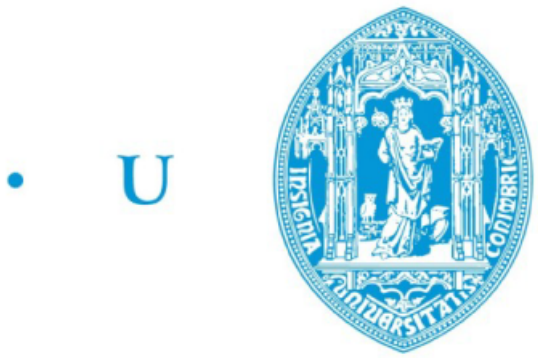


FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Segurança em Tecnologias de Informação - 2022/2023

Trabalho prático #2

Eva Teixeira 2019215185
João Dionísio 2019217030

Índice

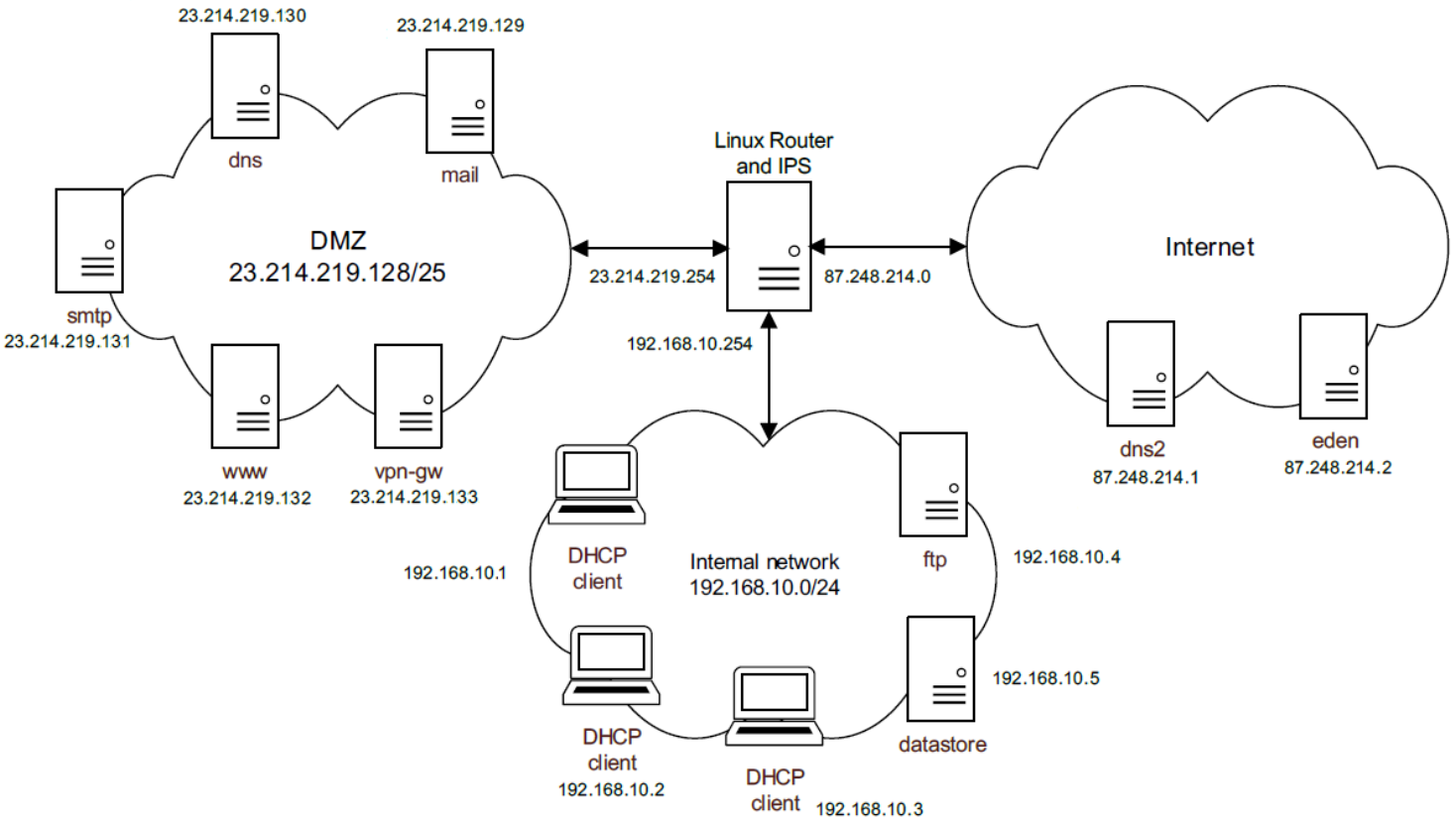
Introdução..... 2
Cenário de Rede..... 3
Conceitos gerais..... 4
    Syntax das regras de “iptables”.....4
    Snort.....6
Configuração da firewall para proteger o router.....7
Configuração da firewall para autorizar comunicações diretas (sem NAT).....8
Configuração da firewall para conexões com o endereço IP externo à firewall (com NAT).. 10
Configuração da firewall para comunicações da rede interna para o exterior (com NAT)..... 12
Deteção e Prevenção de Intrusão.....13
Testagem.....15
    Ferramentas de teste..... 15
        Netcat..... 15
        VSFTPD..... 16
    Tabela de resultados..... 17
Referências..... 19

