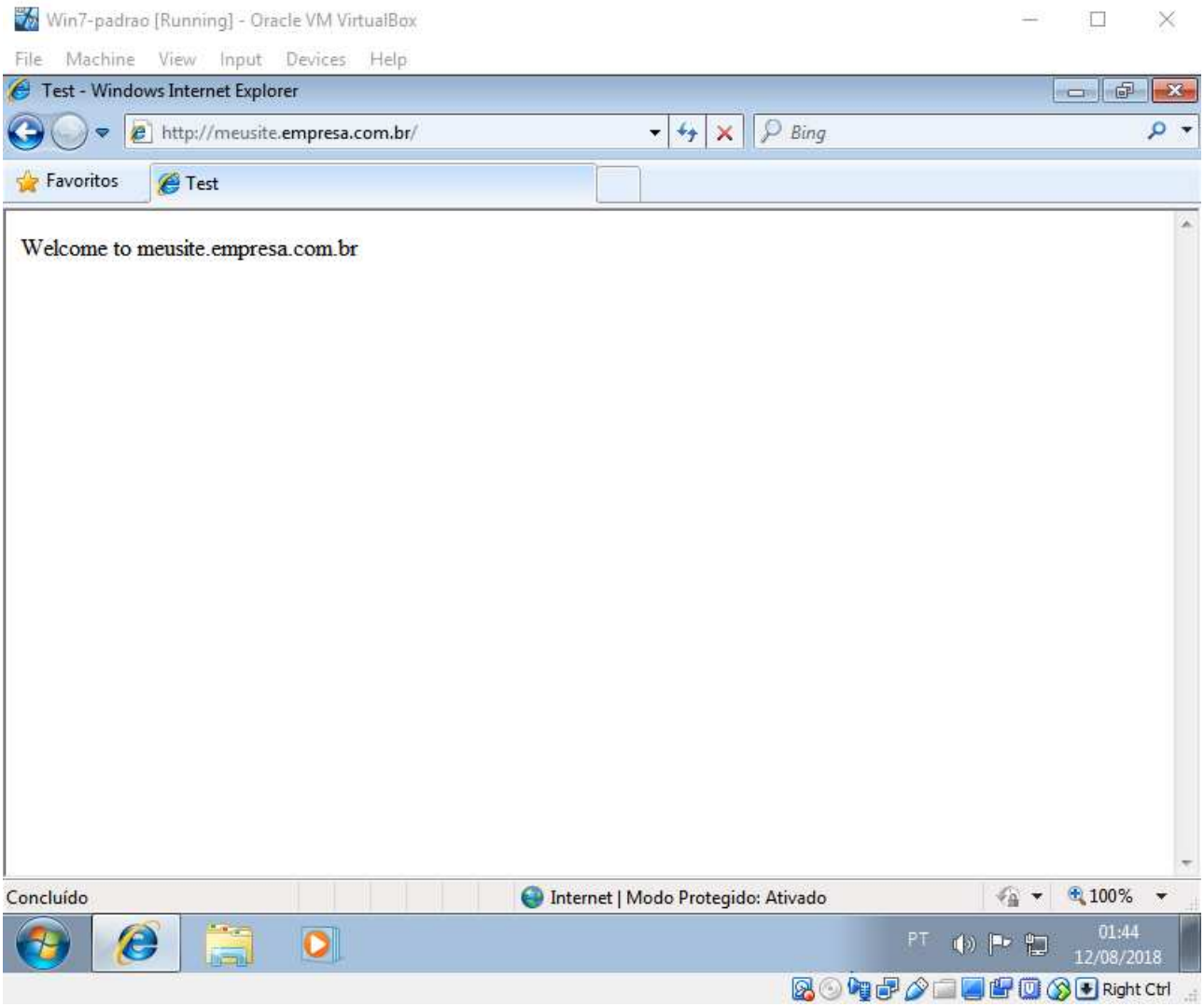

Figura 8: Virtualhost meusite.empresa.com.br acessível

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.

Siga os passos abaixo:

1. Habilite o módulo SSL no Apache:

```
# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create
self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
```

2. Gere o certificado auto-assinado usando o comando `openssl`. Para gerar um par de chaves com os parâmetros solicitados, basta usar as opções `-days 365` e `-newkey rsa:4096`. Observe, ainda, que a permissão da chave privada gerada pelo comando é muito leniente — utilize o comando `chmod 600` para corrigir isso.

```
# openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout
/etc/ssl/private/meusite.key -out /etc/ssl/certs/meusite.crt
Generating a 4096 bit RSA private key
.....................................................++
.....................................................................
.....................++
writing new private key to '/etc/ssl/private/meusite.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:DF
Locality Name (eg, city) []:Brasilia
Organization Name (eg, company) [Internet Widgits Pty Ltd]:RNP
Organizational Unit Name (eg, section) []:ESR
Common Name (e.g. server FQDN or YOUR name) []:meusite.empresa.com.br
Email Address []:webmaster@empresa.com.br

# chmod 600 /etc/ssl/private/meusite.key

# ls -ld /etc/ssl/private/meusite.key
-rw------- 1 root root 3272 Ago 12 02:09 /etc/ssl/private/meusite.key
```

3. Edite o arquivo `/etc/apache2/sites-available/meusite.conf`, habilitando o redirecionamento de requisições da porta 80/HTTP para a porta 443/HTTPS, ativando a *engine* SSL e informando o caminho para as chaves pública e privada do *virtualhost*:

```
<VirtualHost *:80>
  ServerName meusite.empresa.com.br
  Redirect permanent / https://meusite.empresa.com.br/
</VirtualHost>

<VirtualHost *:443>
  ServerAdmin webmaster@empresa.com.br
  ServerName meusite.empresa.com.br
  DocumentRoot /var/www/meusite

  SSLEngine On
  SSLCertificateFile    /etc/ssl/certs/meusite.crt
  SSLCertificateKeyFile /etc/ssl/private/meusite.key

  ErrorLog ${APACHE_LOG_DIR}/meusite-error.log
  CustomLog ${APACHE_LOG_DIR}/meusite-access.log combined
</VirtualHost>
```

4. Reinicie o Apache para que ele ative o módulo SSL e releia o arquivo de configuração do *virtualhost*:

```
# systemctl restart apache2
```

5. Agora, basta testar. Acessamos a URL http://meusite.empresa.com.br e, de fato, o redirecionamento para HTTPS funcionou. Somos apresentados a uma tela de certificado inválido:

