

Universidade de Coimbra



Trabalho Prático 2

Mestrado de Segurança Informática

Segurança em tecnologia de Informação

André Ferreira - uc2023188951@student.uc.pt

Nuno Simões - uc2023173985@student.uc.pt

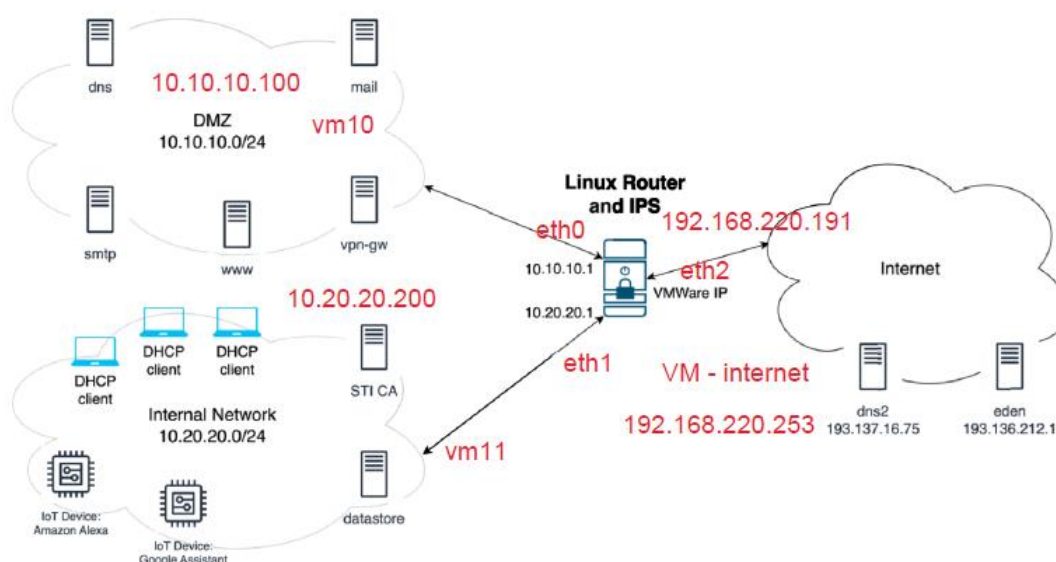
Índice

Introdução.....	3
Packet filtering e NAT usando NFtables	4
Configuração da firewall para proteger o router	4
Configuração da firewall para autorizar comunicações diretas (sem NAT)	6
Configuração da firewall para ligações ao endereço IP externo da firewall (utilizando NAT).....	11
Configuração da firewall para comunicações da rede interna para o exterior (utilizando NAT).....	13
Deteção e prevenção de intrusões (IDS/IPS)	15
Detetar e bloquear (pelo menos) os seguintes ataques	15
Audit any modification in the firewall and IDS/IPS configuration files.....	18
Conclusão.....	19
Referências	20

Introdução

Este trabalho foi realizado no âmbito da cadeira de Segurança em Tecnologias da Informação e visa a implementação de um cenário com a utilização de 4 máquinas virtuais. Os objetivos gerais deste trabalho são os seguintes:

- Configuração dos serviços que são fornecidos na DMZ, na rede interna e no Router Linux.
- Configuração de uma Firewall no Router Linux através de Nftables.
- Configuração do Suricata é um sistema de prevenção de intrusões (IPS), onde ele não só deteta, mas também pode tomar ações para bloquear ou alterar pacotes de rede maliciosos. Isso permite uma resposta proativa a ameaças, mitigando potencialmente o dano antes que ele ocorra.
- Auditar ficheiros importantes de configuração, detetar atividades suspeitas ou maliciosas, como acesso não autorizado ou modificações em arquivos críticos.



Este trabalho tem o cenário acima descrito, composto por quatro máquinas virtuais em VMware, no qual as máquinas “router” e a “VM-Internet” podem variar o IP, visto que este é por DHCP.

Em termos de endereçamento IP, existem 3 redes, DMZ “10.10.10.0/24”, Internal “10.20.20.0/24” e a rede da Internet “192.168.220.0/24”.

As máquinas virtuais contêm os seguintes IPs, VM-DMZ “10.10.10.100”, VM-STI “10.20.20.200”, VM-Internet “192.168.220.185” e Router “eth0- 10.10.10.1”, “eth1- 10.20.20.1” e “eth2- 192.168.220.177”.

Em termos de “máquinas” externas alvos dos nossos testes, está o dns da Universidade do Minho “193.137.16.75”, Cloudflare “1.1.1.1”, Google “8.8.8.8” e Eden “193.137.203.227”.

De forma a termos uma aproximação maior à realidade, decidimos fazer os testes, com serviços instalados e configurados, nomeadamente o DNS(bind9), SMTP(postfix), IMAP(dovecot), WEB(apache2), VPN(openvpn). Para fazer os pedidos de ligação, maioritariamente foi com a ferramentas / programas: NetCat, Hping3, Hydra, mysql, ssh.

Packet filtering e NAT usando NFtables

Configuração da firewall para proteger o router

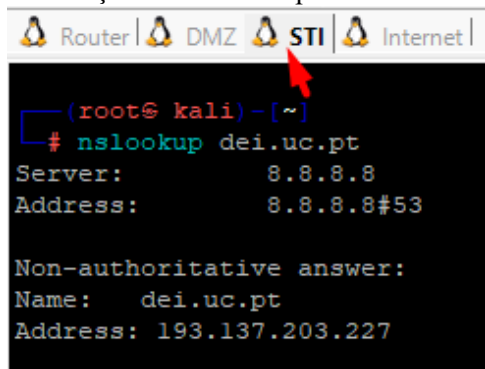
“A configuração da firewall deve eliminar todas as comunicações que entram no sistema do router, exceto as necessárias para o funcionamento normal dos seguintes serviços:”

- Pedidos de resolução de nomes DNS enviados para servidores externos.

Linha de código NFT:

```
Chain forward { udp dport 53 ip daddr 8.8.8.8 counter accept }
```

#Execução do comando para fazer a resolução de nome.

