UNIVERSIDADE D
COIMBRA

Relatório do Assignment 2

Segurança em Tecnologias da Informação (STI)

Bruno Silva 2021232021

Diogo Honório 2021232043

Introdução

O presente relatório tem como objetivo documentar a implementação de uma arquitetura de rede segmentada em múltiplas zonas, com recurso a máquinas virtuais interligadas por interfaces configuradas manualmente. A infraestrutura projetada visa simular um ambiente realista de rede empresarial, promovendo a separação lógica entre diferentes domínios de segurança: rede interna, zona desmilitarizada (DMZ) e acesso à Internet.

Através da configuração de regras de firewall (IPTables), da aplicação de NAT e da integração do sistema de deteção e prevenção de intrusões (IDS/IPS) Suricata, procurou-se garantir a segurança, o controlo e a monitorização do tráfego entre os diferentes segmentos de rede. Foram ainda realizados testes de conectividade e simulação de ataques (como SYN Flood, ICMP Flood e OS Fingerprinting), de modo a validar a eficácia das políticas de segurança implementadas.

1. Arquitetura

1.1. Configuração das Máquinas Virtuais (VM)

O cenário implementado para a resolução deste trabalho prático consiste em quatro máquinas virtuais que simulam a infraestrutura de rede pedida.

VM do Router:

Esta máquina funciona como um router, permitindo, assim, oferecer acesso entre as redes para as restantes máquinas. Esta é o ponto crucial da arquitetura, visto que é responsável por monitorizar o tráfego entre as redes. Para além das funcionalidades de segurança aplicadas como o NAT, regras de filtragem de pacotes (IPTables), entre outros.

É igualmente nesta máquina que está integrado o sistema de deteção e prevenção de intrusões (Suricata), que tem como função monitorizar o tráfego e reagir automaticamente a tentativas de ataques, bloqueando-as em tempo útil.

O seguinte comando foi usado de forma que esta máquina funcione como um router:

```
(root@Router) - [/] # sudo sysctl -w net.ipv4.ip_forward=1
```

VM do DMZ:

Esta máquina representa a zona desmilitarizada (DMZ) da arquitetura, sendo o local onde residem os serviços públicos da organização, acessíveis a partir da Internet. Nesta rede encontram-se os servidores de *web*, *mail*, *DNS*, *vpn* e *smtp*.

A comunicação com o exterior é permitida apenas através das portas e protocolos estritamente necessários, conforme definido nas regras de firewall configuradas na VM do Router.

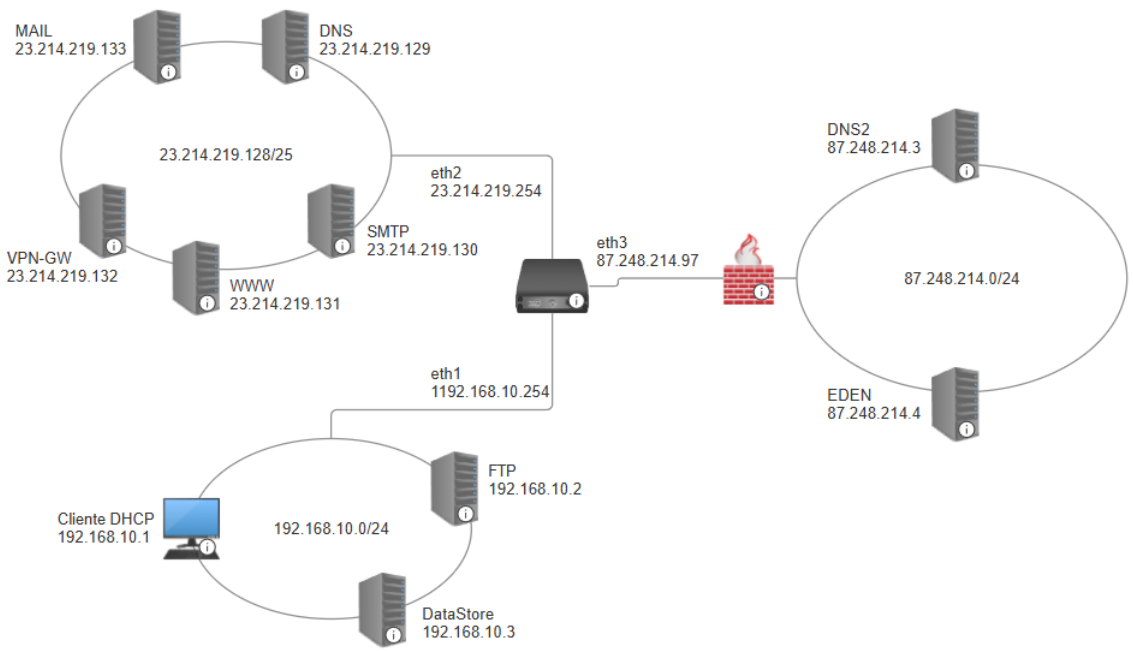
VM da Rede Interna:

A VM da rede interna representa o ambiente privado da organização, reservado a utilizadores e sistemas internos. É nesta rede que se localizam os clientes, bem como servidores internos específicos, como o servidor de *datastore*.