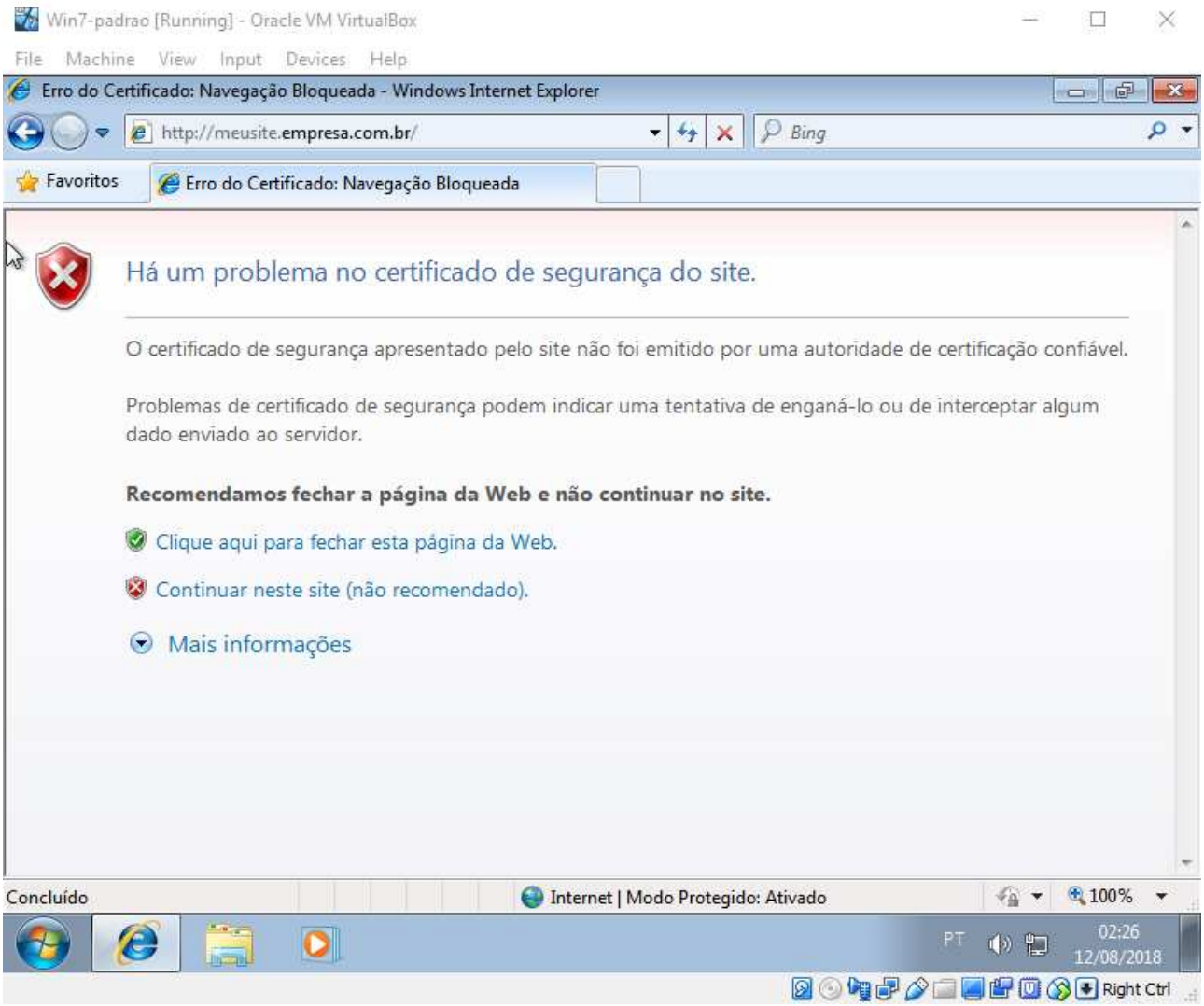

Figura 9: Alerta de certificado inválido

6. Esse erro é esperado, já que o certificado SSL que estamos utilizando é auto-assinado, e não pode ser verificado pelas autoridades certificadoras raiz instaladas no navegador. Após clicar em "Continuar neste *site*", conseguimos acessar a página objetivada:

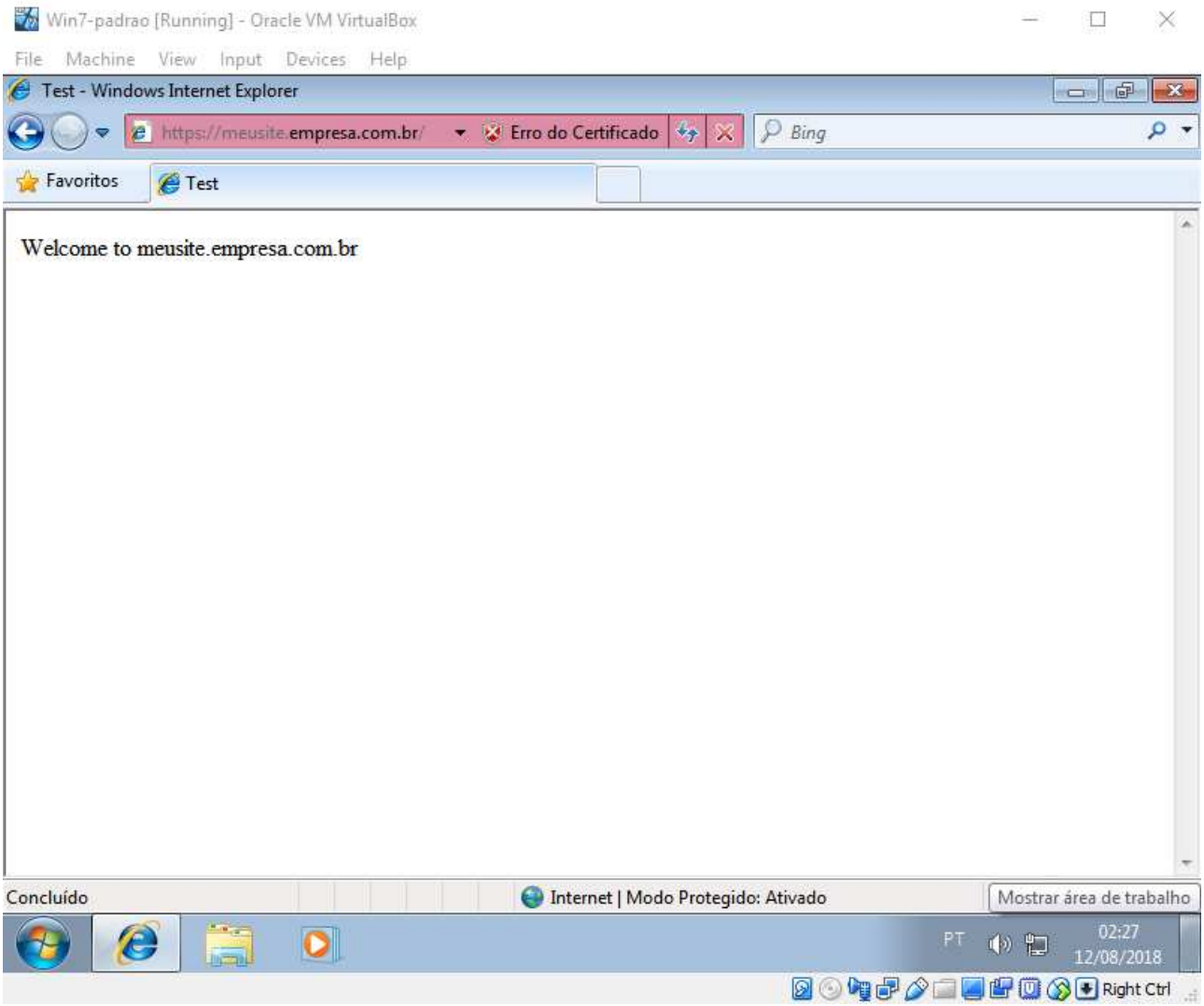

Figura 10: Acesso via HTTPS a meusite.empresa.com.br

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.

Siga os passos abaixo:

1. Habilite o módulo de autenticação LDAP no Apache:

```
# a2enmod authnz_ldap
Considering dependency ldap for authnz_ldap:
Enabling module ldap.
Enabling module authnz_ldap.
To activate the new configuration, you need to run:
  service apache2 restart
```

2. Crie o diretório `/var/www/meusite/restrito`, e dentro dele edite um arquivo `index.html` que indique com clareza que foi possível obter acesso à área restrita:

```
# mkdir /var/www/meusite/restrito

# cat /var/www/meusite/restrito/index.html
<html>
  <head>
    <title>Test auth</title>
</head>
  <body>
    Restricted area for meusite.empresa.com.br
  </body>
</html>
```

3. Edite o arquivo `/etc/apache2/sites-available/meusite.conf`, habilitando autenticação via LDAP caso o cliente solicite acesso ao diretório `/var/www/meusite/restrito`. Tenha especial atenção ao configurar o filtro da URL LDAP:

```
<VirtualHost *:80>
  ServerName meusite.empresa.com.br
  Redirect permanent / https://meusite.empresa.com.br/
</VirtualHost>

<VirtualHost *:443>
  ServerAdmin webmaster@empresa.com.br
  ServerName meusite.empresa.com.br
  DocumentRoot /var/www/meusite

  SSLEngine On
  SSLCertificateFile    /etc/ssl/certs/meusite.crt
  SSLCertificateKeyFile /etc/ssl/private/meusite.key

  <Directory /var/www/meusite/restrito>
    AuthType basic
    AuthBasicProvider ldap
    AuthName "meusite LDAP login"
    AuthLDAPURL ldap://127.0.0.1/ou=People,dc=empresa,dc=com,dc=br?uid?sub?
(objectClass=posixAccount)
    AuthLDAPBindDN cn=admin,dc=empresa,dc=com,dc=br
    AuthLDAPBindPassword rnpesr
    Require valid-user
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/meusite-error.log
  CustomLog ${APACHE_LOG_DIR}/meusite-access.log combined
</VirtualHost>
```

4. Reinicie o Apache para que ele ative o módulo de autenticação LDAP e releia o arquivo de configuração do *virtualhost*:

```
# systemctl restart apache2
```

5. Agora, basta testar. Acessamos a URL https://meusite.empresa.com.br/restrito e imediatamente fomos apresentados a uma tela de autenticação solicitando usuário e senha:

