Sessão 1: 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.

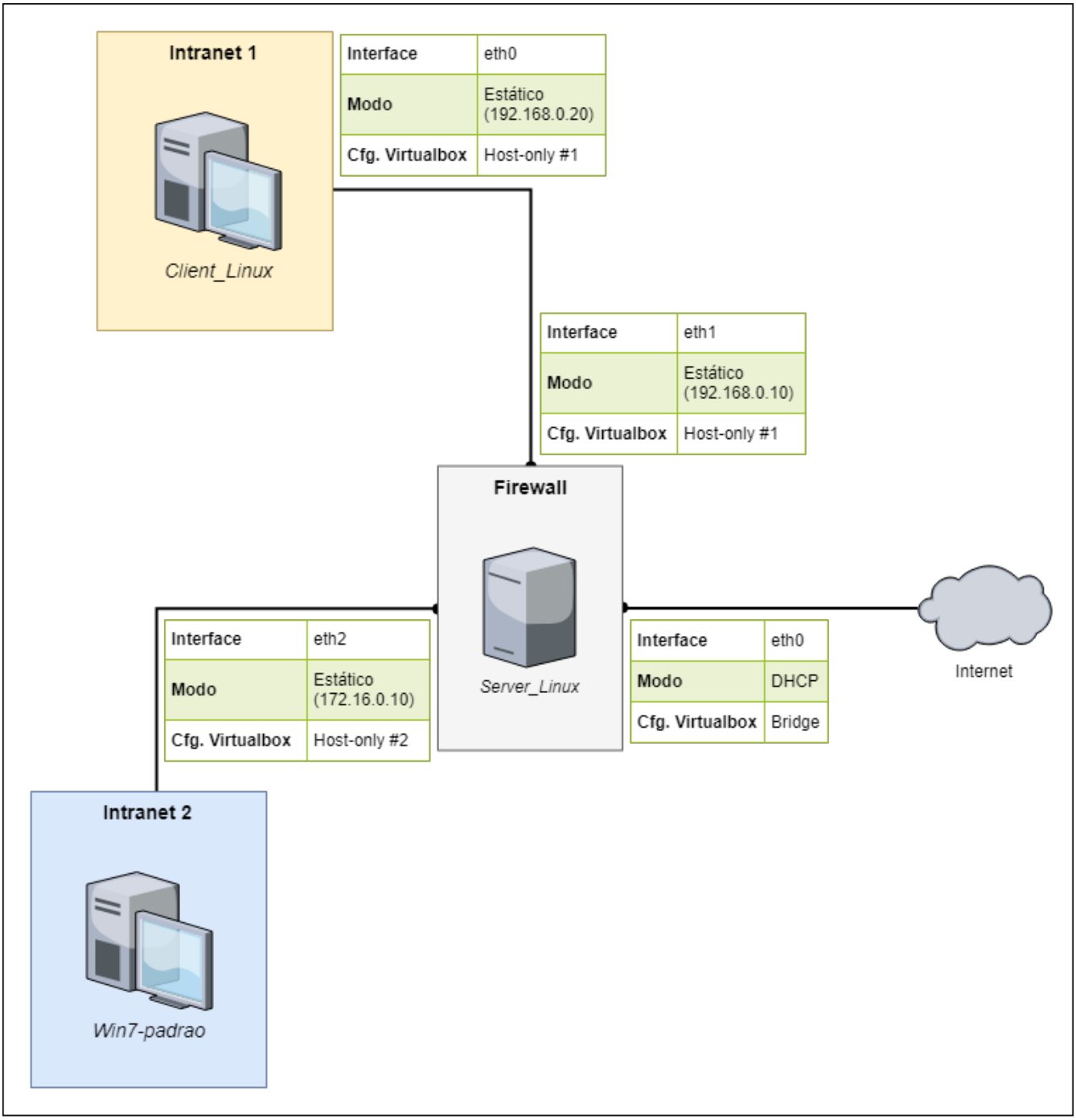

Figura 1: Topologia de rede do curso

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`.
2. Ao longo do curso, iremos editar vários arquivos de texto em ambiente Linux. Há vários

editores de texto disponíveis para a tarefa, como o `vi`, `emacs` ou `nano`. Caso você não esteja familiarizado com um editor de texto, recomendamos o uso do `nano`, que possui uma interface bastante amigável para usuários iniciantes. Para editar um arquivo com o `nano`, basta digitar `nano` seguido do nome do arquivo a editar — não é necessário que o arquivo tenha sido criado previamente:

```
# nano teste
```

Digite livremente a seguir. Use as setas do teclado para navegar no texto, e `DELETE` ou `BACKSPACE` para apagar texto. O `nano` possui alguns atalhos interessantes, como:

- `CTRL + G`: Exibir a ajuda do editor
- `CTRL + X`: Fechar o `buffer` de arquivo atual (que pode ser um texto sendo editado, ou o painel de ajuda), e sair do `nano`. Para salvar o arquivo, digite `Y` (*yes*) ou `S` (*sim*) para confirmar as mudanças ao arquivo, opcionalmente altere o nome do arquivo a ser escrito no disco, e digite `ENTER`.
- `CTRL + O`: Salvar o arquivo no disco sem sair do editor.
- `CTRL + W`: Buscar padrão no texto.
- `CTRL + K`: Cortar uma linha inteira e salvar no `buffer` do editor.
- `CTRL + U`: Colar o `buffer` do editor na posição atual do cursor. Pode ser usado repetidamente.

Para salvar e sair do texto sendo editado, como mencionado acima, utilize `CTRL + X`.