VM da Internet (Rede externa):

Esta máquina simula a presença de um utilizador externo, localizado fora dos domínios da rede da organização, representando o tráfego proveniente da Internet. A sua principal função é a de servir como origem para testes de conectividade e tentativas de ataque, permitindo validar a eficácia das regras de firewall e dos mecanismos de deteção e bloqueio configurados com o Suricata.

O diagrama abaixo oferece uma melhor representação da infraestrutura da rede e distribuição dos servidores:



1.2. Configurações das Redes

A topologia da rede foi organizada de forma a representar três zonas distintas: rede interna (clientes e serviços internos), rede DMZ (serviços públicos expostos à Internet) e rede externa (simulando o acesso externo). As configurações IP das interfaces foram atribuídas manualmente a cada VM, respeitando os intervalos de endereçamento definidos para cada zona:

Máquina Virtual do Router:

- **Eth1:** Interface ligada à rede interna
  - IP: 192.168.10.254/24
- **Eth2:** Interface ligada à rede DMZ
  - IP: 23.214.219.254/25
- **Eth3:** Interface ligada à Internet (rede externa)
  - IP: 87.248.214.97/24

Máquina Virtual da Rede Interna:

- **Eth0:** Interface que corresponde ao cliente DHCP
  - IP: 192.168.10.1/24
- **Eth0:1:** Subinterface correspondente ao servidor FTP
  - IP: 192.168.10.2/24
- **Eth0:2:** Subinterface correspondente ao servidor DataStore
  - IP: 192.168.10.3/24

Máquina Virtual da Rede DMZ:

- **Eth0:** Interface que corresponde ao servidor DNS
  - IP: 23.214.219.129/25
- **Eth0:1:** Subinterface correspondente ao servidor SMTP
  - IP: 23.214.219.130/25
- **Eth0:2:** Subinterface correspondente ao servidor WWW
  - IP: 23.214.219.131/25
- **Eth0:3:** Subinterface correspondente ao servidor VPN-GW
  - IP: 23.214.219.132/25
- **Eth0:4:** Subinterface correspondente ao servidor MAIL
  - IP: 23.214.219.133/25

Máquina Virtual da Rede Externa:

- **Eth0:** Interface que corresponde ao servidor DNS2
  - IP: 87.248.214.3/24
- **Eth0:1:** Subinterface correspondente ao servidor EDEN
  - IP: 87.248.214.4/24

2. Configuração das IPTables

A configuração da firewall foi realizada na Máquina Virtual do Router, utilizando o sistema IPTables, com o objetivo de controlar o tráfego entre as diferentes zonas da rede.

Abaixo, apresenta-se a estrutura geral das regras aplicadas, categorizadas consoante a sua função.

2.1. Alterar a política das tabelas

De forma a adotar uma abordagem mais segura e restritiva, foi aplicada a política de negação a duas cadeias da tabela de filtragem do IPTables: INPUT, FORWARD. Esta medida garante que qualquer tráfego não explicitamente autorizado por regras definidas manualmente será automaticamente rejeitado.

```
(root@Router) - [/] # iptables -P INPUT DROP
(root@Router) - [/] # iptables -P FORWARD DROP
(root@Router) - [/] # iptables -P OUTPUT ACCEPT
```

2.2. Proteção do próprio Router

Nesta secção, são definidas as regras de firewall que visam proteger diretamente a Máquina Virtual do Router, impedindo acessos não autorizados e limitando o tráfego àquele estritamente necessário para o seu correto funcionamento.

Permitir resolução de pedidos DNS

Para permitir que o Router possa efetuar pedidos de resolução de nomes DNS, foi permitida a entrada de tráfego UDP no porto 53 nas interfaces ligadas à rede interna (eth1) e à DMZ (eth2):

```
(root@Router) - [/] # iptables -A INPUT -i eth1 -p udp –-dport 53 -j ACCEPT
(root@Router) - [/] # iptables -A INPUT -i eth2 -p udp –-dport 53 -j ACCEPT
```

Permitir resolução de SSH

De forma a permitir a administração remota do Router, foram autorizadas ligações SSH (protocolo TCP no porto 22) provenientes da rede interna e do server vpn-gw da rede DMZ, através das interfaces eth1 e do endereço 23.214.219.132, respetivamente:

```
(root@Router) - [/] # iptables -A INPUT -i eth1 -p tcp –-dport 22 -j ACCEPT
(root@Router) - [/] # iptables -A INPUT -s 23.214.219.132 -p tcp –-dport 22 -j ACCEPT
```

2.3. Regras de comunicação direta (sem NAT) entre redes internas/DMZ

As regras de comunicação direta permitem o tráfego de protocolos essenciais aos serviços da organização, estabelecendo um fluxo entre redes internas e a DMZ.

Permitir comunicações DNS entre a rede interna e o servidor DNS na DMZ.