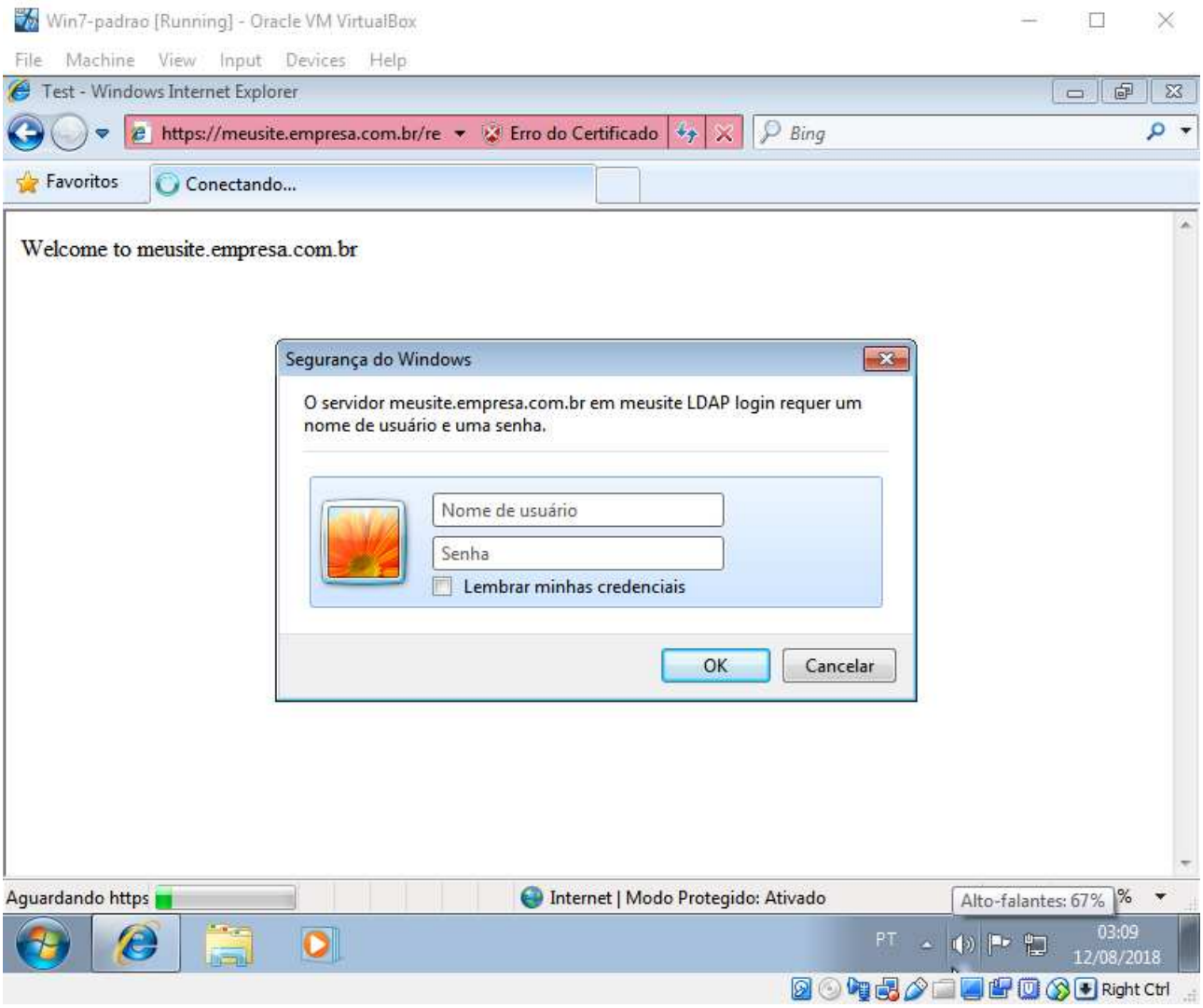

Figura 11: Autenticação LDAP no Apache

6. Ao informar uma combinação válida (por exemplo, usuário `aluno` e senha `rnpesr`), o Apache autoriza o acesso à área restrita do *virtualhost*:

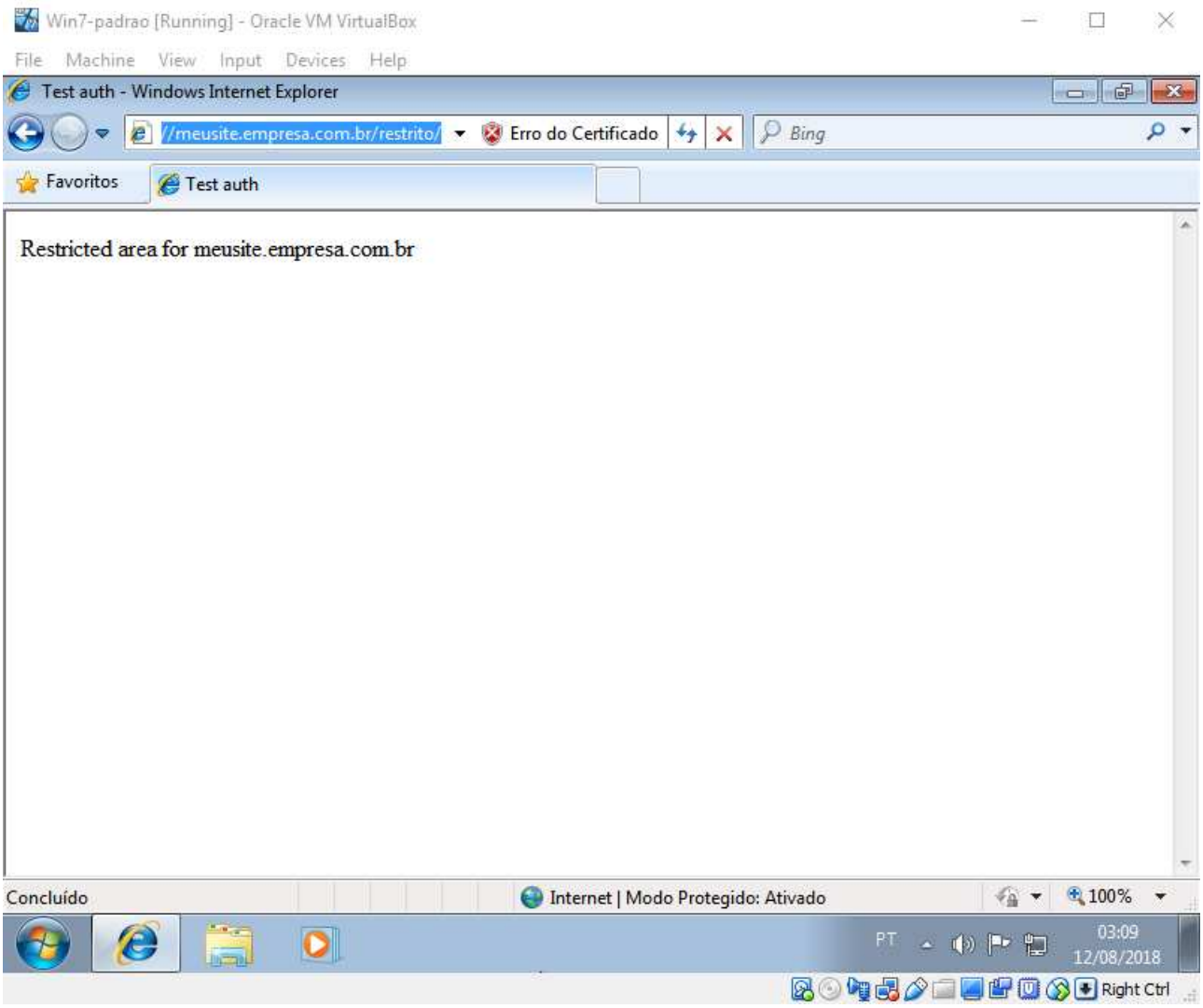

Figura 12: Usuário autenticado no LDAP/Apache com sucesso

7. Refazendo o acesso, mas desta vez informando uma combinação de usuário/senha inexistente, o servidor web informa que não estamos autorizados a acessar a área restrita:

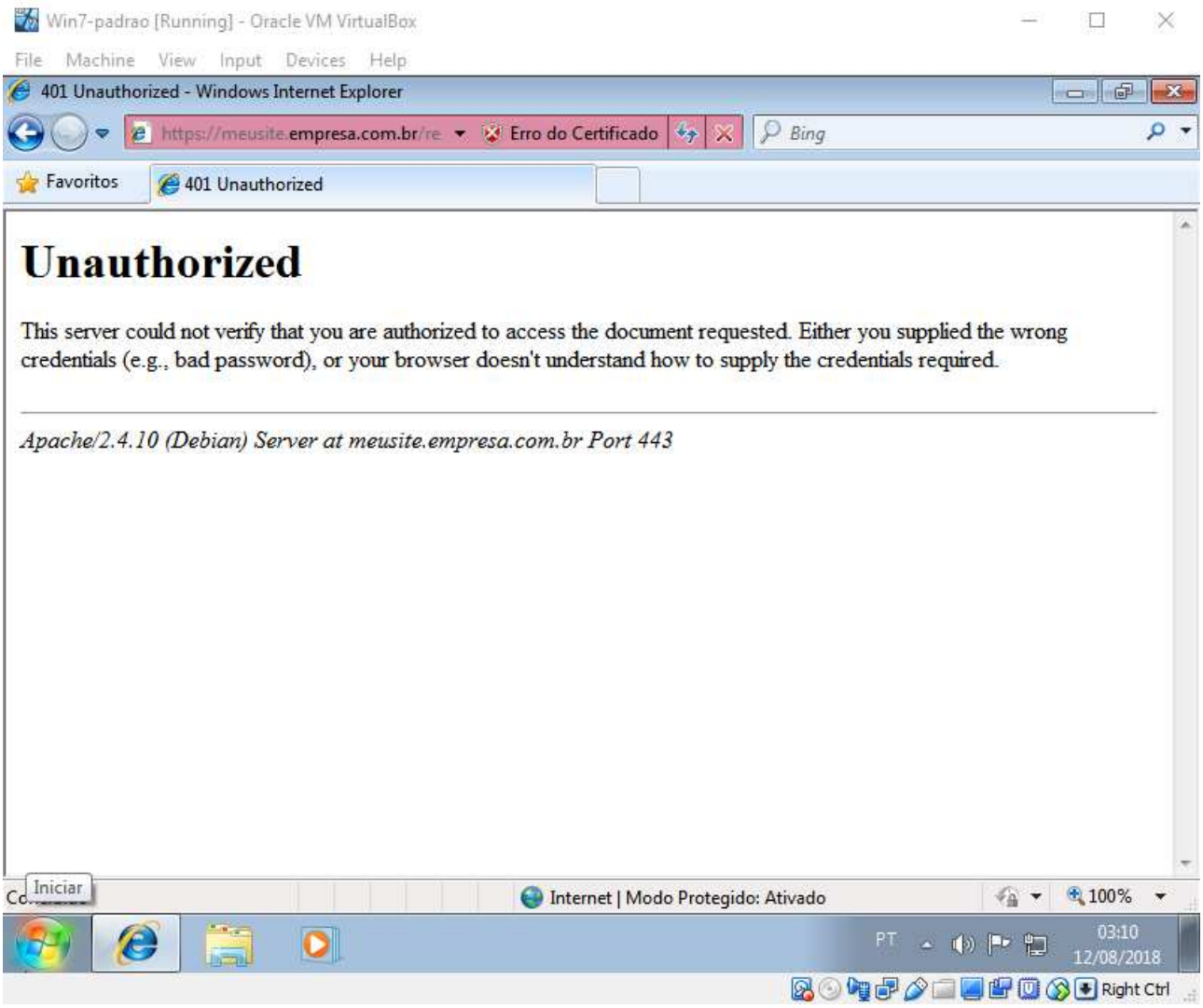

Figura 13: Usuário não autorizado pelo LDAP/Apache

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.

Siga os passos abaixo:

1. Habilite o módulo de publicação de páginas pessoais no Apache:

```
# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  service apache2 restart
```

2. Crie a pasta `/home/aluno/public_html`, e crie nela um arquivo `index.html` com conteúdo sugestivo:

```
$ mkdir ~/public_html

$ cat public_html/index.html
<html>
  <head>
    <title>Test userdir</title>
</head>
  <body>
    Welcome to the ALUNO homepage
  </body>
</html>
```

3. Para que usuários criados no futuro possuam a pasta `public_html` criada automaticamente eu seus diretórios *home*, basta criar uma pasta de mesmo nome no diretório `/etc/skel`:

```
# mkdir /etc/skel/public_html
```

4. Reinicie o Apache para que ele ative o módulo de publicação de páginas pessoais:

```
# systemctl restart apache2
```

5. Agora, basta testar. Acessamos a URL http://192.168.0.10/~aluno e logo podemos ver a página pessoal do usuário `aluno`, como esperado:

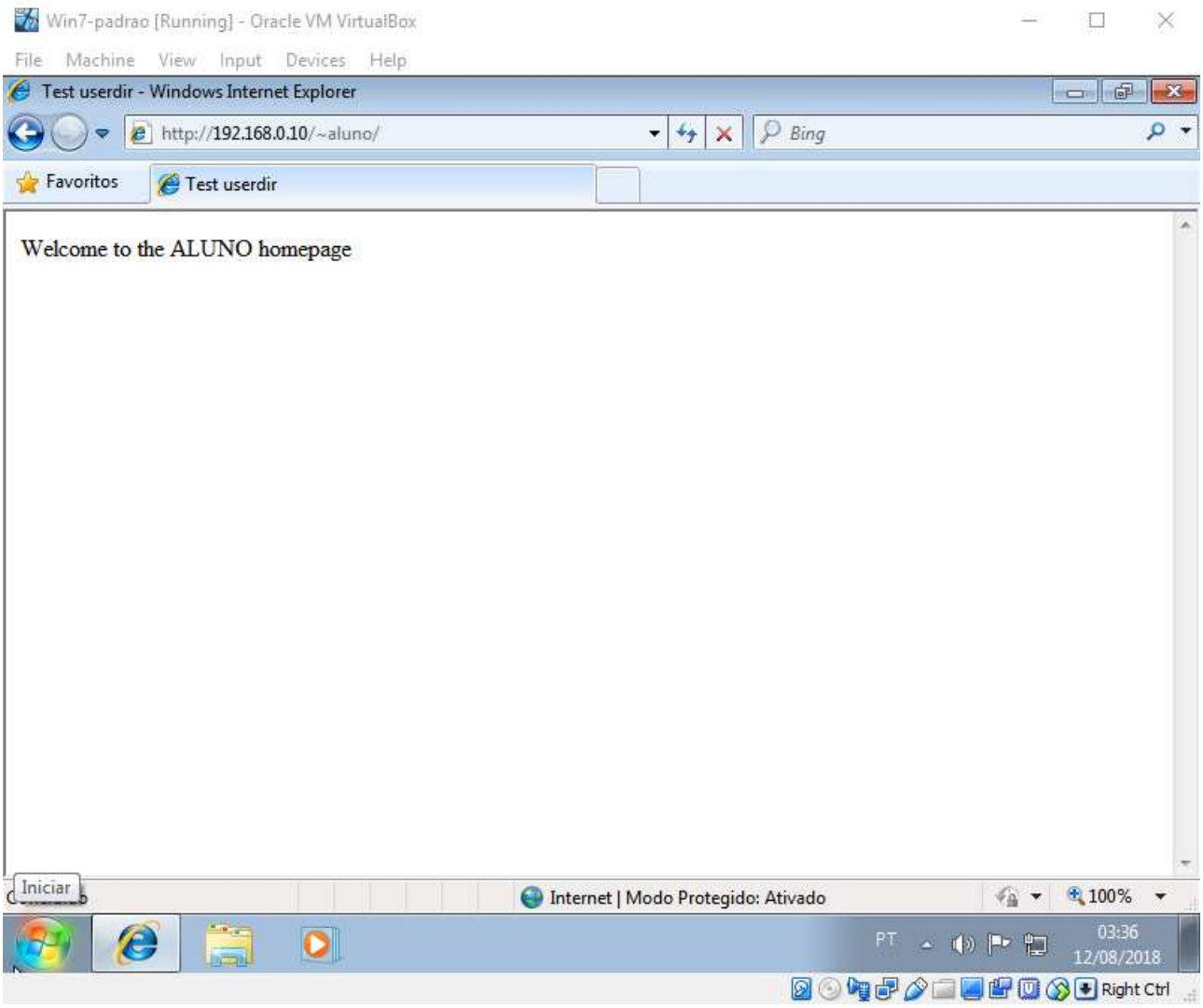

Figura 14: Acesso à página pessoal do usuário aluno

Sessão 11 — 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.

1. Instale o servidor SMTP Postfix e o Cyrus SASL:

   ```
   # apt-get install postfix sasl2-bin mailutils
   ```

2. Reconfigure o Postfix de acordo com os dados apontados na tabela acima:

   ```
   # dpkg-reconfigure postfix
   ```

3. Gere as chaves assimétricas usando o comando `openssl`. Para gerar um par de chaves com os parâmetros solicitados, basta usar as opções `-days 730` e `-newkey rsa:4096`.

```
# openssl req -x509 -nodes -days 730 -newkey rsa:4096 -keyout
/etc/ssl/private/smtpd.key -out /etc/ssl/certs/smtpd.crt
Generating a 4096 bit RSA private key
................++
..............................++
writing new private key to '/etc/ssl/private/smtpd.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:DF
Locality Name (eg, city) []:Brasilia
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Empresa
Organizational Unit Name (eg, section) []:TI
Common Name (e.g. server FQDN or YOUR name) []:email.empresa.com.br
Email Address []:postmaster@empresa.com.br
```

4. Observe que a permissão da chave privada gerada pelo comando acima é muito leniente — utilize o comando `chmod 600` para corrigir isso.

```
# ls -ld /etc/ssl/private/smtpd.key
-rw-r--r-- 1 root root 3272 Ago 12 15:31 /etc/ssl/private/smtpd.key

# chmod 600 /etc/ssl/private/smtpd.key
```