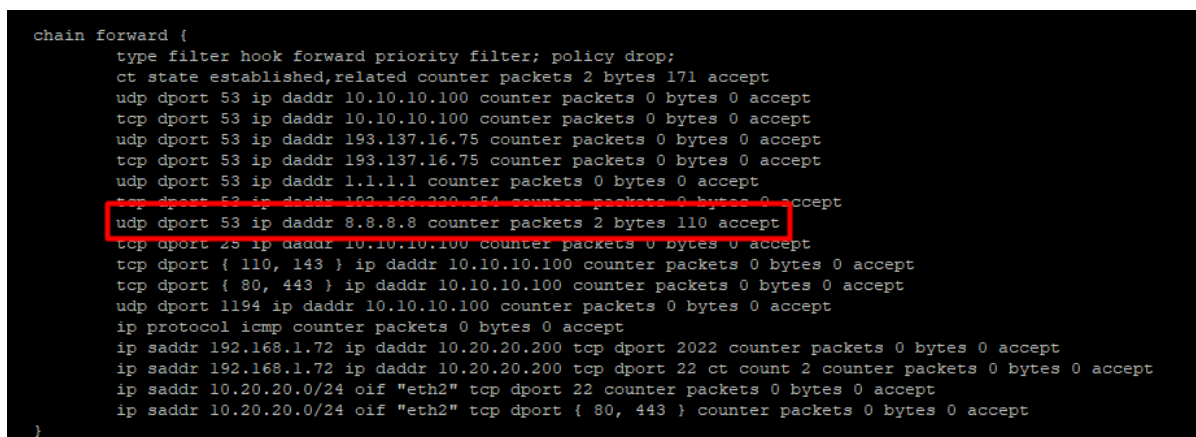


The screenshot shows a network diagram at the top with four nodes: Router, DMZ, STI, and Internet. Below the diagram, a terminal window shows the command `nslookup dei.uc.pt` being executed. The output shows the server as 8.8.8.8 and the address as 8.8.8.8#53. A non-authoritative answer is also shown with the name dei.uc.pt and address 193.137.203.227.

```
(root@ kali)~# nslookup dei.uc.pt
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
Name:   dei.uc.pt
Address: 193.137.203.227
```

#Correspondência dos pacotes de rede com as regras nftables.



The screenshot shows a terminal window with nftables configuration rules. The rules are for a chain named 'forward'. The rules include: a type filter hook, a ct state established rule, and several rules for UDP and TCP ports 53, 110, 143, 80, 443, and 1194. The rule for UDP port 53 with IP address 8.8.8.8 is highlighted with a red box.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 2 bytes 171 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 2 bytes 110 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 192.168.1.72 ip daddr 10.20.20.200 tcp dport 2222 counter packets 0 bytes 0 accept
    ip saddr 192.168.1.72 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept
}
```

●Ligações SSH ao sistema de router, se forem originadas na rede interna ou no gateway VPN (vpn-gw).

“As ligações SSH também devem ser protegidas com o mecanismo de bloqueio de portas (com um mínimo de 5 bloqueios de portas). Deve-se documentar a sequência secreta.”

A sequência do acesso SSH ao router segue a seguinte ordem de portos: 123, 234, 789, 345 e 456, a seguir ele vai iniciar a ligação no porto 22.

Linhas de código NFT:

```
table inet portknock {
    chain debug {
        type filter hook prerouting priority -301;
        meta nftrace set 1;
        # ip protocol (tcp) meta nftrace set 1
    }

    set clients_ipv4 {
        type ipv4_addr
        flags timeout
        counter
    }

    set candidates_ipv4 {
        type ipv4_addr . inet_service
        flags timeout
        counter
    }

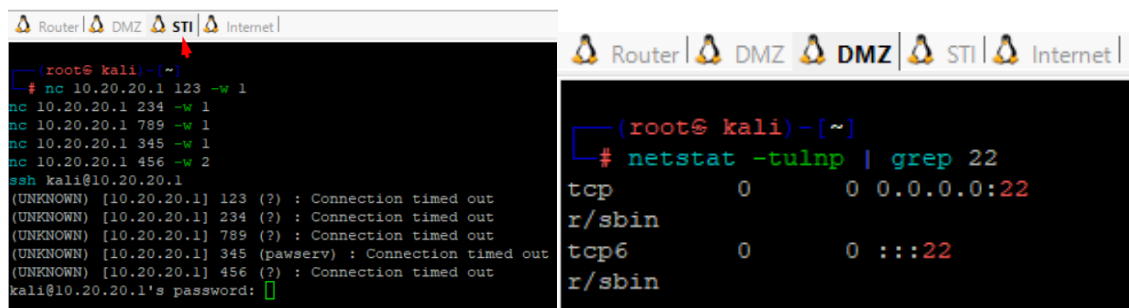
    chain input {
        type filter hook input priority -10; policy drop;

        iifname "lo" return

        # cinco portas para port knock
        tcp dport 123 add @candidates_ipv4 {ip saddr . 234 timeout 60s}
        tcp dport 234 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 {ip saddr . 789 timeout 60s}
        tcp dport 789 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 {ip saddr . 345 timeout 60s}
        tcp dport 345 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 {ip saddr . 456 timeout 60s}
        tcp dport 456 ip saddr . tcp dport @candidates_ipv4 add @clients_ipv4 {ip saddr timeout 60s} log prefix "Successful portknock: "

        tcp dport $guarded_ports ip saddr @clients_ipv4 counter accept
        tcp dport $guarded_ports ct state established,related counter accept
        tcp dport $guarded_ports counter drop
    }
}
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)-[~]
└─# nc 10.20.20.1 123 -w 1
nc 10.20.20.1 234 -w 1
nc 10.20.20.1 789 -w 1
nc 10.20.20.1 345 -w 1
nc 10.20.20.1 456 -w 2
ssh kali@10.20.20.1
(UNKNOWN) [10.20.20.1] 123 (?): Connection timed out
(UNKNOWN) [10.20.20.1] 234 (?): Connection timed out
(UNKNOWN) [10.20.20.1] 789 (?): Connection timed out
(UNKNOWN) [10.20.20.1] 345 (pawserv): Connection timed out
(UNKNOWN) [10.20.20.1] 456 (?): Connection timed out
kali@10.20.20.1's password:

Router | DMZ | STI | Internet |
└─# netstat -tulnp | grep 22
tcp        0      0 0.0.0.0:22
r/sbin
tcp6       0      0 :::22
r/sbin
```

#Correspondência dos pacotes de rede com as regras nftables.