Introdução

O presente relatório aborda os instrumentos utilizados no segundo trabalho prático, que tem como objetivo principal configurar uma firewall de rede capaz de detectar e reagir a ataques de segurança contra serviços implantados numa rede protegida. Para isso, o firewall deve implementar filtragem de pacotes, NAT e deteção de intrusão, bem como mecanismos para reagir a contra ataques de hosts do exterior (Internet).

A tarefa considera o uso de uma DMZ e redes internas, onde a DMZ é onde a maioria dos serviços públicos da organização são colocados, enquanto a rede interna fornece conectividade aos usuários (clientes com endereços IP dinâmicos), bem como suporte a servidores com objetivos específicos. O roteador interconectando as várias redes é executado no Linux e deve suportar todas as funcionalidades de segurança descritas na tarefa. Para todos os sistemas no cenário, endereços IP devem ser atribuídos, conforme apropriado.

A figura seguinte ilustra o cenário considerado para o trabalho.



Cenário de Rede

Para o desenvolvimento do projeto foram utilizadas quatro máquinas virtuais, que correspondem ao Router, DHCP Clients, DMZ e Internet.

As configurações das máquinas virtuais e das suas interfaces foram as seguintes:

DMZ: 23.214.219.128/25

mail: 23.214.219.129
dns: 23.214.219.130
smtp: 23.214.219.131
www: 23.214.219.132
vpn-gw: 23.214.219.133

Internal Network: 192.168.10.0/24

DHCP Client: 192.168.10.1
DHCP Client: 192.168.10.2
DHCP Client: 192.168.10.3
ftp: 192.168.10.4
datastore: 192.168.10.5

Internet: 87.248.214.0/24

dns2: 87.248.214.1
eden: 87.248.214.2

Router:
i-DMZ: 23.214.219.254
i-InternalNetwork: 192.168.10.254
i-Internet: 87.248.214.97

Conceitos gerais

Syntax das regras de “iptables”

Iptables é um firewall de software que permite configurar regras para controlar o tráfego de rede que entra e sai de um sistema.

As regras do iptables são definidas numa tabela de regras, que é usada pelo kernel do sistema operativo para decidir como processar os pacotes de rede. Com o iptables, é possível configurar regras para permitir ou negar o acesso a portos específicos, limitar o tráfego de rede de um determinado endereço IP ou rede, e realizar outras operações de filtragem de rede.