Permite que o router encaminhe os pacotes UDP destinados ao IP 23.214.219.129 na porta 53 (DNS).

```
(root@Router) - [/] # iptables -A FORWARD -p udp -d 23.214.219.129 -–dport 53 -j ACCEPT
```

Permitir comunicações DNS entre a rede interna e o servidor DNS2 na Rede externa.

Permite que o servidor DNS com IP 23.214.219.129 envie pacotes UDP para o servidor DNS2 com IP 87.248.214.2 na porta 53.

```
(root@Router) - [/] # iptables -A FORWARD -s 23.214.219.129 -d 87.248.214.2 -p udp --dport 53 -j ACCEPT
```

Permitir ligações SMTP ao servidor de correio (SMTP).

Permite que o router encaminhe os pacotes TCP destinados ao IP 23.214.219.130 na porta 25 (SMTP).

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.130 -–dport 25 -j ACCEPT
```

Permitir HTTP/HTTPS ao servidor WWW.

Permite que o router encaminhe os pacotes TCP destinados ao IP 23.214.219.131 nas portas 80 e 443 (HTTP/ HTTPS).

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.131 -–dport 80 -j ACCEPT

(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.131 -–dport 443 -j ACCEPT
```

Permitir ligações VPN (OpenVPN) ao servidor vpn-gw.

O primeiro comando permite que o router encaminhe os pacotes TCP destinados ao IP 23.214.219.132 na porta 1194 (VPN), enquanto o segundo permite que os pacotes vindos do IP 23.214.219.132, quando são enviados para a rede interna, sofrem um SNAT de modo a ocultar o seu IP."

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.132 -–dport 1194 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 23.214.219.132 -j MASQUERADE
```

Permitir POP/IMAP ao servidor MAIL.

Permite que o router encaminhe os pacotes TCP destinados ao IP 23.214.219.133 nas portas 110 e 143 (POP/IMAP).

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.133 -–dport 110 -j ACCEPT
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 23.214.219.133 -–dport 143 -j ACCEPT
```

Permitir o envio de respostas

Permite respostas a conexões iniciadas previamente, desta foram garantem que as comunicações bidirecionais funcionam de forma correta.

```
(root@Router) - [/] # iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
(root@Router) - [/] # iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

2.4. Regras para conexões com IPs externos da firewall (com NAT).

Estas conexões permitem controlar o tráfego proveniente do exterior (Internet) com destinos aos endereços IP externos da firewall, nestes casos dos servers FTP e datastore.

O uso do NAT é essencial nestas ocasiões pois permite que os vários servers da rede interna partilhem um único endereço quando comunicam com o exterior. Neste caso será o mesmo endereço usado para que fazer conexões com os IPs externos.

Nesta parte do trabalho foi na mesma necessário colocar regras na iptable FORWARD de modo a permitir o tráfego entre a rede interna e externa. Colocando como destino o endereço da rede interna de modo que os pacotes sejam encaminhados para a mesma

Permitir ligações ao servidor FTP

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -d 192.168.10.2 --dport 21 -j ACCEPT

(root@Router) - [/] # iptables -A FORWARD -p tcp -d 192.168.10.2 --dport 20 -j ACCEPT

(root@Router) - [/] # iptables -A FORWARD -p tcp -d 192.168.10.2 --dport 1024:65535 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A PREROUTING -i eth3 -p tcp --dport 21 -j DNAT --to-destination 192.168.10.2:21

(root@Router) - [/] # iptables -t nat -A PREROUTING -i eth3 -p tcp --dport 20 -j DNAT --to-destination 192.168.10.2:20

(root@Router) - [/] # iptables -t nat -A PREROUTING -i eth3 -p tcp --dport 1024:65535 -j DNAT --to-destination 192.168.10.2
```

Permitir ligações ao servidor DataStore

```
(root@Router) - [/] # iptables -A FORWARD -p tcp -s 87.248.214.3 -d 192.168.10.3 --dport 22 -j ACCEPT

(root@Router) - [/] # iptables -A FORWARD -p tcp -s 87.248.214.4 -d 192.168.10.3 --dport 22 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A PREROUTING -p tcp -s 87.248.214.3 --dport 22 -j DNAT --to-destination 192.168.10.3:22

(root@Router) - [/] # iptables -t nat -A PREROUTING -p tcp -s 87.248.214.4 --dport 22 -j DNAT --to-destination 192.168.10.3:22
```

2.5. Regras de comunicação da rede interna para a rede exterior (com NAT)

Esta comunicação é realizada através de conexões da rede interna para a exterior, fazendo uso do NAT, de modo que os servers internos não tenham o seu IP partilhado para o exterior mantendo assim a rede protegida. As ligações que serão autorizadas ao exterior serão as seguintes: DNS, HTTP, HTTPS, SSH e FTP.

Tal como anteriormente, foi necessário colocar regras na iptable FORWARD de modo a permitir o tráfego entre as duas redes, desta vez o destino sendo a rede externa.