```
(root@kali)~  
# nft list ruleset  
table inet portknock {  
    set clients_ipv4 {  
        type ipv4_addr  
        size 65535  
        flags dynamic,timeout  
        counter  
        elements = { 10.20.20.200 counter packets 16 bytes 3820 timeout 1m expires 10s188ms }  
    }  
  
    set candidates_ipv4 {  
        type ipv4_addr . inet_service  
        size 65535  
        flags dynamic,timeout  
        counter  
        elements = { 10.20.20.200 . 789 counter packets 4 bytes 240 timeout 1m expires 5s744ms,  
                    10.20.20.200 . 345 counter packets 3 bytes 180 timeout 1m expires 8s148ms,  
                    10.20.20.200 . 456 counter packets 3 bytes 180 timeout 1m expires 9s184ms,  
                    10.20.20.200 . 234 counter packets 4 bytes 240 timeout 1m expires 4s500ms }  
    }  
  
    chain debug {  
        type filter hook prerouting priority raw - 1; policy accept;  
        meta nfttrace set 1  
    }  
  
    chain input {  
        type filter hook input priority filter - 10; policy drop;  
        iifname "lo" return  
        tcp dport 123 add @candidates_ipv4 { ip saddr . 234 timeout 1m }  
        tcp dport 234 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 { ip saddr . 789 timeout 1m }  
        tcp dport 789 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 { ip saddr . 345 timeout 1m }  
        tcp dport 345 ip saddr . tcp dport @candidates_ipv4 add @candidates_ipv4 { ip saddr . 456 timeout 1m }  
        tcp dport 456 ip saddr . tcp dport @candidates_ipv4 add @clients_ipv4 { ip saddr timeout 1m } log prefix "Successful portknock: "  
        tcp dport 22 ip saddr @clients_ipv4 counter packets 14 bytes 3700 accept  
        tcp dport 22 ct state established,related counter packets 115 bytes 7832 accept  
        tcp dport 22 counter packets 0 bytes 0 drop  
    }  
}
```

Configuração da firewall para autorizar comunicações diretas (sem NAT)

“A configuração da firewall deve eliminar todas as comunicações entre redes, exceto as necessárias para o funcionamento normal dos seguintes serviços:”

O DNS como é um serviço que utiliza pacotes UDP por defeito (exceto em algumas exceções), e a ferramenta do “Netcat”, por defeito usar pacotes TCP, para executar os dois testes seguintes, de forma mais aproximada à realidade (nslookup) foi necessário adicionar flag “-u” ao comando do netcat forçando assim o uso de UDP.

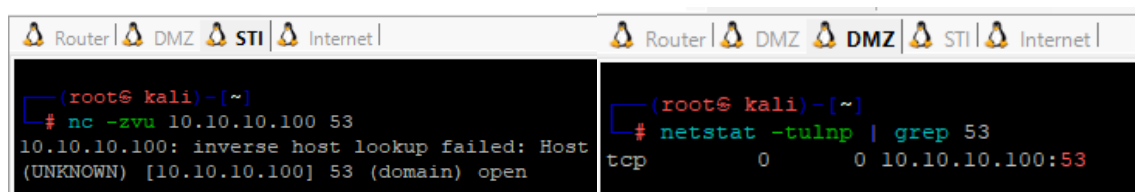
No teste do serviço “openvpn” também foi necessário proceder à adição da flag “-u”, para este comando usar o protocolo UDP, visto que este serviço por norma utiliza pacotes UDP.

● Resoluções de nomes de domínio utilizando o servidor dns.

Linha de código NfTable:

```
Chain forward { tcp dport 53 ip daddr 10.10.10.100 counter accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)~  
# nc -zvu 10.10.10.100 53  
10.10.10.100: inverse host lookup failed: Host  
(UNKNOWN) [10.10.10.100] 53 (domain) open  
  
(root@kali)~  
# netstat -tulnp | grep 53  
tcp        0      0 10.10.10.100:53
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 2 bytes 58 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
}
```

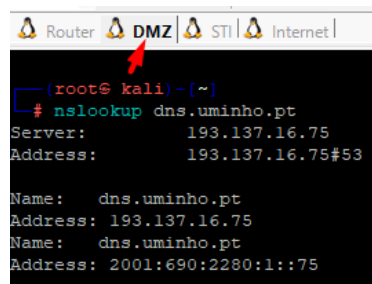
- O servidor dns deve resolver nomes utilizando servidores DNS na Internet (dns2 e 1.1.1.1).

Utilizando servidor DNS2:

Linha de código NfTable:

```
Chain forward { udp dport 53 ip daddr 193.137.16.75 counter accept }
```

#Execução do comando para fazer a resolução de nome.



```
(root@kali)~# nslookup dns.uminho.pt
Server:      193.137.16.75
Address:     193.137.16.75#53

Name:   dns.uminho.pt
Address: 193.137.16.75
Name:   dns.uminho.pt
Address: 2001:690:2280:1::75
```

#Correspondência dos pacotes de rede com as regras nftables.

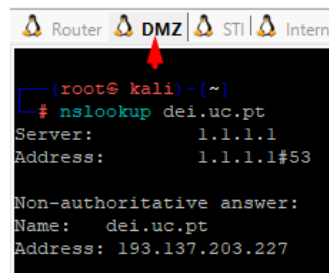
```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 2 bytes 162 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 2 bytes 118 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
}
```

Utilizando servidor 1.1.1.1:

Linha de código NfTable:

```
Chain forward { udp dport 53 ip daddr 1.1.1.1 counter accept }
```

#Execução do comando para fazer a resolução de nome.



```
(root@kali)~# nslookup dei.uc.pt
Server:      1.1.1.1
Address:     1.1.1.1#53

Non-authoritative answer:
Name:   dei.uc.pt
Address: 193.137.203.227
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 2 bytes 171 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 2 bytes 110 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
}
```

- Os servidores dns e dns2 devem ser capazes de sincronizar o conteúdo das zonas DNS.

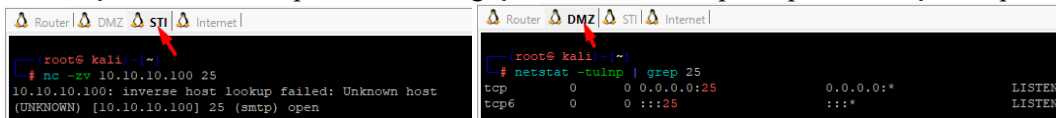
Configuramos ambos os serviços em diferentes servidores, sendo um em modo de Master e outro em modo de Slave, no qual não obtivemos sucesso na sincronização, paralelamente executou-se os comandos “nc 10.10.10.100 53” a partir do servidor DNS2 e o inverso, sem sucesso. Tentou-se também fazer de uma máquina “nc -l 53” e noutra “nc 10.10.10.100 53”, mas sem sucesso.

- Ligações SMTP ao servidor smtp.

Linha de código NfTable:

```
Chain forward { tcp dport 25 ip daddr 10.10.10.100 counter accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)~# nc -zv 10.10.10.100 25
10.10.10.100: inverse host lookup failed: Unknown host
(UNKNOWN) [10.10.10.100] 25 (smtp) open
```

```
(root@kali)~# netstat -tulnp | grep 25
tcp        0      0 0.0.0.0:25          0.0.0.0:*           LISTEN
tcp6       0      0 :::25              :::*                 LISTEN
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 5 bytes 287 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 192.168.220.254 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 2 bytes 142 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 1 bytes 60 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 99, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
}
```

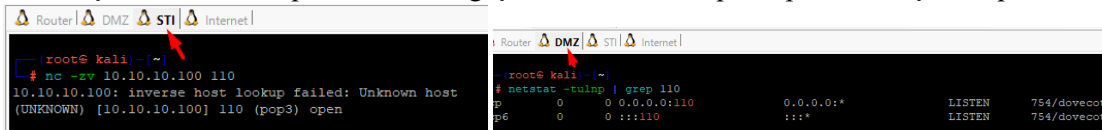
- Ligações POP e IMAP ao servidor de correio eletrónico.

Ligação POP:

Linha de código NfTable:

```
Chain forward { tcp dport { 110, 143 } ip daddr 10.10.10.100 counter accept }
```


#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)~# nc -zv 10.10.10.100 110
10.10.10.100: inverse host lookup failed: Unknown host
(UNKNOWN) [10.10.10.100] 110 (pop3) open

Router DMZ STI Internet
# netstat -tulnp | grep 110
tcp        0      0 0.0.0.0:110          0.0.0.0:*           LISTEN     754/dovecot
tcp6       0      0 :::110              :::*                 LISTEN     754/dovecot
```

#Correspondência dos pacotes de rede com as regras nftables.

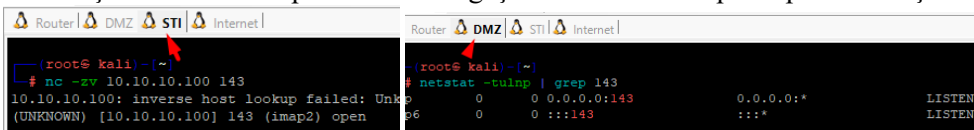
```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 9 bytes 500 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 1 bytes 71 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 1 bytes 60 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
}
```

Ligação IMAP:

Linha de código NfTable:

Chain forward { tcp dport { 110, 143 } ip daddr 10.10.10.100 counter accept }

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)~# nc -zv 10.10.10.100 143
10.10.10.100: inverse host lookup failed: Unknown host
(UNKNOWN) [10.10.10.100] 143 (imap2) open

Router DMZ STI Internet
# netstat -tulnp | grep 143
tcp        0      0 0.0.0.0:143         0.0.0.0:*           LISTEN     36
tcp6       0      0 :::143             :::*                 LISTEN     36
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 7 bytes 391 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 1 bytes 71 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 1 bytes 60 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport { 1154 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
}
```

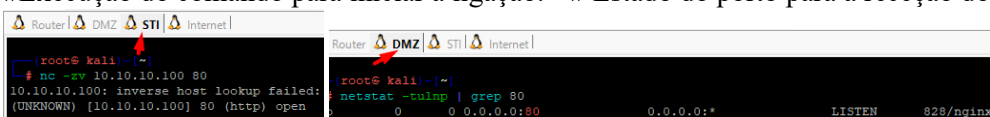
● Ligações HTTP e HTTPS ao servidor www.

Ligação HTTP:

Linha de código NfTable:

Chain forward { tcp dport { 80, 443 } ip daddr 10.10.10.100 accept }

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



```
(root@kali)~# nc -zv 10.10.10.100 80
10.10.10.100: inverse host lookup failed: Unknown host
(UNKNOWN) [10.10.10.100] 80 (http) open

Router DMZ STI Internet
# netstat -tulnp | grep 80
tcp        0      0 0.0.0.0:80          0.0.0.0:*           LISTEN     828/nginx:
tcp6       0      0 :::80              :::*                 LISTEN     828/nginx:
```

#Correspondência dos pacotes de rede com as regras nftables.

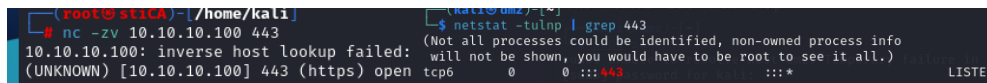
```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 6 bytes 339 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 1 bytes 71 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 1 bytes 60 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 2022 counter packets 0 bytes 0 accept
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept
}
```

Ligação HTTPS:

Linha de código NfTable:

```
Chain forward { tcp dport { 80, 443 } ip daddr 10.10.10.100 accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



#Correspondência dos pacotes de rede com as regras nftables.

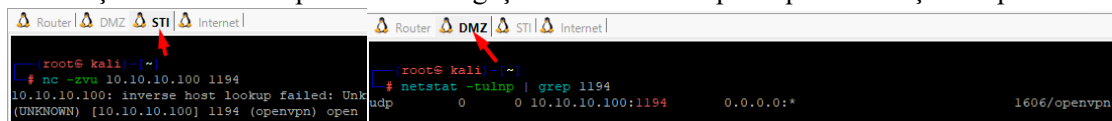
```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 6 bytes 339 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 192.168.220.254 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 1 bytes 71 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 1 bytes 60 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
}
```

● Ligações OpenVPN ao servidor vpn-gw.

Linha de código NfTable:

```
Chain forward { udp dport 1194 ip daddr 10.10.10.100 counter accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 1 bytes 71 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 1 bytes 71 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 a
    udp dport 1194 ip daddr 10.10.10.100 counter packets 2 bytes 58 accept
    ip protocol icmp counter packets 0 bytes 0 accept
}
```

Configuração da firewall para ligações ao endereço IP externo da firewall (utilizando NAT)

“As conexões originadas no exterior (Internet) e destinadas ao endereço IP externo (identificado como VMWare IP) do firewall devem ser autorizadas e tratadas de acordo com os seguintes requisitos:”

