SEG12 - Semana 1 - Sessão 9

Francisco Marcelo, Marcelo Karam e Felipe Scarel

06-08-2018

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:

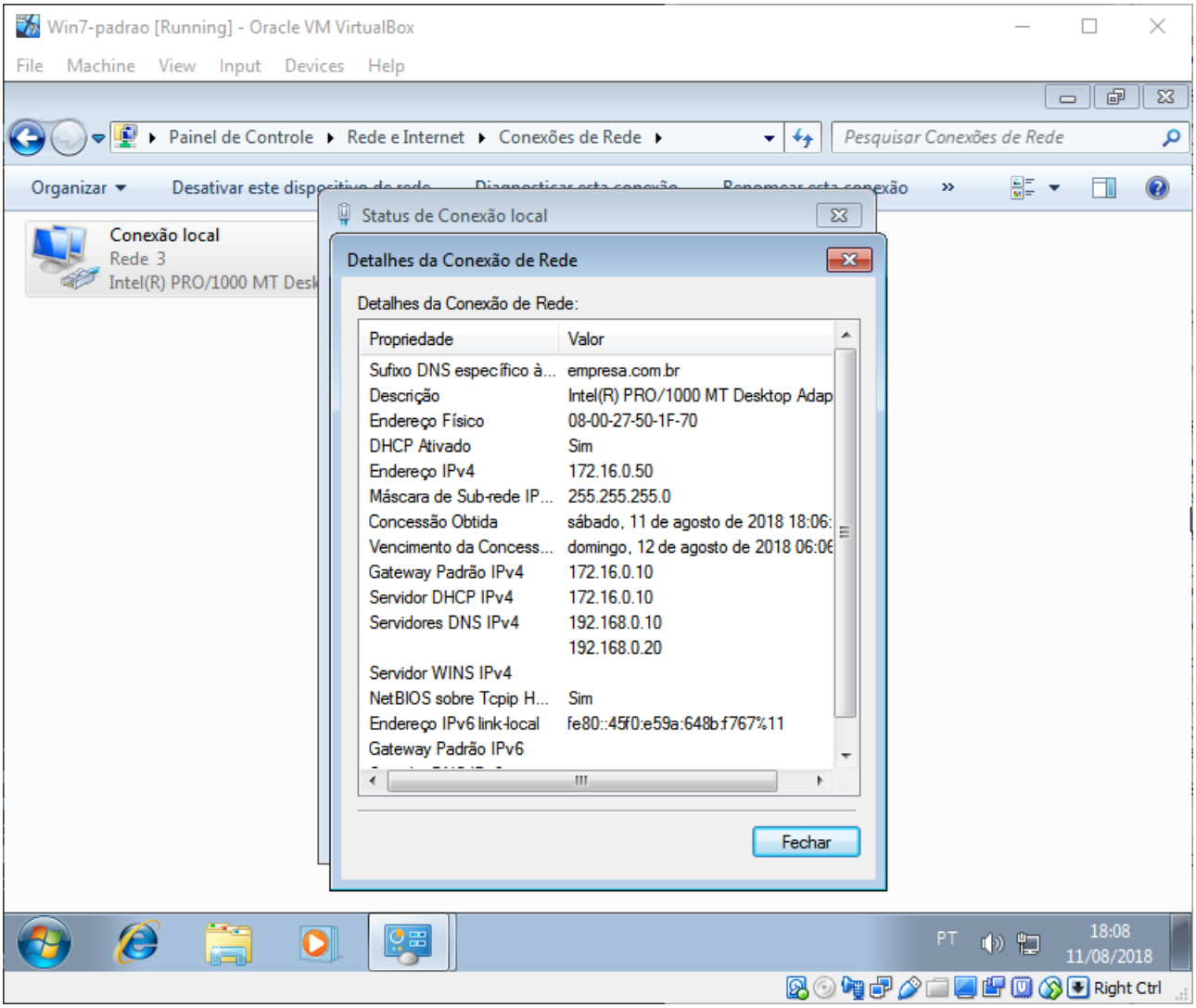