5. Antes de editar o arquivo de configuração do Postfix, faça o *backup* da versão original em caso de necessidade de *rollback*:

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.orig
```

6. Edite o arquivo principal de configuração do Postfix, `/etc/postfix/main.cf`, da seguinte forma:

```
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/smtpd.crt
smtpd_tls_key_file=/etc/ssl/private/smtpd.key

smtpd_use_tls=yes
smtpd_tls_auth_only = no
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache

smtp_use_tls = yes
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# SASL parameters
smtpd_sasl_path = smtpd
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = empresa.com.br

biff = no
append_dot_mydomain = no
readme_directory = no

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination
myhostname = email.empresa.com.br
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = email.empresa.com.br, localhost.empresa.com.br, empresa.com.br,
localhost
relayhost =
mynetworks = 127.0.0.0/8, 192.168.0.0/24, 172.16.0.0/24, [::ffff:127.0.0.0]/104,
[::1]/128
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

7. A configuração de autenticação SASL fica no arquivo `/etc/postfix/sasl/smtpd.conf`, como se segue:

```
pwcheck_method: saslauthd
mech_list: PLAIN LOGIN
```

8. Precisamos ativar o *daemon* `saslauthd`, bem como configurá-lo para operar com o Postfix sob `chroot` no diretório `/var/spool/postfix`:

```
# cat /etc/default/saslauthd | grep '^START=\|^OPTIONS='
START=yes
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd"
```

9. Para que o Postfix consiga se comunicar com o `saslauthd` e autenticar usuários, é necessário adicioná-lo ao grupo deste:

```
# adduser postfix sasl
```

10. Reinicie ambos os *daemons* — em caso de erros, verifique os arquivos `/var/log/syslog` e `/var/log/daemon.log`:

```
# systemctl restart postfix.service
# systemctl restart saslauthd.service
```

11. Agora, basta testar o funcionamento da conexão. O único impeditivo final é que, no método PLAIN, o servidor SMTP espera o envio da combinação usuário/senha em um formato específico — `\0username\0password` — e codificado em base64. Podemos fazer isso usando o comando `openssl`, como se segue:

```
# echo -ne '\000aluno\000rnpesr' | openssl base64
AGFsdW5vAHJucGVzcg==
```

12. Finalmente, basta conectar-se ao servidor SMTP via `telnet` e fornecer as informações de autenticação obtidas acima:

```
# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 email.empresa.com.br ESMTP Postfix

EHLO localhost
250-email.empresa.com.br
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

AUTH PLAIN AGFsdW5vAHJucGVzcg==
235 2.7.0 Authentication successful
```

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.

1. Já que utilizamos o usuário `aluno` no teste da atividade anterior, vamos tentar logar com um usuário diferente. O usuário `esr`, que foi criado anteriormente na base LDAP, será nosso remetente:

```
# getent passwd | grep '^esr:'
esr:x:5000:5000:esr,,,:/home/esr:/bin/bash
```

2. Temos que gerar a *string* de autenticação base64, como feito anteriormente:

```
# echo -ne '\000esr\000rnpesr' | openssl base64
AGVzcgBybnBlc3I=
```

3. Agora, basta logar no servidor SMTP e enviar a mensagem. Usa-se, em ordem, os comandos `EHLO`, `AUTH PLAIN`, `MAIL FROM`, `RCPT TO`, `DATA` e `QUIT`, como mostrado abaixo:

```
# telnet localhost 25
Trying ::1...
Connected to localhost.
Escape character is '^]'.
220 email.empresa.com.br ESMTP Postfix

EHLO localhost
250-email.empresa.com.br
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

AUTH PLAIN AGVzcgBybnBlc3I=
235 2.7.0 Authentication successful

MAIL FROM:esr@empresa.com.br
250 2.1.0 Ok

RCPT TO:aluno@empresa.com.br
250 2.1.5 Ok

DATA
354 End data with <CR><LF>.<CR><LF>
Mensagem de teste
.
250 2.0.0 Ok: queued as C0465A075C

QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

4. Vamos finalmente entrar como o usuário `aluno` e verificar a caixa de entrada:

```
# su - aluno

$ mail
"/var/mail/aluno": 1 message 1 unread
>U   1 esr@empresa.com.br Dom Ago 12 17:29  14/475
? 1
Return-Path: <esr@empresa.com.br>
X-Original-To: aluno@empresa.com.br
Delivered-To: aluno@empresa.com.br
Received: from localhost (localhost [IPv6:::1])
        by email.empresa.com.br (Postfix) with ESMTPA id C0465A075C
        for <aluno@empresa.com.br>; Sun, 12 Aug 2018 17:29:34 -0300 (-03)
Message-Id: <20180812202940.C0465A075C@email.empresa.com.br>
Date: Sun, 12 Aug 2018 17:29:34 -0300 (-03)
From: esr@empresa.com.br
X-IMAPbase: 1534106068 10
Status: O
X-UID: 9

Mensagem de teste
```

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?

Vamos abrir um terminal monitorando por mudanças no arquivo de log do servidor SMTP com o comando `tail -f -n0 /var/log/mail.log`. Em outro terminal, vamos executar uma nova sessão de envio de email usando o comando `telnet`, como feito anteriormente.

1. Assim que a conexão é aberta, visualiza-se a mensagem:

   ```
   Aug 12 17:57:02 email postfix/smtpd[6039]: connect from localhost[::1]
   ```

   Logo, o IP de origem da conexão é `localhost`, ou 127.0.0.1.

2. Assim que o comando `RCPT TO` é enviado, surge uma nova mensagem:

```
Aug 12 18:02:15 email postfix/smtpd[6079]: F0D56A0282: client=localhost[::1],
sasl_method=PLAIN, sasl_username=esr@empresa.com.br
```

Assim, o usuário que efetuou login e deseja enviar a mensagem é o `esr@empresa.com.br`.

3. Quando o caractere ".", que delimita o final da mensagem, é enviado, vemos novas mensagens no log:

```
Aug 12 18:03:31 email postfix/cleanup[6082]: F0D56A0282: message-
id=<20180812210215.F0D56A0282@email.empresa.com.br>
Aug 12 18:03:31 email postfix/qmgr[5983]: F0D56A0282: from=<esr@empresa.com.br>,
size=333, nrcpt=1 (queue active)
Aug 12 18:03:31 email postfix/local[6091]: F0D56A0282: to=<aluno@empresa.com.br>,
relay=local, delay=81, delays=81/0/0/0, dsn=2.0.0, status=sent (delivered to
command: procmail -a "$EXTENSION")
Aug 12 18:03:31 email postfix/qmgr[5983]: F0D56A0282: removed
```

No campo `to=` da terceira linha do log acima pode-se observar que o endereço do destinatário é `aluno@empresa.com.br`. Na mesma linha, vê-se que o método de entrega é via *relay* local, usando o programa auxiliar `procmail`.

Sessão 12 — 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.

1. Para os usuários preexistentes, é necessário criar um diretório de nome `Maildir` com as pastas mencionadas pela atividade. Por exemplo, para o usuário `aluno`:

```
# mkdir -p /home/aluno/Maildir/{cur,new,tmp,.Drafts,.Spam,.Trash}
# chown -R aluno.aluno /home/aluno/Maildir
# chmod 700 /home/aluno/Maildir
```

2. Para fazer com que todos os usuários futuros já possuam a pasta `Maildir` e seus subdiretórios criados de forma automática no *home*, basta:

```
# mkdir -p /etc/skel/Maildir/{cur,new,tmp,.Drafts,.Spam,.Trash}
# chmod 700 /etc/skel/Maildir
```

3. Altere os parâmetros `home_mailbox` e `mailbox_command` do Postfix no arquivo `/etc/postfix/main.cf`, como se segue:

```
# grep '^home_mailbox\|^mailbox_command' /etc/postfix/main.cf
home_mailbox = Maildir/
mailbox_command =
```

4. Reinicie o Postfix:

```
# systemctl restart postfix.service
```

5. Envie um email para um usuário qualquer, digamos, o usuário `aluno`, usando o comando `mail`. Verifique nos logs do servidor SMTP o estilo da entrega:

```
# echo 'Teste Maildir' | mail -s 'Teste' aluno@empresa.com.br

# tail -n4 /var/log/mail.log
Aug 12 21:54:00 email postfix/pickup[1587]: 78FA3A02AF: uid=0
from=<root@email.empresa.com.br>
Aug 12 21:54:00 email postfix/cleanup[1593]: 78FA3A02AF: message-
id=<20180813005400.78FA3A02AF@email.empresa.com.br>
Aug 12 21:54:00 email postfix/qmgr[1588]: 78FA3A02AF:
from=<root@email.empresa.com.br>, size=363, nrcpt=1 (queue active)
Aug 12 21:54:00 email postfix/local[1595]: 78FA3A02AF: to=<aluno@empresa.com.br>,
relay=local, delay=0.01, delays=0.01/0/0/0, dsn=2.0.0, status=sent (delivered to
maildir)
Aug 12 21:54:00 email postfix/qmgr[1588]: 78FA3A02AF: removed
```

6. Verifique se a mensagem foi enviada corretamente para o diretório `Maildir` do usuário:

```
# ls -R /home/aluno/Maildir/
/home/aluno/Maildir/:
cur  new  tmp

/home/aluno/Maildir/cur:

/home/aluno/Maildir/new:
1534121640.V801I605d1M501046.email

/home/aluno/Maildir/tmp:
```

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`.

1. Primeiro, vamos instalar os pacotes solicitados:

```
# apt-get install --no-install-recommends courier-imap-ssl courier-pop-ssl
libsasl2-modules-ldap gamin
```

2. Tente a conexão com o servidor POP. Em caso de sucesso, faça login como usuário `aluno` usando os comandos `USER` e `PASS`.