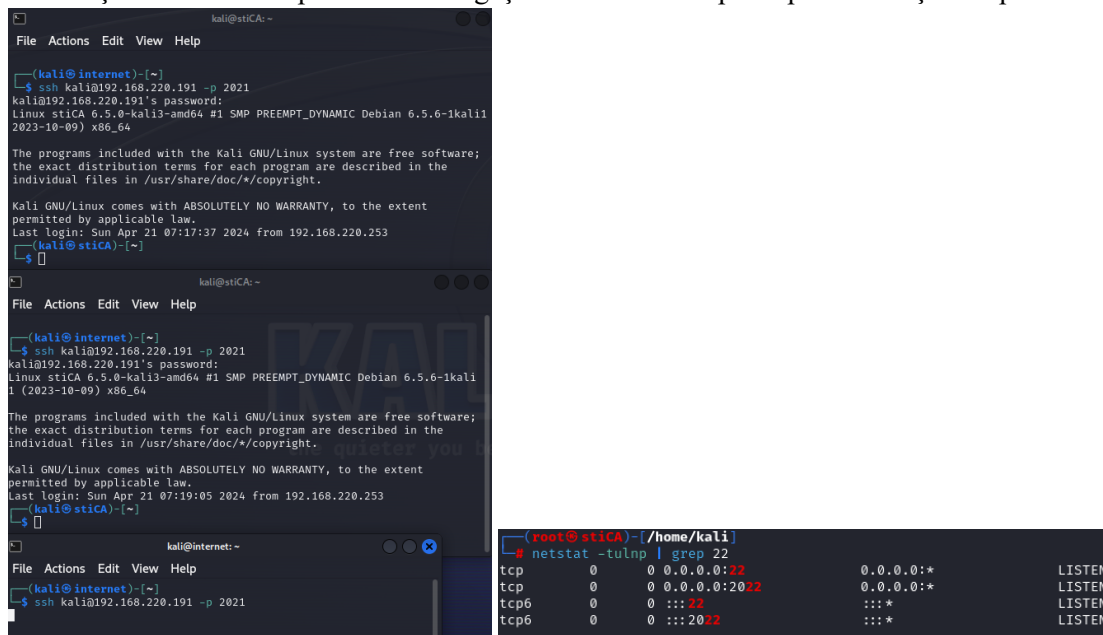
- As ligações SSH para a porta 2021 do router Linux devem ser redirecionadas para a porta SSH do servidor de armazenamento de dados e devem ser limitadas a 2 ligações simultâneas.

Linha de código NfTable:

```
Chain forward {ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 22 ct count 2 accept}
```

```
Chain prerouting { ip saddr 192.168.220.253 tcp dport 2021 counter accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.



The image contains three terminal screenshots. The top two show SSH sessions from a Kali machine (kali@kali) to a target machine (kali@192.168.220.191) on port 2021. The bottom screenshot shows a netstat command output on a Kali machine, displaying listening ports for TCP and TCP6, with port 2021 highlighted in red.

```
(kali@kali)~$ ssh kali@192.168.220.191 -p 2021
kali@192.168.220.191's password:
Linux stiCA 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1
2023-10-09) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 21 07:17:37 2024 from 192.168.220.253
(kali@stiCA)~$

(kali@kali)~$ ssh kali@192.168.220.191 -p 2021
kali@192.168.220.191's password:
Linux stiCA 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1
1 (2023-10-09) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 21 07:19:05 2024 from 192.168.220.253
(kali@stiCA)~$

(kali@kali)~$ netstat -tulnp | grep 22
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN
tcp        0      0 0.0.0.0:2021       0.0.0.0:*           LISTEN
tcp6       0      0 :::22              :::*                LISTEN
tcp6       0      0 :::2021            :::*                LISTEN
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 84 bytes 20992 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 192.168.220.254 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 0 bytes 0 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 2022 counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 2 bytes 120 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept
}
table ip nat {
    chain prerouting {
        type nat hook prerouting priority filter; policy accept;
        ip saddr 192.168.220.253 tcp dport 2021 counter packets 12 bytes 720 dnat to 10.20.20.200:22
        ip saddr 192.168.220.253 tcp dport 2022 counter packets 0 bytes 0 dnat to 10.20.20.200:2022
    }
}
```

- As ligações para a porta 2022 do Router Linux devem ser redirecionadas para a porta 2022 do servidor STI CA e só devem ser permitidas a partir de um único endereço IP da Internet (este endereço deve ser documentado no relatório).

Para permitir acesso SSH ao servidor através de 2 portas, foi preciso aceder ao servidor e alterar o seu ficheiro “/etc/ssh/sshd_config” e adicionar a linha “Port 2022”.

Linha de código NfTable:

```
Chain forward {ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 2022 ct count 2 accept}
```

```
Chain prerouting { ip saddr 192.168.220.253 tcp dport 2022 counter accept }
```

#Execução do comando para iniciar a ligação. # Estado do porto para a receção do pedido.

```
(kali@internet)-[~]
$ ssh kali@192.168.220.191 -p 2022
kali@192.168.220.191's password:
Linux stiCA 6.5.0-kali3-and64 #1 SMP PREEMPT_DYNAMIC Debian
2023-10-09) x86_64

The programs included with the Kali GNU/Linux system are free
the exact distribution terms for each program are described in
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Apr 21 07:19:16 2024 from 192.168.220.253
kali@stiCA)-[~]
$
```

```
(root@stiCA)-[/home/kali]
# netstat -tulnp | grep 2022
tcp        0      0 0.0.0.0:2022        0.0.0.0:*          LISTEN
tcp6       0      0 :::2022            :::*                LISTEN
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 243 bytes 28484 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 192.168.220.254 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 0 bytes 0 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 2022 counter packets 1 bytes 60 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept
}
table ip nat {
    chain prerouting {
        type nat hook prerouting priority filter; policy accept;
        ip saddr 192.168.220.253 tcp dport 2021 counter packets 0 bytes 0 dnat to 10.20.20.200:22
        ip saddr 192.168.220.253 tcp dport 2022 counter packets 1 bytes 60 dnat to 10.20.20.200:2022
    }
}
```

Configuração da firewall para comunicações da rede interna para o exterior (utilizando NAT)

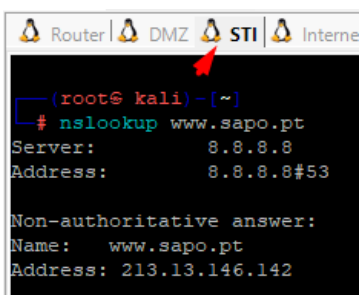
“As seguintes comunicações da rede interna para o exterior (Internet) devem ser autorizadas utilizando NAT:”

- **Resoluções de nomes de domínio utilizando DNS.**

Linha de código NfTable:

```
Chain forward { udp dport 53 ip daddr 8.8.8.8 counter accept }
```

#Execução do comando para fazer a resolução de nome.