Permitir comunicações DNS entre a rede interna e a rede externa.

```
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p udp --dport 53 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p udp --dport 53 -j SNAT --to-source 87.248.214.97
```

Permitir comunicações HTTP e HTTPS entre a rede interna e a rede externa.

```
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp --dport 80 -j ACCEPT
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp --dport 443 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 80 -j SNAT --to-source 87.248.214.97
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 443 -j SNAT --to-source 87.248.214.97
```

Permitir comunicações SSH entre a rede interna e a rede externa.

```
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp --dport 22 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 22 -j SNAT --to-source 87.248.214.97
```

Permitir comunicações FTP entre a rede interna e a rede externa.

```
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp --dport 21 -j ACCEPT
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp --dport 20 -j ACCEPT
```

```
(root@Router) - [/] # iptables -A FORWARD -d 87.248.214.0/24 -p tcp –dport 1024:65535 -j ACCEPT
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 21 -j SNAT --to-source 87.248.214.97

(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 20 -j SNAT --to-source 87.248.214.97
```

```
(root@Router) - [/] # iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -p tcp --dport 1024:65535 -j SNAT --to-source 87.248.214.97
```

3. Prevenção de Ataques

3.1. Configuração do Suricata

Com recurso às instruções dadas pelo professor no ficheiro "PL-SuricataNotes", conseguimos implementar uma configuração inicial, com algumas alterações para melhor servir a nossa arquitetura do trabalho. Aqui estão as alterações feitas no arquivo *suricata.yaml*:

/etc/suricata/suricata.yaml

```
$HOME_NET = [192.168.10.0/24, 23.214.219.128/25, 87.248.214.97]

af-packet:

  - interface: eth3

    cluster-id: 99

    cluster-type: cluster_flow

    defrag: yes

    use-mmap: yes
```

Desta forma, foi definida a variável $HOME_NET com as redes internas da infraestrutura, bem como o endereço IP da interface do router associada à rede externa. O modo de captura de pacotes (af-packet) foi configurado para escutar na interface eth3, correspondente à ligação com a rede externa.

Para garantir que os pacotes de rede sejam inspecionados pelo Suricata antes de alcançarem os seus destinos, foi necessário implementar uma regra iptables utilizando a funcionalidade NFQUEUE, aplicada à interface eth3:

```
(root@Router) - [/] # iptables -I FORWARD -i eth3 -j NFQUEUE --queue-num 0
```

3.2. Regras do Suricata

Para permitir a prevenção de ataques, é necessário definir e implementar regras específicas no Suricata. Estas regras determinam os padrões de tráfego de rede a monitorizar e as ações a tomar perante comportamentos indesejados.

O ficheiro usado para estas regras foi o predefinido pelo suricata, suricata.rules, acedido da seguinte forma:

```
(root@Router) - [/] # nano /var/lib/suricata/rules/ suricata.rules
```

De seguida, serão analisados os ataques que foram prevenidos, bem como as respetivas regras implementadas para os mitigar.

3.3. SYN Flood – *Denial of Service* (DoS)

Um ataque SYN Flood é um tipo de ataque de negação de serviço (DoS) que explora o funcionamento do processo de estabelecimento de ligação TCP, mais concretamente o "three-way handshake", para sobrecarregar um sistema alvo.

```
drop tcp any any -> $HOME_NET any (msg:"DoS Attack (SYN FLOOD)"; flags:S;
flow:stateless; threshold: type both, track by_src, count 200, seconds 2;
sid:1000001; rev:1;)
```

Parâmetro	Função
drop	Bloqueia os pacotes
tcp	Identifica o protocolo
any any -> $HOME_NET any	Especifica o IP e porta da origem e destinatário, neste caso é usada a variável que contem os IPs definidos no ficheiro suricata.yaml
msg : "DoS Attack (SYN FLOOD)"	Mensagem que aparecerá no log quando detetada
flags: S	Filtra pacotes com a flag SYN ativa
flow:stateless	Analisa pacotes individualmente, sem verificar o estado da conexão (útil para detectar ataques sem estabelecer sessão)
type both	Gera alerta e aplica a ação (drop) quando o limite é atingido
track by_src	Conta pacotes por IP de origem
count 200, seconds 2	Aciona a regra se um IP enviar >200 pacotes em 2 segundos
sid: 1000001	ID único da regra
rev: 1	Versão da regra

Para executar o ataque neste ambiente de teste é usado o seguinte comando na máquina da rede externa:

```
(root@Internet) - [/] # hping3 -S -p 80 –flood 192.168.10.1
```

Este comando é uma ferramenta de teste de rede e ataque de negação de serviço (DoS) que envia um grande volume de pacotes TCP SYN para um alvo, neste caso o IP da rede interna, sobrecarregando-o.