```
# telnet localhost 110
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Aug 12 22:10:20 email pop3d: Connection, ip=[::1]
+OK Hello there.

USER aluno
+OK Password required.

PASS rnpesr
+OK logged in.
```

3. Excelente! Vamos listar as mensagens usando o comando `LIST`. A seguir, exibir uma delas usando o comando `RETR`:

```
Aug 12 22:10:25 email pop3d: LOGIN, user=aluno, ip=[::1], port=[38447]

LIST
+OK POP3 clients that break here, they violate STD53.
1 478
2 478
.

RETR 1
+OK 478 octets follow.
Return-Path: <root@email.empresa.com.br>
X-Original-To: aluno@empresa.com.br
Delivered-To: aluno@empresa.com.br
Received: by email.empresa.com.br (Postfix, from userid 0)
        id 78FA3A02AF; Sun, 12 Aug 2018 21:54:00 -0300 (-03)
Subject: Teste
To: <aluno@empresa.com.br>
X-Mailer: mail (GNU Mailutils 2.99.98)
Message-Id: <20180813005400.78FA3A02AF@email.empresa.com.br>
Date: Sun, 12 Aug 2018 21:54:00 -0300 (-03)
From: root@email.empresa.com.br (root)

Teste Maildir
.

QUIT
Aug 12 22:10:40 email pop3d: LOGOUT, user=aluno, ip=[::1], port=[38447], top=0,
retr=465, rcvd=20, sent=607, time=15
+OK Bye-bye.
Connection closed by foreign host.
```

4. Finalmente, vamos testar a conexão com o servidor IMAP. Apenas o procedimento de login e listagem de pastas é suficiente:

```
# telnet localhost 143
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Aug 12 22:16:26 email imapd: Connection, ip=[::1]
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT
THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION] Courier-IMAP ready. Copyright
1998-2011 Double Precision, Inc.  See COPYING for distribution information.

01 LOGIN aluno rnpesr
01 OK LOGIN Ok.
Aug 12 22:16:33 email imapd: LOGIN, user=aluno, ip=[::1], port=[50201],
protocol=IMAP

. LIST "" "*"
* LIST (\Marked \HasNoChildren) "." "INBOX"
. OK LIST completed
```

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`.

1. Primeiro, edite o arquivo `/etc/default/saslauthd` e altere o mecanismo de autenticação de `pam` para `ldap`:

```
# cat /etc/default/saslauthd | grep '^MECHANISMS='
MECHANISMS="ldap"
```

2. A seguir, edite o arquivo `/etc/postfix/sasl/smtpd.conf` para o Cyrus SASL utilizar o *plugin* `ldapdb` em procedimentos de autenticação. Informe também a URL do diretório LDAP, usuário administrativo e senha:

```
pwcheck_method: saslauthd
auxprop_plugin: ldapdb
mech_list: PLAIN LOGIN DIGEST-MD5
ldapdb_uri: ldap://127.0.0.1
ldapdb_id: admin
ldapdb_pw: rnpesr
ldapdb_mech: DIGEST-MD5
```

3. Crie o arquivo novo `/etc/saslauthd.conf`. Nele, indique qual é o *search base* de pesquisa no diretório LDAP, bem como o filtro de pesquisa:

```
ldap_servers: ldap://127.0.0.1/
ldap_search_base: ou=People,dc=empresa,dc=com,dc=br
ldap_auth_method: bind
ldap_filter: uid=%U
```

4. Reinicie os serviços `postfix` e `saslauthd`:

```
# systemctl restart postfix.service
# systemctl restart saslauthd.service
```

5. Teste o funcionamento da configuração com o comando `testsaslauthd`. Como o Postfix está em `chroot` no diretório `/var/spool/postfix`, temos que passar o caminho completo até o *socket* com a opção `-f`. Vamos testar tanto com usuário local quanto com um usuário existente apenas no diretório LDAP:

```
# testsaslauthd -u aluno -p rnpesr -f /var/spool/postfix/var/run/saslauthd/mux
0: OK "Success."

# testsaslauthd -u esr -p rnpesr -f /var/spool/postfix/var/run/saslauthd/mux
0: OK "Success."
```

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

Parâmetro	Valor
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.

Sessão 13 — *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.

1. Primeiro, vamos instalar os pacotes:

```
# apt-get install squid3 sarg
```

2. Note que o arquivo de configuração do Squid é imenso, com 7655 linhas. Ele é tão grande porque inclui comentários extremamente detalhados para cada opção de configuração — excluindo-se linhas comentadas e em branco, restam apenas 24 linhas efetivas de configuração. Vamos fazer um backup do arquivo original e trabalhar apenas com o conteúdo relevante:

```
# wc -l /etc/squid3/squid.conf
7655 /etc/squid3/squid.conf

# grep -v '^#' /etc/squid3/squid.conf.orig | sed '/^$/d' | wc -l
24

# cp /etc/squid3/squid.conf /etc/squid3/squid.conf.orig
```

3. A seguir, vamos editar o arquivo de acordo com as especificações da atividade. Preste especial atenção aos blocos `http_access`, que são lidos sequencialmente de cima para baixo:

```
acl intnet1 src 192.168.0.0/24  # rede Client_Linux
acl intnet2 src 172.16.0.0/24   # rede Win7-padrao

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

# allow local networks
http_access allow intnet1
http_access allow intnet2

# default http_access block
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localhost
http_access deny all

# cache options
cache_effective_user proxy
cache_dir ufs /var/spool/squid3 100 16 256
cache_log /var/log/squid3/cache.log
cache_access_log /var/log/squid3/access.log
cache_store_log none

# additional configuration
http_port 3128
coredump_dir /var/spool/squid3
refresh_pattern ^ftp:   1440  20% 10080
refresh_pattern ^gopher:  1440  0%  1440
refresh_pattern -i (/cgi-bin/|\?) 0 0%  0
refresh_pattern .   0 20% 4320
shutdown_lifetime 1 second
```

4. Pare o serviço do Squid e invoque-o com a opção `-z`, que irá criar as pastas do diretório de *cache*. Após o final das mensagens de log, digite `CTRL + C` para cancelar o comando.

```
# systemctl stop squid3.service
# squid3 -z
(...)
^C
```

5. Finalmente, inicie o processo do Squid.