Figura 3: IP recebido via DHCP pelo Windows 7

6. Cheque se a resolução de nomes está operacional:

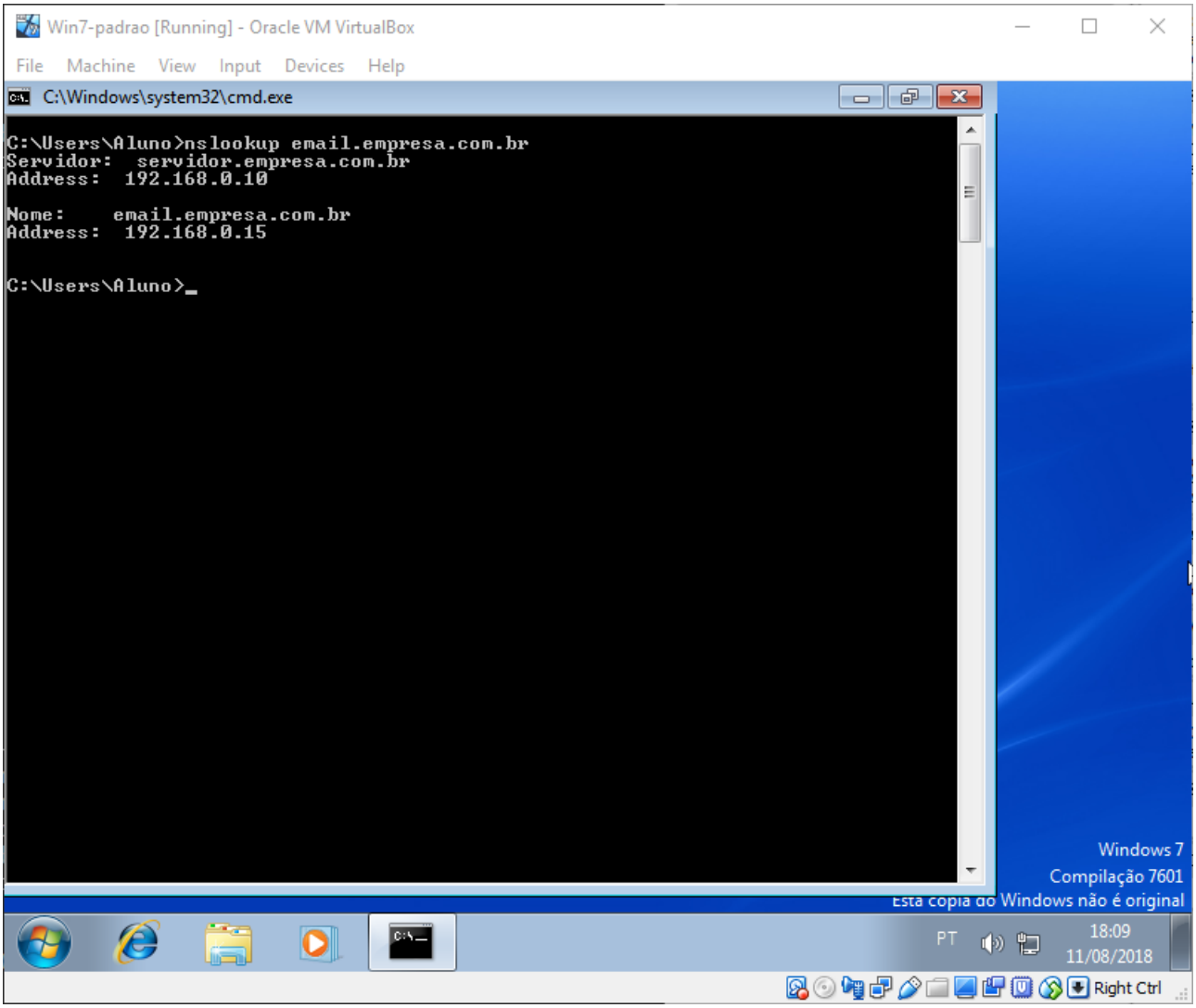

Figura 4: Resolução de nomes no Windows 7