3.4. ICMP Flood – *Denial of Service* (DoS)

Neste tipo de ataque, o atacante envia uma grande quantidade de pacotes ICMP (normalmente do tipo Echo Request, como no ping) a um alvo específico. O objetivo é sobrecarregar os recursos da máquina (como CPU e largura de banda) com pedidos de resposta, tornando o sistema lento ou completamente inacessível para utilizadores legítimos.

```
drop icmp any any -> $HOME_NET any (msg:"DoS Attack (ICMP)"; itype:8; threshold:
type both, track by_src, count 200, seconds 2; sid:1000002; rev:1;)
```

Parâmetro	Função
drop	Bloqueia os pacotes
icmp	Identifica o protocolo
any any -> $HOME_NET any	Especifica o IP e porta da origem e destinatário, neste caso é usada a variável que contem os IPs definidos no ficheiro suricata.yaml
msg : "DoS Attack (ICMP)"	Mensagem que aparecerá no log quando detetada
Itype:8	filtra apenas ICMP Echo Request
type both	Gera alerta e aplica a ação (drop) quando o limite é atingido
track by_src	Conta pacotes por IP de origem
count 200, seconds 2	Aciona a regra se um IP enviar >200 pacotes em 2 segundos
sid: 1000002	ID único da regra
rev: 1	Versão da regra

O comando abaixo foi utilizado na máquina da rede externa para executar o ataque ICMP no ambiente de testes:

```
(root@Internet) - [/] # hping3 –icmp –flood 192.168.10.1
```

3.5. OS Fingerprint

OS fingerprinting é uma técnica utilizada para determinar o sistema operativo e outras informações sobre uma máquina remota, com base no comportamento do protocolo TCP/IP e das respostas a pacotes específicos.

```
drop tcp any any -> $HOME_NET any (msg:"NMAP OS Attack"; flags:S; threshold: type both, track by_src, count 3, seconds 10; sid:1000005; rev:1;)
```

Parâmetro	Função
drop	Bloqueia os pacotes
tcp	Identifica o protocolo
any any -> $HOME_NET any	Especifica o IP e porta da origem e destinatário, neste caso é usada a variável que contem os IPs definidos no ficheiro suricata.yaml
msg: "NMAP OS Attack”	Mensagem que aparecerá no log quando detetada
flags: S	Filtra pacotes com a flag SYN ativa
type both	Gera alerta e aplica a ação (drop) quando o limite é atingido
track by_src	Conta pacotes por IP de origem
count 3, seconds 10	Aciona a regra se um IP enviar >3 pacotes em 100 segundos
sid: 1000005	ID único da regra
rev: 1	Versão da regra

Este comando utiliza o Nmap, uma ferramenta amplamente utilizada para varrimento de redes e deteção de serviços.

- **-O:** ativa a deteção do sistema operativo (OS detection).
- **-sV:** ativa a deteção de versões dos serviços (ex: HTTP, SSH, Apache 2.4.52).

```
(root@Internet) - [/] # nmap -O -sV 192.168.10.1
```

4. Testes e Resultados

Nesta secção mostramos todos os testes realizados no nosso trabalho:

4.1. Permitir resolução de DNS de servers exteriores

```
[admin@a23-214-219-254 ~]$ sudo nc -u -v -l 53
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.129.
hello
```

```
[admin@localhost ~]$ nc -u 23.214.219.254 53
hello
```

4.2. Permitir resolução de SSH

Foi usado o “ssh” e foi bem-sucedido.

```
┌──(root㉿admin)-[/home/adminsti]
└─# ssh adminsti@192.168.10.254
adminsti@192.168.10.254's password:
Linux admin 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-1
5) 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: Tue Apr 15 10:58:14 2025 from 23.214.219.254
```

4.3. Permitir comunicações DNS entre a rede interna e o servidor DNS na DMZ.

Foi usado o “dig” e foi bem-sucedido.

```
┌──(root㉿admin)-[/home/adminsti]
└─# dig @23.214.219.129 google.com

; <<>> DiG 9.20.7-1-Debian <<>> @23.214.219.129 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 52884
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 6dce58795d19ee47010000006802b38b72df84f7c588ee09 (good)
; EDE: 18 (Prohibited)
;; QUESTION SECTION:
;google.com.                    IN      A

;; Query time: 4 msec
;; SERVER: 23.214.219.129#53(23.214.219.129) (UDP)
;; WHEN: Fri Apr 18 16:18:20 EDT 2025
;; MSG SIZE  rcvd: 73
```

4.4. Permitir ligações SMTP ao servidor de correio (SMTP).

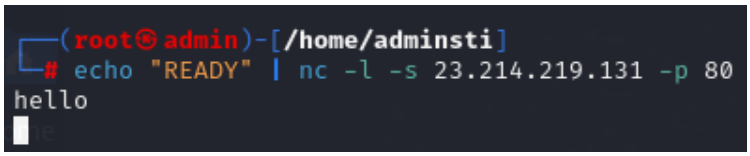
Foi usado o “nc” e foi bem-sucedido.

