Segurança Informática e nas Organizações

Guiões das Aulas Práticas

João Paulo Barraca[1] e Hélder Gomes[2]

[1]Departamento de Eletrónica, Telecomunicações e Informática
[2]Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

2015–2016

Conteúdo

2

Firewalls com `iptables` em Linux

Resumo:

- Configuração de uma rede privada com uma *gateway* com NAT.
- Filtragem de pacotes com `iptables`.
- Mecanismos de encaminhamento de tráfego de entrada na rede privada.
- Configuração e exploração de um *proxy* HTTP (Squid).

2.1 Introdução

Neste trabalho pretende-se realizar uma demonstração prática dos conteúdos aprendidos acerca de *firewalls*. Comece por configurar a rede ilustrada na Fig. 2.1 e siga o guião.

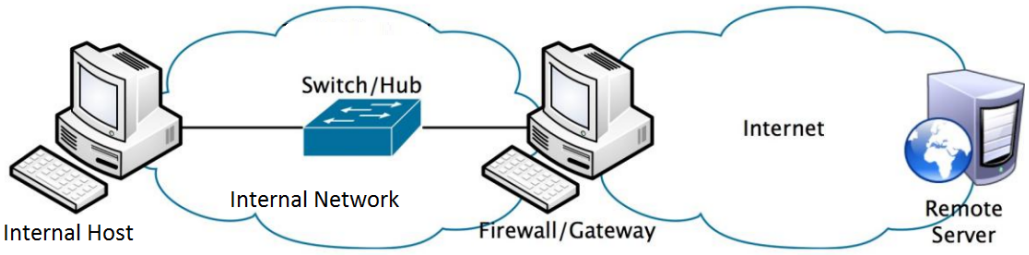

Figura 2.1: Arquitetura da rede para a demonstração da atuação de uma *firewall*.

Para implementar a rede da figura 2.1 pode usar as duas máquinas virtuais que criou e usou no guião 1. Tal como fez no referido guião, interligue as duas máquinas vituais usando interfaces de rede configuradas como interfaces de rede interna do Virtual Box. Quanto à interface que liga a máquina *Gateway* à rede externa, no VirtualBox configure-a como NAT.

Como todas as operações que serão realizadas neste guião carecem de privilégios de administração, para simplificar, considere o uso de uma consola de administração, que pode ser criada a partir de uma consola normal executando o comando

```
sudo bash
```

2.2 Montagem da rede

A *firewall*/*gateway* deverá esconder a existência de um rede interna privada perante a rede exterior. Nesse sentido, deverá realizar NAT relativamente ao tráfego que a atravessa, entre a rede interna e a exterior.

Para que a *gateway* se comporte como tal é preciso que seja ativado o encaminhamento de tráfego IP através de si, porque tal não o está por omissão. Para além disso, o encaminhamento deverá ser acompanhada pela devida tradução de endereços IP e seletores de transporte inerente ao funcionamento do NAT.

Em primeiro lugar é necessário configurar o Linux para fazer encaminhamento IP (*routing*), o que pode ser feito de três maneiras:

- através do comando

  ```
  echo 1 > /proc/sys/net/ipv4/ip_forward
  ```

- ou através do comando

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

- ou através da alteração do ficheiro

  ```
  /etc/sysctl.conf
  ```

  de modo a remover o comentário (`#`) da linha

  ```
  #net.ipv4.ip_forward=1
  ```

  a que se deve seguir a execução do comando

  ```
  sysctl -p
  ```

  para que a alteração seja aplicada.

A configuração do NAT é feita no `iptables` com o comando, assumindo que a rede interna usa endereços na gama 10.0.0.0/8, e que a interface para a rede externa é a `eth1`.

```
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth1 -j MASQUERADE
```

Este comando indica à tabela de NAT da cadeia de `POSTROUTING` (a última aplicada ao pacotes IP antes de o mesmo ser enviado para a rede) que todos os pacotes cujo endereço IP de origem pertença à rede `10.X.X.X` e que se destinem à rede exterior sejam transformados via NAT dinâmico (*masquerading*), i.e., que os seus endereços IP de origem sejam subsituidos pelo endereço IP da interface para a rede externa (`eth1`).

A listagem das regras NAT atuais do `iptables` é realizada com o comando

```
iptables -t nat -n -L
```

onde a opção `-n`, não sendo fundamental, evita traduções, que neste caso são indesejadas, entre endereços IP e nomes DNS.

Há uma decisão semelhante à MASQUERADE, que é a SNAT (de Source NAT). Consulte o manual do `iptables` com o comando

```
man iptables
```

para perceber as diferença entre essas decisões e conclua sobre qual a mais adequada neste caso.