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

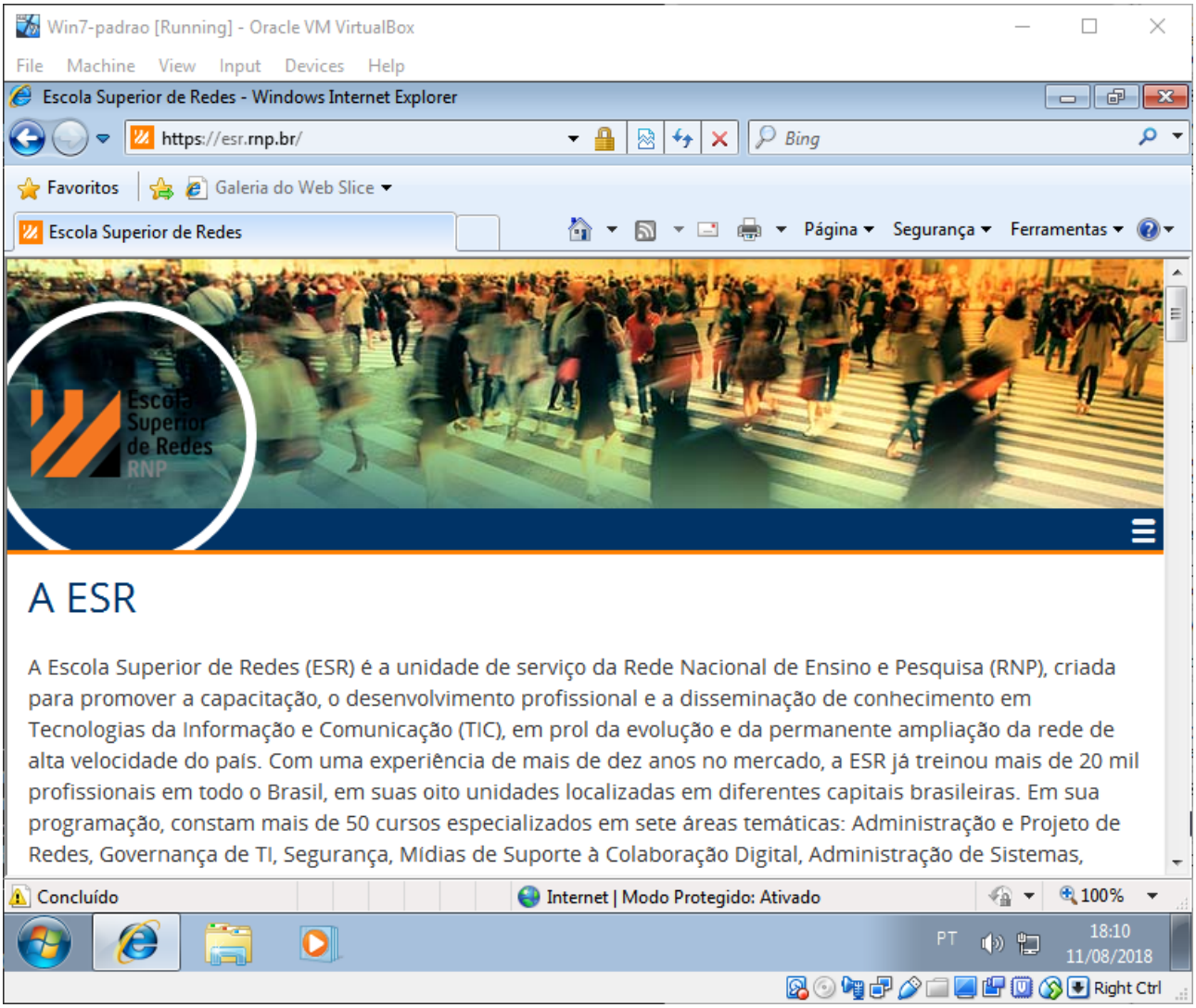

Figura 5: Navegação no Windows 7

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
```