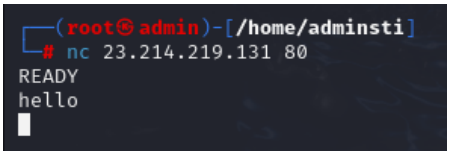
```
┌──(root㉿admin)-[/home/adminsti]
└─# echo "READY" | nc -l -s 23.214.219.130 -p 25
hello
```

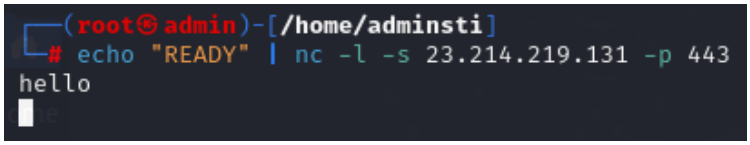
```
┌──(root㉿admin)-[/home/adminsti]
└─# nc 23.214.219.130 25
READY
hello
```

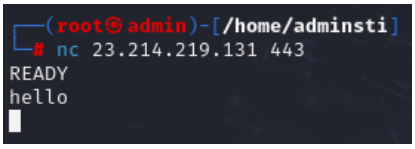
4.5. Permitir HTTP/HTTPS ao servidor WWW.

Foi usado o “nc” e foi bem-sucedido.



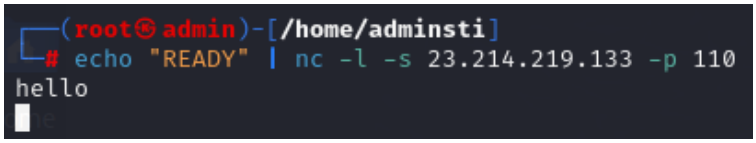


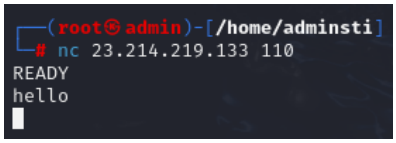


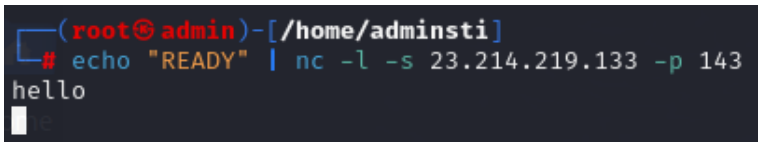


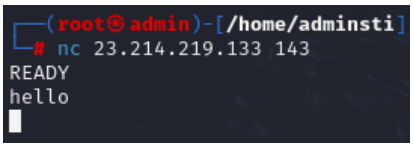
4.6. Permitir POP/IMAP ao servidor MAIL.

Foi usado o “nc” e foi bem-sucedido.



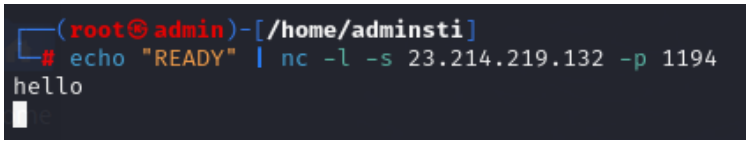


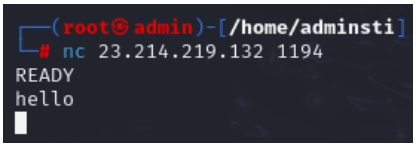




4.7. Permitir ligações VPN (OpenVPN) ao servidor vpn-gw.

Foi usado o “nc” e foi bem-sucedido.





4.8. Tabela IPTables:

```
[admin@a23-214-219-254 ~]$ sudo iptables -L -n -v
Chain INPUT (policy DROP 1221 packets, 220K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     udp  --  enp0s3 *       0.0.0.0/0            0.0.0.0/0           udp dpt:53
    0     0 ACCEPT     udp  --  enp0s8 *       0.0.0.0/0            0.0.0.0/0           udp dpt:53
    0     0 ACCEPT     tcp  --  enp0s3 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       23.214.219.132       0.0.0.0/0           tcp dpt:22

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            23.214.219.129      udp dpt:53
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.130      tcp dpt:25
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.131      tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.131      tcp dpt:443
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.132      tcp dpt:1194
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.133      tcp dpt:110
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            23.214.219.133      tcp dpt:143
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            192.168.10.2        tcp dpt:21
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            192.168.10.2        tcp dpt:20
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            192.168.10.2        tcp dpts:1024:65535
    0     0 ACCEPT     tcp  --  *      *       87.248.214.3         192.168.10.3        tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       87.248.214.4         192.168.10.3        tcp dpt:22
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            87.248.214.0/24     udp dpt:53
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpt:443
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpt:21
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpt:20
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            87.248.214.0/24     tcp dpts:1024:65535
    0     0 ACCEPT     udp  --  *      *       23.214.219.129       87.248.214.3        udp dpt:53

Chain OUTPUT (policy ACCEPT 4 packets, 268 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

4.9. Tabelas de NAT

```
[admin@a23-214-219-254 ~]$ sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 3258 packets, 492K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       tcp  --  *      *       87.248.214.3         0.0.0.0/0           tcp dpt:22 to:192.168.10.3:22
    0     0 DNAT       tcp  --  *      *       87.248.214.4         0.0.0.0/0           tcp dpt:22 to:192.168.10.3:22
    0     0 DNAT       tcp  --  enp0s9 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:21 to:192.168.10.2:21
    0     0 DNAT       tcp  --  enp0s9 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:20 to:192.168.10.2:20
    0     0 DNAT       tcp  --  enp0s9 *       0.0.0.0/0            0.0.0.0/0           tcp dpts:1024:65535 to:192.168.10.2