2.2.1 Salvaguarda das regras do `iptables`

O `iptables` não salvaguarda as regras com as quais foi configurado, o que faz com que as regras se percam sempre que o sistema se desliga. É no entanto possível salvaguardar as regras para um ficheiro e posteriormente recarregá-las sempre que for necessário. Salvaguarde as regras do `iptables` num ficheiro, apague-as integralmente e reponha a configuração do `iptables` anteriormente salvaguardada com os seguintes comandos:

```
iptables-save > /etc/iptables.save
iptables -F
iptables-restore < /etc/iptables.save
```

2.2.2 Configuração das máquinas da rede interna

Para simplificar a configuração da máquina interna, vamos instalar um servidor `dnsmasq` na máquina `Gateway`, com o comando

```
apt-get install dnsmasq
```

Este servidor possui um serviço de DHCP que é útil para configurar a rede interior (endereços IP, máscara de rede e servidor DNS) e para fazer o reencaminhamento de pedidos DNS através de um *gateway* com NAT. Para este fim, altere o respetivo ficheiro de configuração `/etc/dnsmasq.conf` para definir a interface da *gateway* que liga à rede interna, que aqui assumimos que é a interface `eth0`

```
interface=eth0
```

indicar o endereço IP do servidor de DHCP

```
dhcp-host=10.0.0.1
```

e definir a gama de endereços IP fornecidos pelo DHCP para as máquinas da rede interna.

```
dhcp-range=10.0.0.2,10.0.0.254,255.0.0.0
```

Feitas estas configurações, reinicie o serviço com o comando

```
service dnsmasq restart
```

No final desta configuração qualquer máquina que se ligar na rede interna deverá obter automaticamente a sua configuração de rede e conseguir ligar-se à rede exterior através do *gateway*. Verifique este facto antes de prosseguir!

2.3 Controlo de tráfego

2.3.1 Filtragem de tráfego com destino a certas máquinas

Nesta secção vamos exemplificar a aplicação de regras de filtragem de tréfego da rede interna para o exterior. Em particular, iremos configurar a *firewall* de tal forma que as máquinas da rede interior não consigam aceder ao servidor `www.youtube.com`, o que se consegue com o comando

```
iptables -A FORWARD -d www.youtube.com -j DROP
```

Experimente a sua eficácia tentando aceder ao servidor bloqueado a partir de uma máquina da rede interna.

Remova esse regra (usando a opção `-D` em vez da `-A`), e instale uma nova onde limita a filtragem à maquina cliente que possui na rede interna (vamos assumir que o seu endereço de rede é `10.0.0.2`):

```
iptables -A FORWARD -s 10.0.0.2 -d www.youtube.com -j DROP
```

Experimente novamente a sua eficácia tentando aceder ao servidor bloqueado a partir da máquina cujo endereço IP é `10.0.0.2`.

2.3.2 Decisões DROP vs. REJECT

Altere as regras anteriores substituindo a decisão DROP por uma decisão REJECT. Verifique o resultado prático dessa decisão.

2.3.3 Registo de tráfego

Nesta secção vamos exemplificar a aplicação de regras que criam registos de atividade de rede, os quais poderão ser úteis para analisar à posterior que tráfego foi efetivamente processo pela *firewall* e por uma regra em particular. No seguimento da secção anterior, vamos configurar o `iptables` para que os pacotes rejeitados devido aos comandos da alínea anterior apareçam evidenciados num registo:

```
iptables -I FORWARD -s 10.0.0.2 -d www.youtube.com -j LOG --log-prefix "DROP "
```

Este comando faz com que os pacotes rejeitados em consequência da regra da alínea anterior apareçam evidenciados num `log`. Note-se que ele só funcionará se a regra for colocada (processada) antes das regras da alínea anterior,

o que não aconteceria se tivéssemos usado a opção `-A` (*append*, ou seja, colocar no fim). Por isso, neste caso usou-se a opção `-I` (*insert*, sem número de ordem), o que coloca a regra no início de todas as demais na cadeia `FORWARD`.

Use o seguinte comando para listar todas as regras de filtragem após a instalação desta última regra:

```
iptables -L
```

2.3.4 Filtragem de tráfego consoante os serviços acedidos

Use o seguinte comando para que não seja possível à máquina da rede interna aceder a qualquer serviço exterior excepto HTTP:

```
iptables -I FORWARD -s 10.0.0.2 -p tcp ! --dport 80 -j DROP
```

Explique os parâmetros usados recorrendo para esse efeito ao manual interativo do `iptables`.

2.4 Encaminhamento de tráfego do exterior para a rede interior

Deviso à existência de NAT, não é possível da rede exterior aceder a serviços da rede interior sem realizar alguma configuração que torne tal possível. Essa configuração designa-se por *port forwarding*.