```
(root@ kali) -[~]
# nslookup www.sapo.pt
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
Name:   www.sapo.pt
Address: 213.13.146.142
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 8 bytes 694 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 5 bytes 301 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
```

- **Ligações HTTP, HTTPS e SSH a partir de dispositivos com endereço IP dinâmico (clientes DHCP).**

Para os testes HTTP e HTTPS, as regras são as mesmas e os outputs do “nft list ruleset” são os mesmos.

Ligação HTTP:

Linhas de código NfTable:

```
Chain forward{ ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter}
```

```
Chain postrouting { ip saddr 10.20.20.0/24 oifname "eth2" counter accept }
```

#Execução do comando para fazer download da página web.

```
Router| DMZ | STI | Internet |
--(root@ kali)-[~]
# wget http://www.sapo.pt
--2024-04-21 04:54:30-- http://www.sapo.pt/
Resolving www.sapo.pt (www.sapo.pt)... 213.13.146.142
Connecting to www.sapo.pt (www.sapo.pt)|213.13.146.142|:80... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://www.sapo.pt/ [following]
--2024-04-21 04:54:30-- https://www.sapo.pt/
Connecting to www.sapo.pt (www.sapo.pt)|213.13.146.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 641481 (626K) [text/html]
Saving to: 'index.html'

index.html                               100%[=====]
2024-04-21 04:54:30 (4.24 MB/s) - 'index.html' saved [641481/641481]
```

#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    ct state established,related counter packets 511 bytes 669740 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 2 bytes 114 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 2022 counter packets 0 bytes 0 accept
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 2 bytes 120 accept
}

table ip nat {
    chain prerouting {
        type nat hook prerouting priority filter; policy accept;
        ip saddr 10.0.4.15 tcp dport 2021 counter packets 0 bytes 0 dnat to 10.20.20.200:22
        ip saddr 10.0.4.15 tcp dport 2022 counter packets 0 bytes 0 dnat to 10.20.20.200:2022
    }

    chain postrouting {
        type nat hook postrouting priority srcnat; policy accept;
        ip saddr 10.20.20.0/24 oifname "eth2" counter packets 3 bytes 177 masquerade
    }
}
```

Ligação HTTPS:

#Execução do comando para fazer download da página web.

```
Router| DMZ | STI | Internet |
--(root@ kali)-[~]
# wget https://www.sapo.pt
--2024-04-21 04:59:00-- https://www.sapo.pt/
Resolving www.sapo.pt (www.sapo.pt)... 213.13.146.142
Connecting to www.sapo.pt (www.sapo.pt)|213.13.146.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 641333 (626K) [text/html]
Saving to: 'index.html.1'

index.html.1                             100%[=====]
2024-04-21 04:59:00 (5.10 MB/s) - 'index.html.1' saved [641333/641333]
```

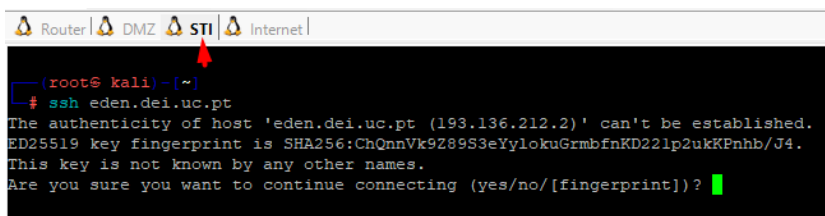
Ligação SSH:

Linha de código NfTable:

```
Chain forward { ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter accept }
```

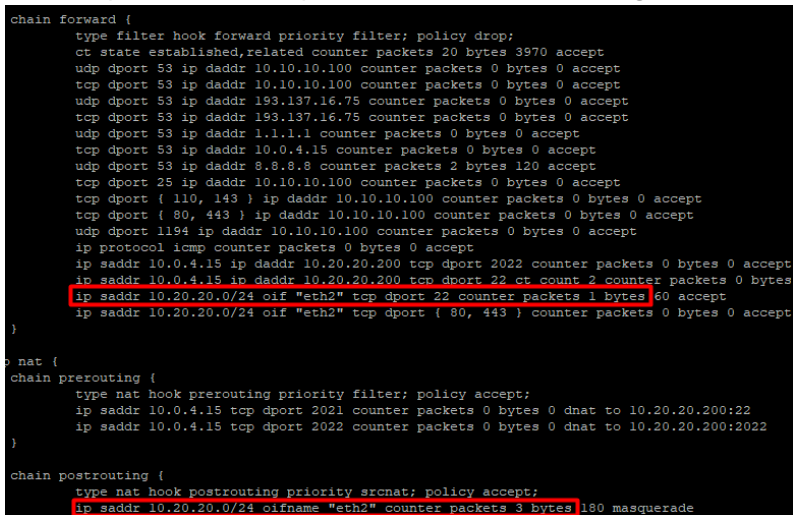
```
Chain postrouting { ip saddr 10.20.20.0/24 oifname "eth2" counter accept }
```

#Execução do comando para fazer ligação SSH.



```
(root@kali)~  
# ssh eden.dei.uc.pt  
The authenticity of host 'eden.dei.uc.pt (193.136.212.2)' can't be established.  
ED25519 key fingerprint is SHA256:ChQnnVk9Z89S3eYylokuGrmbfnKD22lp2ukKPnhb/J4.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

#Correspondência dos pacotes de rede com as regras nftables.



```
chain forward {  
    type filter hook forward priority filter; policy drop;  
    ct state established,related counter packets 20 bytes 3970 accept  
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept  
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept  
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept  
    tcp dport 53 ip daddr 10.0.4.15 counter packets 0 bytes 0 accept  
    udp dport 53 ip daddr 8.8.8.8 counter packets 2 bytes 120 accept  
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept  
    ip protocol icmp counter packets 0 bytes 0 accept  
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 2022 counter packets 0 bytes 0 accept  
    ip saddr 10.0.4.15 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept  
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 1 bytes 60 accept  
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept  
}  
  
nat {  
    chain prerouting {  
        type nat hook prerouting priority filter; policy accept;  
        ip saddr 10.0.4.15 tcp dport 2021 counter packets 0 bytes 0 dnat to 10.20.20.200:22  
        ip saddr 10.0.4.15 tcp dport 2022 counter packets 0 bytes 0 dnat to 10.20.20.200:2022  
    }  
  
    chain postrouting {  
        type nat hook postrouting priority srcnat; policy accept;  
        ip saddr 10.20.20.0/24 oifname "eth2" counter packets 3 bytes 180 masquerade  
    }  
}
```

Deteção e prevenção de intrusões (IDS/IPS)

Detetar e bloquear (pelo menos) os seguintes ataques

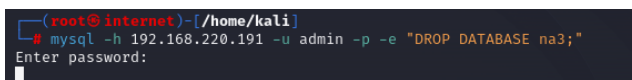
Um tipo de injeção SQL:

Para realizar este ataque SQL foi necessário instalar o serviço “mariadb” no qual procedeu-se à criação de base dados de teste. Foram criadas 2 regras no nftable, bem como adicionadas 2 regras no ficheiro “/etc/suricata/suricata.rules”.

Esta experiência teve como objetivo apagar a base de dados “na3”, no qual o Suricata cumpriu com a sua função e ao ler a regra de “drop” para este efeito bloqueou de imediato.

Comando executado:

mysql -h 10.10.10.100 -u admin -p -e "DROP DATABASE na3;"