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 4 packets, 268 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 4 packets, 268 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       udp  --  *      *       192.168.10.0/24      0.0.0.0/0           udp dpt:53 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpt:80 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpt:443 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpt:22 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpt:21 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpt:20 to:87.248.214.97
    0     0 SNAT       tcp  --  *      *       192.168.10.0/24      0.0.0.0/0           tcp dpts:1024:65535 to:87.248.214.97
    0     0 MASQUERADE  all  --  *      *       23.214.219.132       0.0.0.0/0
```

4.10. Testes de conexões com IPs externos da firewall (SSH) - NAT:

Foi usado o “nc” e foi bem-sucedido.

```
[admin@localhost ~]$ sudo nc -v -l -p 22
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:22
Ncat: Connection from 87.248.214.3.
Ncat: Connection from 87.248.214.3:34052.
olá
tudo bem
```

```
[admin@localhost ~]$ nc 87.248.214.97 22
olá
tudo bem
```

Podemos verificar que através do DNAT conseguimos enviar uma mensagem para um IP publico neste caso o gateway da rede externa (87.248.214.97) e a mensagem é automaticamente redirecionada para o destino correto da rede interna.

4.11. Testes de conexões com IPs externos da firewall (FTP) - NAT:

Para a realização destes testes foi necessário a inicialização do server vsftpd na rede interna da seguinte forma:

```
sudo systemctl start vsftpd

sudo systemctl enable vsftpd
```

De seguida, testámos a ligação ao servidor FTP na rede externa. Depois de nos conseguirmos conectar, usámos o comando “ls” para listar os ficheiros no servidor e foi bem-sucedido.

```
[admin@localhost ~]$ ftp 87.248.214.97
Connected to 87.248.214.97 (87.248.214.97).
220 (vsFTPd 3.0.5)
Name (87.248.214.97:admin): admin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,10,2,51,248).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Documentos
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Imagens
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Modelos
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Música
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Público
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Transferências
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Vídeos
-rw-r--r--    1 1000     1000            0 Apr 17 13:42 a.out
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Área de Trabalho
226 Directory send OK.
```

Posteriormente, usámos o comando “passive” para desativar o modo passivo. Verificámos que, mesmo em modo ativo, o teste foi bem-sucedido

```
ftp> passive
Passive mode off.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Documentos
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Imagens
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Modelos
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Música
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Público
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Transferências
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Vídeos
-rw-r--r--    1 1000     1000            0 Apr 17 13:42 a.out
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Área de Trabalho
226 Directory send OK.
```

4.12. Testes de comunicação da rede interna para a rede exterior (DNS) - NAT:

```
┌──(adminsti㉿admin)-[~]
└─$ echo "READY" | nc -l -u -p 53
s
```

```
┌──(root㉿admin)-[/home/adminsti]
└─# nc -u 87.248.214.97 53
s
READY
```

4.13. Testes de comunicação da rede interna para a rede exterior (HTTP/ HTTPS) - NAT:

```
[admin@localhost ~]$ sudo nc -v -l -p 80
[sudo] senha para admin:
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 87.248.214.97.
Ncat: Connection from 87.248.214.97:58854.
ola
```

```
[admin@localhost ~]$ sudo nc -v -l -p 443
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 87.248.214.97.
Ncat: Connection from 87.248.214.97:33806.
ola
```

4.14. Testes de comunicação da rede interna para a rede exterior (SSH) - NAT:

```
[admin@localhost ~]$ sudo nc -v -l -p 22
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::22
Ncat: Listening on 0.0.0.0:22
Ncat: Connection from 87.248.214.97.
Ncat: Connection from 87.248.214.97:49136.
ola
```

Em todas estas ligações (DNS / HTTP / HTTPS / SSH), podemos observar que o IP que a rede externa recebe não corresponde ao IP da máquina da rede interna, mas sim ao endereço IP do gateway (router). Isto acontece devido à utilização de SNAT, que substitui o IP de origem por um IP público, permitindo uma comunicação segura com o exterior. Tendo isto em consideração podemos afirmar que o teste foi bem-sucedido.

4.15. Testes de comunicação da rede interna para a rede exterior (FTP) - NAT:

Os testes FTP procederam-se da mesma forma que anteriormente com a única diferença que agora o server estaria na máquina da rede externa.

Podemos observar que em modo passivo tudo funciona como é suposto, no entanto em modo ativo percebesse que existiu um erro de endereço já em uso.