```
# systemctl start squid3.service
```

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.

1. Para configurar o *proxy* no Windows, acesse: Iniciar → Opções da Internet → Aba Conexões → Configurações da LAN. Desmarque a caixa "Detectar automaticamente as configurações", e marque as caixas "Usar um servidor *proxy* para a rede local" e "Não usar servidor *proxy* para endereços locais". Finalmente, insira o IP da máquina *Server_Linux* (172.16.0.10) e porta do Squid (3128) nos campos apropriados, como se segue:

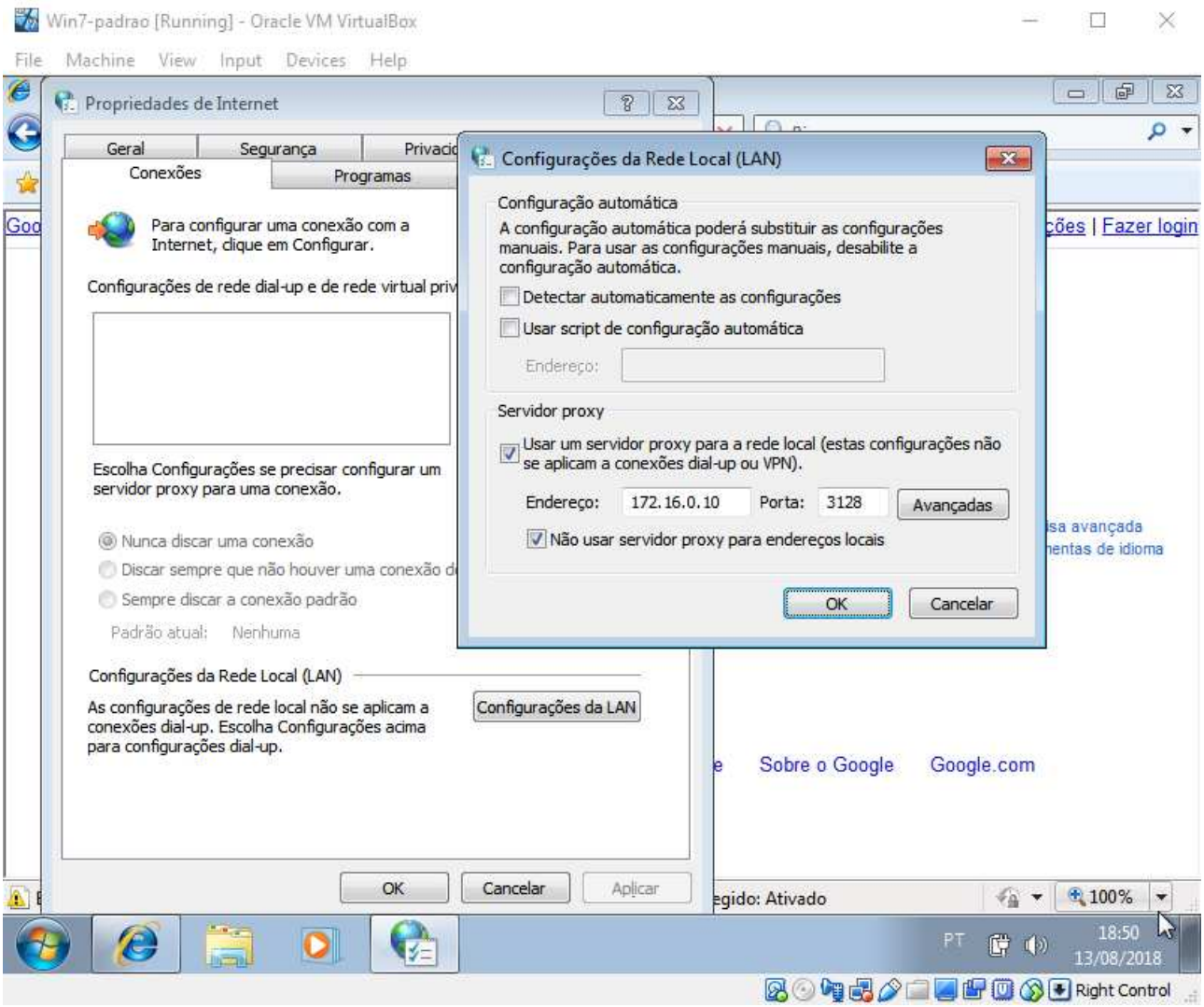

Figura 15: Configuração de proxy direto

2. Feito isso, basta acessar http://www.openbsd.org e verificar os eventos registrados no arquivo `/var/log/squid3/access.log`.

```
# tail -f -n0 /var/log/squid3/access.log
1534200817.622   2157 172.16.0.51 TCP_MISS/200 5495 GET http://www.openbsd.org/ -
HIER_DIRECT/129.128.5.194 text/html
1534200818.683   1260 172.16.0.51 TCP_MISS/200 20896 GET
http://www.openbsd.org/images/puffy63.gif - HIER_DIRECT/129.128.5.194 image/gif
1534200818.977   1323 172.16.0.51 TCP_MISS/200 50729 GET
http://www.openbsd.org/images/rack2009-s.png - HIER_DIRECT/129.128.5.194 image/png
1534200819.749    572 172.16.0.51 TCP_MISS/200 5003 GET
http://www.openbsd.org/favicon.ico - HIER_DIRECT/129.128.5.194 image/x-icon
```

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.

1. Primeiro, vamos implementar o controle por endereço físico. Edite o arquivo `/etc/squid3/acl/mac.conf` e inclua:

```
08:00:27:00:ca:5f
```

No topo do bloco `acl` do arquivo de configuração `/etc/squid3/squid.conf`, inclua a linha:

```
acl block_mac  arp           "/etc/squid3/acl/mac.conf"
```

No topo do bloco `http_access`, faça o bloqueio.

```
http_access deny block_mac
```

Recarregue a configuração do Squid:

```
# systemctl reload squid3.service
```

Na máquina *Win7-padrao*, teste a configuração acessando algum link no website http://www.openbsd.org . Como a configuração deste bloqueio reagiria em casos de *arp spoofing*?

2. Agora vamos implementar o bloqueio por IP. Edite `/etc/squid3/acl/ip.conf`:

```
172.16.0.51
```

Da mesma forma que antes, inclua no topo do bloco `acl`:

```
acl block_ip   src           "/etc/squid3/acl/ip.conf"
```

E agora, no topo do bloco `http_access`:

```
http_access deny block_ip
```

Recarregue o Squid e teste na máquina *Win7-padrao*. Como esta configuração reagiria no caso de troca do *lease* pelo servidor DHCP?

3. Vamos partir para o controle por horário. Edite `/etc/squid3/acl/time.conf`:

```
MTWHF 00:00-06:00
MTWHF 19:00-23:59
```

No topo do bloco `acl`:

```
acl block_time time          "/etc/squid3/acl/time.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_time
```

Ajuste o relógio do sistema para um horário bloqueado, como 23h por exemplo:

```
# date -s 23:00:00
# hwclock --systohc
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*.

4. Vamos para o bloqueio de extensões. No arquivo `/etc/squid3/acl/regex_ext.conf`:

```
\.avi$
\.mp3$
\.pdf$
```

No topo do bloco `acl`:

```
acl block_ext  urlpath_regex "/etc/squid3/acl/regex_ext.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_ext
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. Para encontrar um arquivo que se ajuste ao bloqueio elaborado, basta pesquisar no Google algo como `site:ftp.openbsd.org filetype:pdf`, e clicar em um dos resultados.

5. Agora o bloqueio por palavras ocorrendo em URLs. Edite `/etc/squid3/acl/regex_word.conf`:

```
crypto
playboy
sexo
torrent
virus
```

No topo do bloco `acl`:

```
acl block_word urlpath_regex "/etc/squid3/acl/regex_word.conf"
```

Agora, no topo do bloco `http_access`:

```
http_access deny block_word
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. A URL http://www.openbsd.org/crypto.html é uma boa candidata para testar a efetividade do bloqueio.

6. Finalmente, vamos para o bloqueio por domínio. Edite `/etc/squid3/acl/domain.conf`:

```
.facebook.com
.instagram.com
.twitter.com
.whatsapp.com
```

No topo do bloco `acl`, inclua:

```
acl block_domain  dstdomain     "/etc/squid3/acl/domain.conf"
```

E no topo do bloco `http_access`:

```
http_access deny block_domain
```

Recarregue o Squid e teste o acesso na máquina *Win7-padrao*. A URL http://web.whatsapp.com é um bom exemplo do alvo do bloqueio.

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.

1. Assim como o Squid, o arquivo de configuração do SARG é imenso — 687 linhas. Dessas, apenas 43 são configurações efetivas.

```
# wc -l /etc/sarg/sarg.conf
687 /etc/sarg/sarg.conf

# grep -v '^#' /etc/sarg/sarg.conf | sed '/^$/d' | wc -l
43

# cp /etc/sarg/sarg.conf /etc/sarg/sarg.conf.orig
```

2. Vamos fazer o backup do arquivo original e trabalhar com algo mais gerenciável:

```
# mytemp=$(mktemp) && grep -v '^#' /etc/sarg/sarg.conf | sed '/^$/d' > $mytemp &&
mv $mytemp /etc/sarg/sarg.conf
```

3. Das linhas originais, precisamos alterar o valor de apenas seis, que se seguem:

```
access_log /var/log/squid3/access.log
output_dir /var/www/meusite/squid-reports
resolve_ip no
date_format e
use_comma no
charset UTF-8
```

4. Agora rode o comando `sarg`. Observe que o diretório `/var/www/meusite/squid-reports` foi criado automaticamente. Para tornar a geração de relatórios periódica, pode ser interessante agendar a execução do `sarg` no `cron` do sistema.

```
# sarg

# ls -1 /var/www/meusite/
index.html
restrito
squid-reports
```