```
(root@Internet)~  
# mysql -h 192.168.220.191 -u admin -p -e "DROP DATABASE na3;"  
Enter password:  
█
```


#Correspondência dos pacotes de rede com as regras nftables.

```
chain forward {
    type filter hook forward priority filter; policy drop;
    queue to 0
    ct state established,related counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 193.137.16.75 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 1.1.1.1 counter packets 0 bytes 0 accept
    tcp dport 53 ip daddr 192.168.220.253 counter packets 0 bytes 0 accept
    udp dport 53 ip daddr 8.8.8.8 counter packets 0 bytes 0 accept
    tcp dport 25 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 110, 143 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    tcp dport { 80, 443 } ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    udp dport 1194 ip daddr 10.10.10.100 counter packets 0 bytes 0 accept
    ip protocol icmp counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 2022 counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.20.20.200 tcp dport 22 ct count 2 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport 22 counter packets 0 bytes 0 accept
    ip saddr 10.20.20.0/24 oif "eth2" tcp dport { 80, 443 } counter packets 0 bytes 0 accept
    ip saddr 192.168.220.253 ip daddr 10.10.10.100 tcp dport 3306 counter packets 0 bytes 0 accept
}

table ip nat {
    chain prerouting {
        type nat hook prerouting priority filter; policy accept;
        ip saddr 192.168.220.253 tcp dport 2021 counter packets 0 bytes 0 dnat to 10.20.20.200:22
        ip saddr 192.168.220.253 tcp dport 2022 counter packets 0 bytes 0 dnat to 10.20.20.200:2022
        ip saddr 192.168.220.253 tcp dport 3306 counter packets 9 bytes 540 dnat to 10.10.10.100:3306
    }
}
```

No seguinte ficheiro “/var/log/suricata/fast.log” é possível ver o drop e o aviso dos pacotes a serem bloqueados.

```
(root@router1)-[/var/log/suricata]
# cat fast.log
04/21/2024-14:44:32.147081 [Drop] [**] [1:1000011:1] Possível comando DROP DATABASE SQL [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:34394 → 10.10.10.100:3306
04/21/2024-14:44:32.147081 [**] [1:1000001:1] Possível comando DROP DATABASE SQL [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:34394 → 10.10.10.100:3306
```

Dois tipos de ataques DoS:

1º Ataque:

Para este tipo de ataque foi necessário recorrer à ferramenta “hping3”, no qual vai enviar pacotes para o router do tipo “TCP-SYN”, com a opção “--flood” vai enviar o mais rápido possível até este deixar de responder.

Comando executado:

hping3 -S --flood 192.168.220.191

```
(root@internet)-[/home/kali]
# hping3 -S --flood 192.168.220.191
HPING 192.168.220.191 (eth0 192.168.220.191): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

No seguinte ficheiro “/var/log/suricata/fast.log” é possível ver o drop e o aviso dos pacotes a serem bloqueados.

```
04/21/2024-14:53:36.500519 [**] [1:1100001:1] DoS TCP Attack Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.220.253:64941 → 192.168.220.191:0
04/21/2024-14:53:36.520022 [Drop] [**] [1:1100011:1] DoS TCP Attack Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.220.253:65296 → 192.168.220.191:0
04/21/2024-14:53:36.520022 [**] [1:1100001:1] DoS TCP Attack Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.220.253:65296 → 192.168.220.191:0
```

2º Ataque:

Conforme anteriormente visto voltou-se a usar a ferramenta “hping3”, no qual vai enviar pacotes para o router do tipo “ICMP”, com a opção “--flood” enviando assim o mais rápido possível, até este deixar de responder.

Comando executado:

hping3 --icmp --flood 192.168.220.191


```
(root@internet)-[/home/kali]
# hping3 --icmp --flood 192.168.220.191
HPING 192.168.220.191 (eth0 192.168.220.191): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
-- 192.168.220.191 hping statistic --
1745850 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
(root@internet)-[/home/kali]
#
```

No seguinte ficheiro “/var/log/suricata/fast.log” é possível ver o drop e o aviso dos pacotes a serem bloqueados.

```
04/21/2024-15:04:57.189537 [**] [1:1100001:1] DoS ICMP Attack Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.220.253:8 → 192.168.220.191:0
04/21/2024-15:04:57.617777 [Drop] [**] [1:1100010:1] DoS ICMP Attack Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.220.253:8 → 192.168.220.191:0
04/21/2024-15:04:57.617777 [**] [1:1100001:1] DoS ICMP Attack Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.220.253:8 → 192.168.220.191:0
04/21/2024-15:04:58.191465 [wDrop] [**] [1:1100010:1] DoS ICMP Attack Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.220.253:8 → 192.168.220.191:0
04/21/2024-15:04:58.191465 [**] [1:1100001:1] DoS ICMP Attack Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.220.253:8 → 192.168.220.191:0
```

Um tipo de ataque de força bruta:

Para este ataque foi usada a ferramenta “hydra” que é muito popular nos dias de hoje, para fazer ataques de força bruta.

No comando usou-se os seguintes atributos / flags:

“-l kali “ de forma a identificar o utilizador alvo do ataque;

“-P /usr/share/wordlists/rockyou.txt” dizer qual é o dicionário que ele vai usar;

“192.168.220.191” Endereço IP alvo do ataque;

“-s 2021 ssh “ é o porto e o serviço a ser explorados.