```
[admin@localhost ~]$ ftp 87.248.214.3
Connected to 87.248.214.3 (87.248.214.3).
220 (vsFTPd 3.0.5)
Name (87.248.214.3:admin): admin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (87,248,214,3,41,23).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Documentos
drwxr-xr-x    2 1000     1000           57 Apr 14 17:12 Imagens
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Modelos
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Música
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Público
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Transferências
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Vídeos
-rw-r--r--    1 1000     1000            0 Apr 17 15:03 a.out
drwxr-xr-x    2 1000     1000            6 Apr 08 18:04 Área de Trabalho
226 Directory send OK.
```

```
ftp> passive
Passive mode off.
ftp> ls
500 Illegal PORT command.
ftp: bind: Endereço já em uso
```

4.16. Resultados do ataque SYN Flood (DoS)

Como é possível observar pelas imagens abaixo, o ataque foi prevenido com sucesso.

```
┌──(root㉿admin)-[/home/adminsti]
└─# suricata -c /etc/suricata/suricata.yaml -q 0
i: suricata: This is Suricata version 7.0.10 RELEASE running in SYSTEM mode
i: threads: Threads created → RX: 1 W: 1 TX: 1 FM: 1 FR: 1   Engine started.
^Ci: suricata: Signal Received.  Stopping engine.
i: nfq: (RX-NFQ#0) Treated: Pkts 120247, Bytes 4809880, Errors 0
i: nfq: (RX-NFQ#0) Verdict: Accepted 2, Dropped 120245, Replaced 0
```

```
04/18/2025-19:31:38.109970  [Drop] [**] [1:1000001:1] DoS Attack (SYN FL
OOD) [**] [Classification: (null)] [Priority: 3] {TCP} 87.248.214.3:6217
1 → 192.168.10.1:80
```

```
┌──(root㉿admin)-[/home/adminsti]
└─# hping3 -S -p 80 --flood 192.168.10.1
HPING 192.168.10.1 (eth0 192.168.10.1): S set, 40 headers + 0 data byt
es
hping in flood mode, no replies will be shown
^C
— 192.168.10.1 hping statistic —
120246 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

4.17. Resultados do ataque ICMP Flood (DoS)

Como é possível observar pelas imagens abaixo, o ataque foi prevenido com sucesso.

```
┌──(root㉿admin)-[/home/adminsti]
└─# suricata -c /etc/suricata/suricata.yaml -q 0
i: suricata: This is Suricata version 7.0.10 RELEASE running in SYSTEM mode
i: threads: Threads created → RX: 1 W: 1 TX: 1 FM: 1 FR: 1   Engine started.
^Ci: suricata: Signal Received.  Stopping engine.
i: nfq: (RX-NFQ#0) Treated: Pkts 8430, Bytes 236040, Errors 0
i: nfq: (RX-NFQ#0) Verdict: Accepted 654, Dropped 6877, Replaced 0
```

```
04/18/2025-19:41:14.626594  [wDrop] [**] [1:1000002:1] DoS Attack (ICMP)
 [**] [Classification: (null)] [Priority: 3] {ICMP} 87.248.214.3:8 → 19
2.168.10.1:0
```

```
┌──(root㉿admin)-[/home/adminsti]
└─# hping3 --icmp --flood 192.168.10.1
HPING 192.168.10.1 (eth0 192.168.10.1): icmp mode set, 28 headers + 0
data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.10.1 hping statistic ---
66178 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

4.18. Resultados do ataque OS Fingerprint

Como é possível observar pelas imagens abaixo, o ataque foi prevenido com sucesso.

```
┌──(root㉿admin)-[/home/adminsti]
└─# suricata -c /etc/suricata/suricata.yaml -q 0
i: suricata: This is Suricata version 7.0.10 RELEASE running in SYSTEM mode
i: threads: Threads created → RX: 1 W: 1 TX: 1 FM: 1 FR: 1   Engine started.
^Ci: suricata: Signal Received.  Stopping engine.
i: nfq: (RX-NFQ#0) Treated: Pkts 926, Bytes 40585, Errors 0
i: nfq: (RX-NFQ#0) Verdict: Accepted 22, Dropped 904, Replaced 0
```

```
04/18/2025-19:45:09.546507  [Drop] [**] [1:1000005:1] NMAP OS Attack [**
] [Classification: (null)] [Priority: 3] {TCP} 87.248.214.3:52823 → 192
.168.10.1:110
```

```
┌──(root㉿admin)-[/home/adminsti]
└─# nmap -O -sV 192.168.10.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-18 19:44 EDT
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Ste
alth Scan
SYN Stealth Scan Timing: About 6.80% done; ETC: 19:50 (0:05:15 remaini
ng)
```

Conclusão

Através deste trabalho, foi possível compreender e implementar uma infraestrutura de rede segura e segmentada, com controlo rigoroso do tráfego entre zonas por meio de regras IPTables e NAT. A utilização do Suricata permitiu reforçar a camada de segurança com capacidades de deteção e prevenção de ataques em tempo real.

Os testes realizados demonstraram o correto funcionamento das regras implementadas, tanto em cenários de comunicação legítima como em contextos de ataques simulados. Todos os ataques (SYN Flood, ICMP Flood e OS Fingerprinting) foram devidamente detetados e bloqueados, comprovando a eficácia do sistema de proteção adotado. Porém, não foi possível realizar os ataques nem regras para os dois tipos de ataques SQL *injection*.