Admitamos que existe um servidor Web na máquina `10.0.0.2` que quer tornar acessível a partir da rede exterior. Neste caso, precisa de realizar uma operação de *port forwarding* na *gateway* que encaminhe um porto TCP/IP da interface exterior para o porto TCP/IP do servidor, na rede interna. O seguinte comando faz essa operação para um servidor Web padrão, ou seja, que usa o porto TCP 80, usando para o efeito um porto TCP exterior com o mesmo valor:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -d X.X.X.X  \
    -j DNAT --to 10.0.0.2
```

onde `X.X.X.X` representa o endereço IP da interface exterior do *gateway*.

2.5 Servidor Squid

O servidor Squid é dos mais famosos *proxies* HTTP para ambientes Linux, devido à sua facilidade de utilização e flexibilidade. Este servidor permite fazer vários tipos de filtragens e operações ao nível aplicacional, as quais iremos exercitar neste trabalho.

Para realizar o trabalho reutilize a rede anterior e configure o navegador na máquina `10.0.0.2` para usar o *proxy* HTTP que irá instalar na máquina *gateway*. Esta configuração é feita no navegador (configuração de rede), e deve ser feita para o *proxy* HTTP no endereço 10.0.0.1 e porto 3128.

2.5.1 Instalação e configuração base

Instale o Squid na máquina *gateway* com o comando

```
apt-get install squid3
```

Uma vez instalado, é preciso configurar o Squid para permitir o acesso a páginas Web a partir das máquinas da rede interna. Para isso, edite o ficheiro `/etc/squid3/squid.conf` e introduza a linha

```
acl rede_interna src 10.0.0.0/24
```

na secção `ACL (TAG: acl)`, e a linha

```
http_access allow rede_interna
```

na secção `http_access (TAG: http_access)`, a seguir à linha que começa com `#INSERT YOUR OWN RULES(S) HERE ....`
Nota: não esquecer que é necessário reiniciar o Squid para que as novas regras tenham efeito. Para isso deverá ser usado o comando

```
service squid restart
```

2.5.2 Filtragem de acessos HTTP via Squid

Vamos agora bloquear todos os acessos ao servidor `www.abola.pt` via Squid. Para isso, no ficheiro de configuração do mesmo deverá introduzir a seguinte linha na secção `ACL (TAG: acl)`

```
acl bloqueados dstdomain "/etc/squid3/bloq.conf"
```

e no ficheiro `/etc/squid3/bloq.conf` deverá colocar a lista de destinos proibidos, um por linha. Por fim, no ficheiro de configuração do Squid deverá colocar a linha

```
http_access deny bloqueados
```

na secção `http_access (TAG: http_access)`, imediatamente antes da linha antes adicionada para permitir o acesso das máquinas da rede interna.

Verifique na prática a eficácia da configuração realizada, antes e depois da mesma entrar em efeito.

2.5.3 Filtragem de conteúdos HTTP via Squid

Vamos agora bloquear todos os acessos via Squid a URLs com determinadas palavras (por exemplo, `facebook`). Para isso, no ficheiro de configuração do mesmo deverá introduzir a seguinte linha na secção `ACL (TAG: acl)`

```
acl palavras url_regex "/etc/squid3/palavras.conf"
```

e no ficheiro `/etc/squid3/palavras.conf` deverá colocar a lista de padrões proibidos nos URL, um por linha. Por fim, no ficheiro de configuração do Squid deverá colocar a linha

```
http_access deny palavras
```

na secção `http_access (TAG: http_access)`.

Verifique na prática a eficácia da configuração realizada, antes e depois da mesma entrar em efeito.

2.5.4 Controlo de horários de acesso

Vamos agora bloquear todos os acessos aos servidores `www.abola.pt` e `www.record.xl.pt` via Squid durante o intervalo 9h00-18h00 dos dias úteis. Para isso, no ficheiro de configuração do mesmo deverá colocar as linhas

```
acl horario_trabalho time M T W H F 9:00-18:00
acl desportivos dstdomain .abola.pt .record.lx.pt
```

na secção `http_access (TAG: http_access)` e a linha

```
http_access deny horario_trabalho desportivos
```

na secção `http_access (TAG: http_access)`.

Verifique na prática a eficácia da configuração realizada, antes e depois da mesma entrar em efeito.

2.5.5 Modo transparente

O Squid poderá ser usado em modo transparente, que consiste na sua utilização pelas aplicações clientes (navegadores) sem que estes tenham sido configurados para o usar (daí a noção de exploração transparente para os clientes). Para testar esta funcionalidade, remova a configuração do *proxy* que anteriormente realizou no navegador cliente.

Para funcionar neste modo, o ficheiro de configuração do Squid deverá conter a linha

```
http_port 3128 transparent
```

na secção `http_port (TAG: http_port)`.

Feita esta configuração do Squid, é preciso que todo o tráfego HTTP iniciado pela rede interna e com destino (direto) à rede exterior. Para isso, deverá ser usadas a seguinte regra do `iptables`:

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 \
   -j REDIRECT --to-ports 3128
```

Com esta regra, todo o tráfego HTTP vindo da rede interior não é simplesmente reencaminhado para a rede exterior, mas sim reencaminhado para para o porto TCP 3128 da máquina local, que não é mais do que o porto de transporte do *proxy* Squid local.

2.6 Bibliografia

- *Network Address Translation* (NAT), `http://en.wikipedia.org/wiki/Network_address_translation`
- `dnsmasq`, `http://en.wikipedia.org/wiki/Dnsmasq`
- `iptables`, `http://en.wikipedia.org/wiki/Iptables`
- Squid, `http://en.wikipedia.org/wiki/Squid_%28software%29`