Comando executado:

```
hydra -l kali -P /usr/share/wordlists/rockyou.txt 192.168.220.191 -s 2021 ssh
```

```
(root@internet)-[/home/kali]
# hydra -l kali -P /usr/share/wordlists/rockyou.txt 192.168.220.191 -s 2021 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-21 15:03:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.220.191:2021/
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-21 15:04:06
```

No seguinte ficheiro “/var/log/suricata/fast.log” é possível ver o drop e o aviso dos pacotes a serem bloqueados.

```
04/21/2024-15:03:34.493534 [Drop] [**] [1:1110011:1] Possivel ataque de força bruta SSH [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:52732 → 10.20.20.200:22
04/21/2024-15:03:34.493534 [**] [1:1110001:1] Possivel ataque de força bruta SSH [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:52732 → 10.20.20.200:22
04/21/2024-15:03:34.493540 [Drop] [**] [1:1110011:1] Possivel ataque de força bruta SSH [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:52740 → 10.20.20.200:22
04/21/2024-15:03:34.493540 [**] [1:1110001:1] Possivel ataque de força bruta SSH [**] [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 192.168.220.253:52740 → 10.20.20.200:22
```

Audit any modification in the firewall and IDS/IPS configuration files

“Utilize os recursos de auditoria do Linux para encontrar qualquer modificação nos ficheiros de segurança da firewall e do IDS/IPS. Qualquer tentativa de modificar as permissões e qualquer tentativa de ler estes ficheiros de configuração deve ser auditada.”

apt-get install auditd

systemctl enable auditd

systemctl restart auditd

auditctl -w /etc/nftables.conf -p wa -k nftables_changes

auditctl -w /etc/suricata/rules/suricata.rules -p wa -k suricata_rules_changes

Auditoria ao ficheiro de configuração das regras do NFtables:

Comando para realizar a auditoria ao ficheiro:

auresearch -k nftables_changes

```
root@kali: ~#
# auresearch -k nftables_changes
---
time-->Thu Apr 18 23:14:06 2024
type=PROCTITLE msg=audit(1713478446.834:3): proctitle=2F7362696E2F617564697463746C002D52002F6574632F61756469742E72756C6573
type=SOCKADDR msg=audit(1713478446.834:3): saddr=10000000000000000000000000000000
type=SYSCALL msg=audit(1713478446.834:3): arch=c000003e syscall=44 success=yes exit=1092 a0=3 a1=7ffe7301d680 a2=444 a3=0 items=0 ppid=1506 pid=1517 auid=4294967295
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1713478446.834:3): auid=4294967295 ses=4294967295 subj=unconfined op=remove_rule key="nftables_changes" list=4 res=1
---
time-->Thu Apr 18 23:14:58 2024
type=PROCTITLE msg=audit(1713478498.760:92): proctitle=617564697463746C002D77002F6574632F6E667461626C65732E636F6E66002D770007761002D6B006E667461626C65735F6368616E676
3
type=SYSCALL msg=audit(1713478498.760:92): arch=c000003e syscall=44 success=yes exit=1092 a0=4 a1=7ffde824fcb0 a2=444 a3=0 items=0 ppid=934 pid=1825 auid=0 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=2 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1713478498.760:92): auid=0 ses=2 subj=unconfined op=add_rule key="nftables_changes" list=4 res=1
---
time-->Thu Apr 18 23:15:10 2024
type=PROCTITLE msg=audit(1713478510.230:100): proctitle=6E616E6F002F6574632F6E667461626C65732E636F6E66
type=PATH msg=audit(1713478510.230:100): item=1 name="/etc/nftables.conf" inode=4849888 dev=08:01 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap
fd=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1713478510.230:100): item=0 name="/etc/" inode=4849665 dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0 cap_fe=0
cap_fver=0 cap_frootid=0
type=CWD msg=audit(1713478510.230:100): cwd="/root"
type=SYSCALL msg=audit(1713478510.230:100): arch=c000003e syscall=257 success=yes exit=3 a0=ffffff9c a1=5573c4975450 a2=241 a3=1b6 items=2 ppid=934 pid=1835 auid=0
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=2 comm="nano" exe="/usr/bin/nano" subj=unconfined key="nftables_changes"
```

Auditoria ao ficheiro de configuração das regras do Suricata:

Comando para realizar a auditoria ao ficheiro:

auresearch -k suricata_rules_changes

```
root@kali: ~#
# nano /etc/suricata/rules/suricata.rules
---
root@kali: ~#
# auresearch -k suricata_rules_changes
---
time-->Sun Apr 21 05:24:02 2024
type=PROCTITLE msg=audit(1713673442.538:449): proctitle=617564697463746C002D77002F6574632F73757269636174612F72756C65732F73757269636174612E72756C6573002D
type=SYSCALL msg=audit(1713673442.538:449): arch=c000003e syscall=44 success=yes exit=1112 a0=4 a1=7ffde0d218a0 a2=458 a3=0 items=0 ppid=1039 pid=3025 auid=0 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=7 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1713673442.538:449): auid=0 ses=7 subj=unconfined op=add_rule key="suricata_rules_changes" list=4 res=1
---
time-->Sun Apr 21 05:24:42 2024
type=PROCTITLE msg=audit(1713673482.870:450): proctitle=6E616E6F002F6574632F73757269636174612F72756C65732F73757269636174612E72756C6573
type=PATH msg=audit(1713673482.870:450): item=1 name="/etc/suricata/rules/suricata.rules" inode=4853006 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00
uid=0
type=PATH msg=audit(1713673482.870:450): item=0 name="/etc/suricata/rules/" inode=4851865 dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT
type=CWD msg=audit(1713673482.870:450): cwd="/root"
type=SYSCALL msg=audit(1713673482.870:450): arch=c000003e syscall=257 success=yes exit=3 a0=ffffff9c a1=5574bf9ae1e0 a2=241 a3=1b6 items=2 ppid=1039 pid=3025 auid=0
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=7 comm="nano" exe="/usr/bin/nano" subj=unconfined key="suricata_rules_changes"
```

Conclusão

Este trabalho foi interessante, pois permitiu implementar um cenário semelhante à realidade, apesar de ter sido concentrado vários tipos de serviços numa única máquina apenas, neste caso a VM DMZ.

Foi possível aprofundar ainda mais os conhecimentos de NFT, bem como configuração de outros serviços pedidos no trabalho (smtp,dns,imap,pop...).

Como nos dias de hoje é quase imperativo ter um IDS /IPS, foi também implementado o Suricata, de forma a bloquear certas tentativas de ataque, conseguindo assim ganhar um maior à vontade com esta ferramenta.

Referências

- [1] https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
- [2] <https://docs.suricata.io/en/latest/quickstart.html>
- [3] <https://www.stamus-networks.com/suricata-rules>
- [4] Laboratórios da prática de STI