3. Agora, edite o arquivo `/etc/network/interfaces` como se segue, reinicie a rede e verifique o funcionamento:

```
# hostname
servidor
```

```
# whoami
root
```

```
# nano /etc/network/interfaces
(...)
```

```
# 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
```

Observe que como a interface `eth0` da máquina *Server_Linux* está em modo *bridge*, ela terá atribuída a si um endereço IP dinâmico, fornecido pelo servidor DHCP da sua rede. É provável que esse endereço seja diferente do exibido no exemplo acima.

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

```
# hostname
cliente
```

```
# whoami
root
```

```
# nano /etc/network/interfaces
(...)
```

```
# 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
```

Verifique se a resolução de nomes está corretamente configurada, checando o arquivo `/etc/resolv.conf`:

```
# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
```

5. 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:

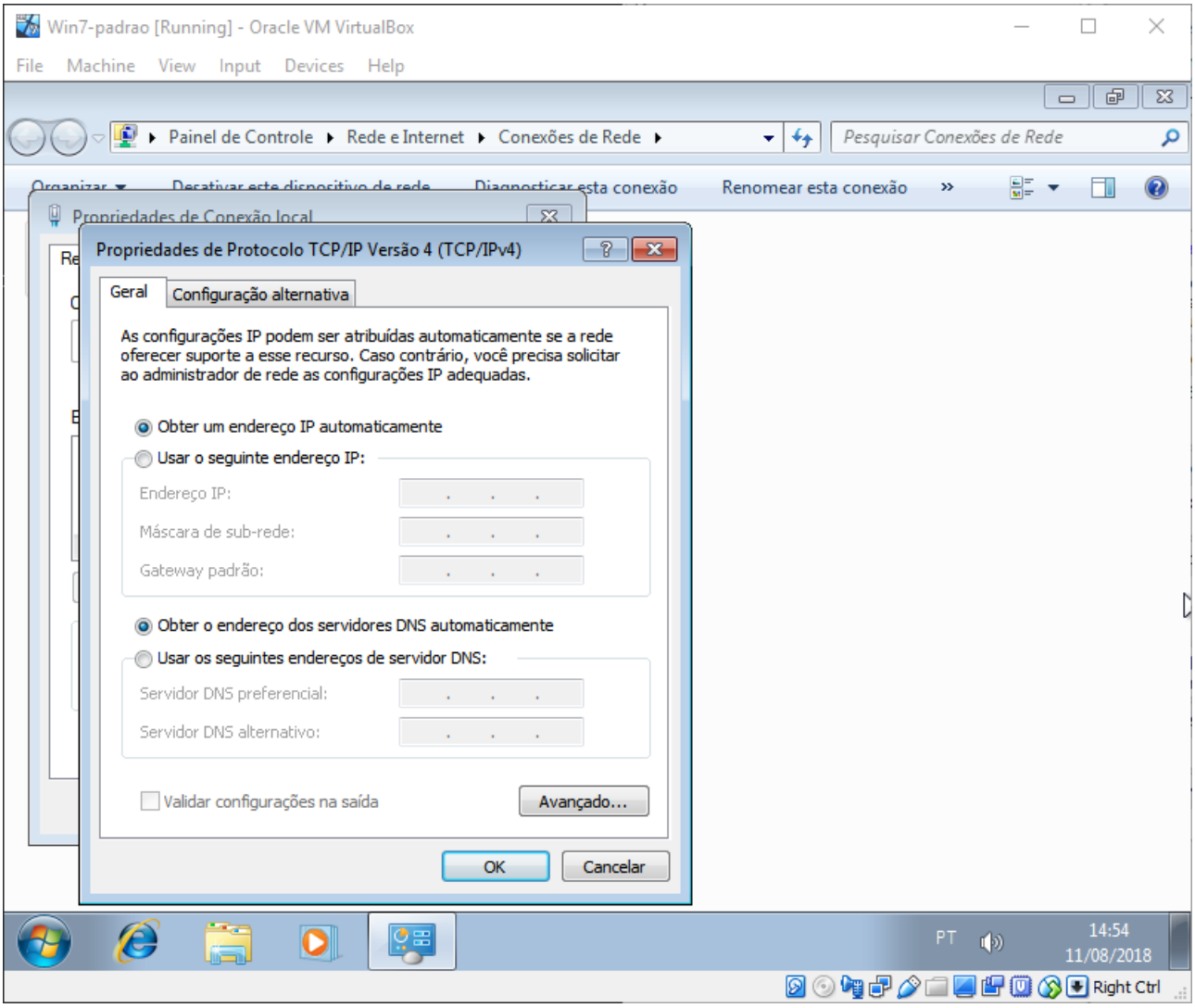

Figura 2: Configuração de rede da máquina Win7-padrao

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:

```
$ hostname
cliente
```

```
$ 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
```

Teste também a resolução de nomes. Tente, por exemplo, `ping` para o *host* `www.google.com`:

```
$ ping -c3 www.google.com
PING www.google.com (216.58.202.164) 56(84) bytes of data.
64 bytes from gru06s30-in-f164.1e100.net (216.58.202.164): icmp_seq=1 ttl=55
time=16.8 ms
64 bytes from gru06s30-in-f164.1e100.net (216.58.202.164): icmp_seq=2 ttl=55
time=16.9 ms
64 bytes from gru06s30-in-f164.1e100.net (216.58.202.164): icmp_seq=3 ttl=55
time=17.4 ms

--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 16.816/17.063/17.408/0.292 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.

```
# hostname
servidor
```

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