Practical Assignment #1

João Neto – 2023234004

Vasco Alves – 2022228207

22 de março de 2026

Conteúdo

1 Introduction

O objetivo principal deste trabalho era aprender IPTables e como configurar um com o Suricata um sistema de filtração e deteção de ataques. Para esse fim, foi simulado um sistema dividido em três redes e um router para conectar-las. As três redes são a DMZ (23.214.219.128/25, enp0s8), Internal network (192.168.10.0/24, enp0s9) e Internet (87.248.214.0/24, enp0s10). As três redes tem varios serviços, o DMZ tem dns(23.214.219.130), mail(23.214.219.134), vpn-gw(23.214.219.133), www(23.214.219.132) e smpt(23.214.219.131). A Internal network tem ftp(192.168.10.2), datastore(192.168.10.3) e clientes (nos testes os clientes tem ip 192.168.10.4, mas está configurado para dar para qualquer edereço). Por fim a rede Internet tem dns2 (87.248.214.99) e eden (87.248.214.100), existe também outros serviços (87.248.214.98). Para facilitar a recriação deste sistema foi criado 4 ficheiros .sh (um para cada rede e o router), e disponibilizamos os ficheiros suricata.rules e suricata.yaml, para o suricata que estiver ligado ao Router. Os ficheiros .sh vão ter comandos para configurar o sistema para este exercicio.

2 Firewall

2.1 Packet fileter without NAT

O policy que foi escolhido foi:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

Foi escolhido porque é mais facil dar DROP a todos os pacotes que não foi criado regras do que criar uma regra de DROP para todos os protocolos e possibilidades, o OUTPUT ficou para ACCEPT porque não existe razão para dar DROP dos pacotes que estamos a enviar neste trabalho. Para o router conseguir resolver DNS requests e para aceitar coneções SSH da rede interna ou da VPN gateway foi utilizado estes comandos:

```
sudo iptables -A INPUT -o enp0s10 -p udp --dport 53 -j ACCEPT
sudo iptables -A INPUT -i enp0s9 -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -i enp0s8 -s 23.214.219.133 -p tcp --dport 22 -j ACCEPT
```

Para conseguirmos a confirguração pedida entre redes foi utilizado estes commandos:

```
sudo iptables -A FORWARD -i enp0s8 -o enp0s10 -s 23.214.219.130 -p udp --dport 5
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.130 -p udp --dport 53
sudo iptables -A FORWARD -i enp0s8 -o enp0s10 -s 23.214.219.130 -p tcp --dport 5
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.131 -p tcp --dport 58
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.134 -p tcp --dport 14
```

```
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.134 -p tcp --dport 11
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.132 -p tcp --dport 80
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.132 -p tcp --dport 44
sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -d 23.214.219.133 -p udp --dport 11
sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -s 23.214.219.133 -d 192.168.10.2 -
sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -s 23.214.219.133 -d 192.168.10.3 -
```

2.2 Packet filtering with NAT

Para conecções com origem/destino na internet foi utilizado DNAT/SNAT e iptables para ”esconder”o ip para a internet que quer aceder a rede interna e iproutes para bloquear certos pacotes de entrar, para conseguir a configuração utilizamos estes comandos:

```
sudo iptables -A FORWARD -i enp0s10 -o enp0s9 -d 192.168.10.2
-p tcp --dport 21 -j ACCEPT
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p tcp --sport 20 -j ACCEPT
sudo iptables -t nat -A PREROUTING -s $dns2 -p tcp --dport 22
 -j DNAT --to-destination 192.168.10.3
sudo iptables -t nat -A PREROUTING -s $eden -p tcp --dport 22
 -j DNAT --to-destination 192.168.10.3
sudo iptables -t nat -A PREROUTING -i enp0s10 -p tcp --dport 21
 -j DNAT --to-destination 192.168.10.2
sudo iptables -A FORWARD -i enp0s10 -o enp0s9 -d 192.168.10.3 -s $dns2
 -p tcp --dport 22 -j ACCEPT
sudo iptables -A FORWARD -i enp0s10 -o enp0s9 -d 192.168.10.3 -s $eden
 -p tcp --dport 22 -j ACCEPT
sudo iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o enp0s10 -j SNAT
  --to-source 87.248.214.97
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p udp --dport 53 -j ACCEPT
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p tcp --dport 80 -j ACCEPT
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p tcp --dport 443 -j ACCEPT
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p tcp --sport 21 -j ACCEPT
sudo iptables -A FORWARD -i enp0s9 -o enp0s10 -p tcp --dport 21 -j ACCEPT
```

3 Intrusion Detection

As regras que utilizamos para o suricata foram estas:

```
drop tcp \$EXTERNAL\_NET any -> \$HOME\_NET any (msg:"ET"; flags:S; threshold:ty
drop tcp any any -> any 80 (msg:"SQL injection"; content:"union"; nocase; conten
```

```
drop tcp any any -> any 80 (msg:"SQl injection"; content:"'or 1=1"; nocase; clas
drop tcp any any -> any 80 (msg:"XSS"; content:"<script"; nocase; classtype:web-
```

A primeira é para port scaning, a segunda e a terceira é para o caso de SQL injection, e a ultima é para XSS atacks.

4 Tests utilizados

Netcat foi utilizado para maior

5 Conclusion

Fuck we learned alot