5. Agora, basta acessar a URL `https://meusite.empresa.com.br/squid-reports/` e verificar os relatórios produzidos.

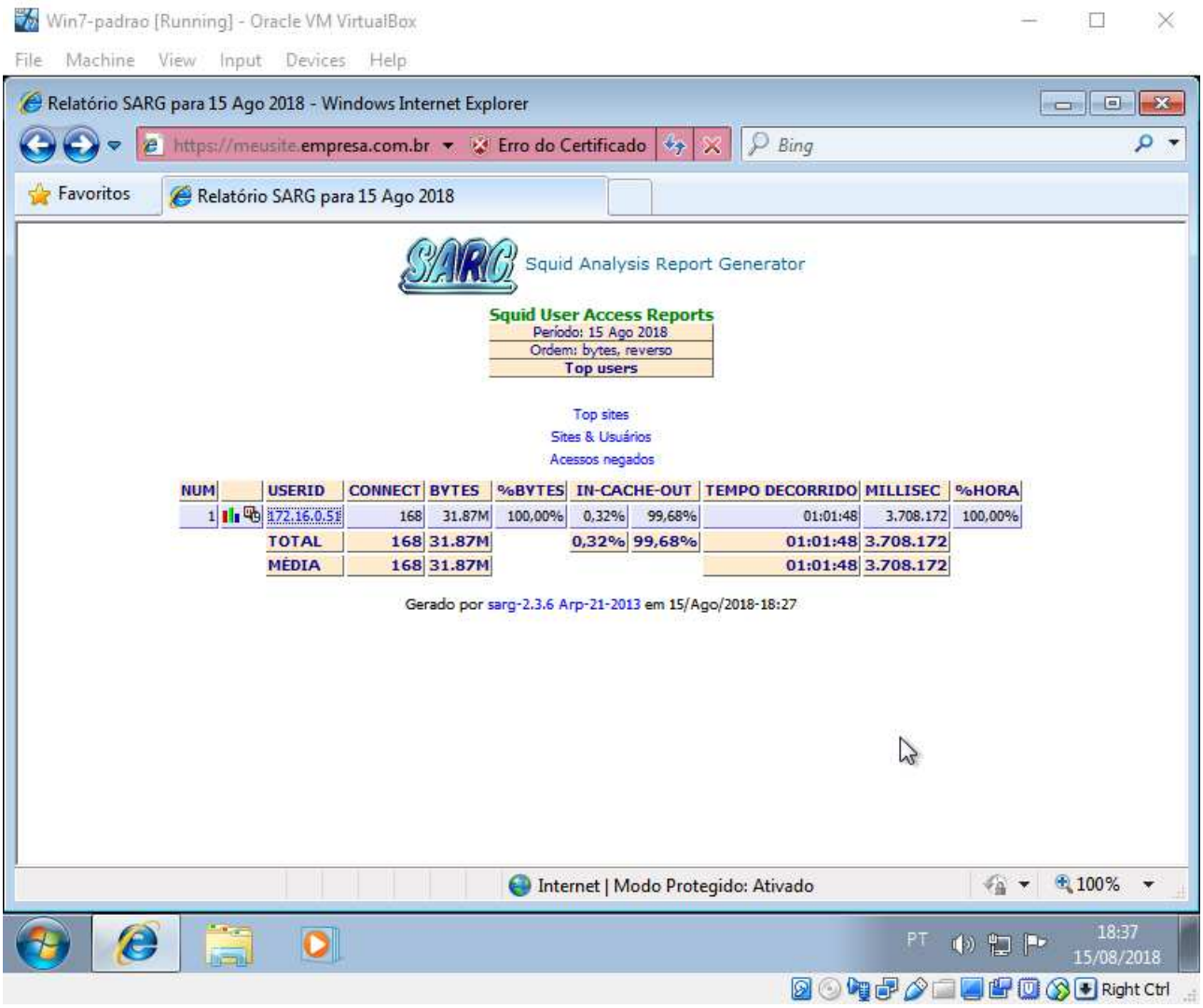

Figura 16: Visualização dos relatórios do SARG

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.

1. As configurações do firewall feitas através do comando `iptables` ficam apenas em memória, e se perdem após o `reboot` da máquina. Instale o `iptables-persistent` para corrigir isso:

   ```
   # apt-get install iptables-persistent
   ```

2. Verifique que as configurações do firewall estão vazias, exceto pelo *masquerading* que criamos na atividade inicial de configuração do laboratório:

```
# iptables -L -vn
Chain INPUT (policy ACCEPT 450 packets, 68775 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 213 packets, 53922 bytes)
 pkts bytes target     prot opt in     out     source               destination

# iptables -L -vn -t nat
Chain PREROUTING (policy ACCEPT 344 packets, 26393 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 334 packets, 25657 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 449 packets, 32617 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 77 packets, 5657 bytes)
 pkts bytes target     prot opt in     out     source               destination
  377 27216 MASQUERADE  all  --  *      eth0    0.0.0.0/0            0.0.0.0/0
```

3. Faça o `REDIRECT` de pacotes com destino à porta 80/HTTP para a porta 3128/TCP da máquina local. Depois, grave as configurações no arquivo `/etc/iptables/rules.v4`:

```
# iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-port 3128

# iptables-save > /etc/iptables/rules.v4
```

4. No `/etc/squid3/squid.conf`, configure a porta 3128 em modo transparente:

```
http_port 3128 transparent
```

5. De volta à máquina *Win7-padrao*, limpe as configurações de *proxy*:

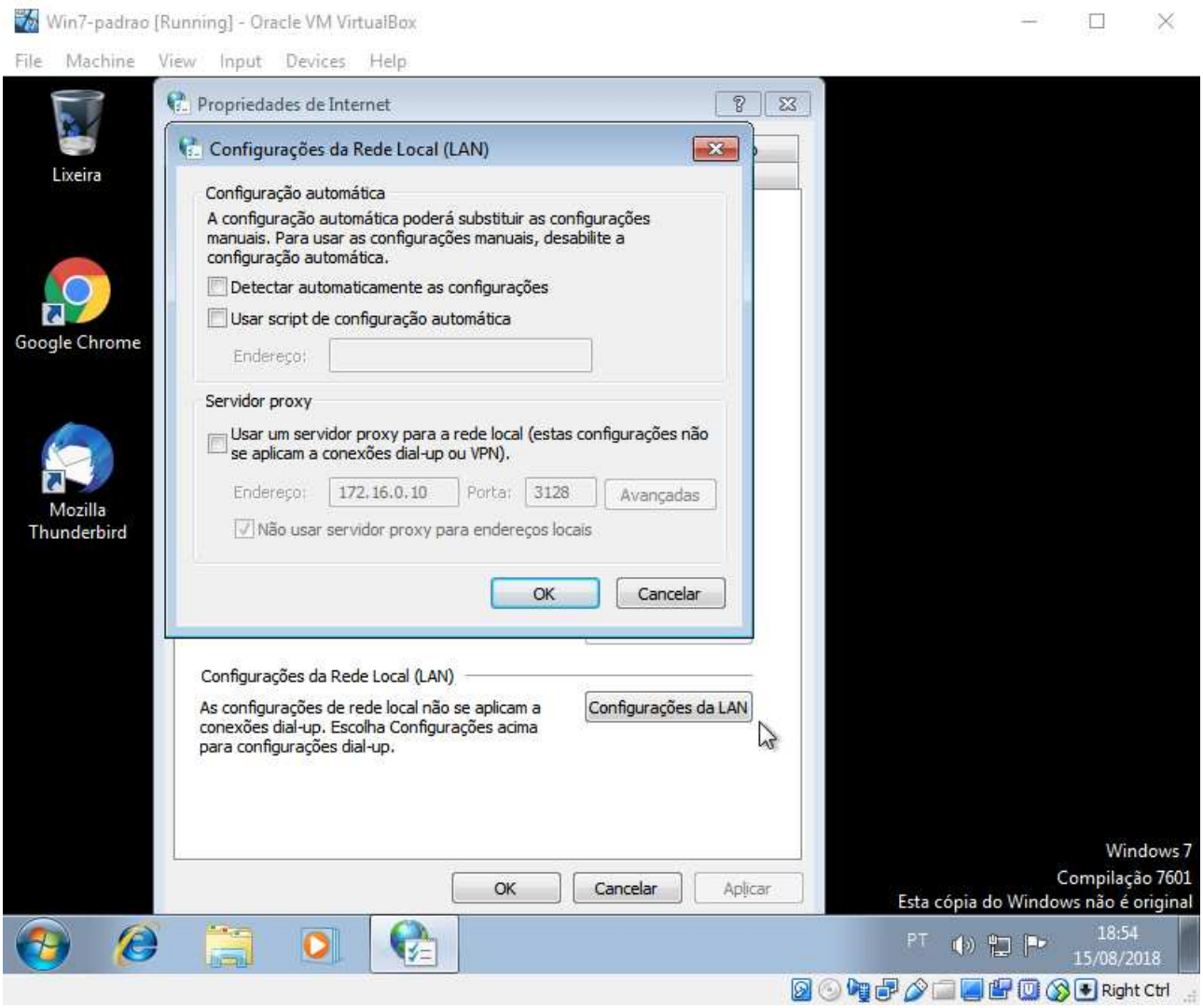

Figura 17: Configuração de proxy transparente

6. Finalmente, acesse uma URL na porta 80/HTTP e verifique se o *proxy* continua operacional.

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.