As iptables são constituídas por diversos componentes:

- **Tabelas:** são estruturas de dados que contêm um conjunto de cadeias. As tabelas mais comuns no iptables são: filter (filtro), nat (tradução de endereços de rede) e mangle (manipulação de pacotes);

- **Cadeias**: são conjuntos de regras que são aplicadas a um determinado tipo de tráfego de rede. Existem três cadeias padrão: INPUT (entrada), OUTPUT (saída) e FORWARD (encaminhamento);

- **Regras:** são instruções que especificam como as iptables devem tratar o tráfego de rede. Podem permitir ou negar o tráfego de rede com base em várias condições, como endereços IP de origem e destino, portas de rede, protocolos de rede e outros fatores;

- **Alvos:** são ações que as iptables realizam quando uma regra é correspondida. Por exemplo, usa-se DROP para descartar pacotes de rede indesejados, e ACCEPT para permitir o tráfego de rede;

- **Módulos:** são usados para fornecer funcionalidades adicionais, como a filtragem de pacotes baseada em estado (stateful filtering) e a correspondência de padrões (pattern matching).

iptables -t [table] -[option] [CHAIN] [matching component] [action component]

Main tables:

- Filter (Default table) - usada para filtrar pacotes.
- NAT(Network Address Translation) - usada para funcionalidades de NAT.
- Mangle - usada para modificar pacotes específicos.

Main options:

- A (append)	- Z (zero counters)	- F (flush)
- D (delete)	- L (list)	- N (new user defined chain)
- I (insert)	- P (policy)	- X (delete chain)
- R (replace)	- E (rename)	

Chains:

- PREROUTING - altera pacotes antes de serem direcionados.
- INPUT - lida com o tráfego de entrada destinado à máquina local.
- FORWARD - lida com o tráfego de entrada com um destino diferente da máquina local.
- OUTPUT - lida com o tráfego de saída originado a partir da máquina local.
- POSTROUTING - altera pacotes depois de serem direcionados.

Matching component:

- usado para corresponder e filtrar pacotes.

Action component:

- Especifica a ação a ser exercida sobre os pacotes filtrados.

Snort

O Snort é um sistema de detecção de intrusão de rede que é utilizado para monitorizar o tráfego de rede e identificar atividades maliciosas. Tem como objetivo analisar o tráfego de rede em tempo real e alertar os administradores sobre possíveis ameaças, incluindo tentativas de invasão, ataques de DoS, atividades suspeitas de malware e outros comportamentos maliciosos.

Tem 4 modos de funcionamento, que são:

- **Modo Sniffer:** captura e exibe pacotes de rede em tempo real;
- **Modo Packet Logger:** captura e registra pacotes de rede num arquivo de log;
- **Modo NIDS (Network Intrusion Detection System):** monitoriza o tráfego de rede em tempo real e alerta os administradores sobre possíveis atividades maliciosas com base em regras de detecção de ameaças previamente definidas;
- **Modo NIPS (Network Intrusion Prevention System):** monitora o tráfego de rede em tempo real e alerta os administradores sobre possíveis atividades maliciosas, para além de também pode tomar ações preventivas para bloquear ou filtrar o tráfego de rede malicioso.

Configuração da firewall para proteger o router

Inicialmente descartámos todas as configurações pré-definidas existentes nas iptables e parámos e desativámos a firewall.

```
iptables -F  # clean all firewall rules
iptables -X  # delete all chains
iptables -Z  # reset packet counter of all default table rules

systemctl stop firewalld # stop firewalld service
systemctl disable firewalld # disable firewalld services
```

Em seguida, alterámos a política das tabelas “INPUT” e “FORWARD” para descartar todos os pacotes deste tipo.

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

Depois usámos o seguinte comando para permitir que as conexões já estabelecidas possam continuar a comunicar-se normalmente sem serem bloqueadas pelo firewall.

```
# Allow incoming and outgoing traffic for established connections and to allow responses to outgoing traffic
iptables -t filter -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Em seguida, permitimos apenas as ligações necessárias para o funcionamento normal dos seguintes serviços, nomeadamente:

• Solicitações de resolução de nomes DNS enviadas para servidores externos
Permitir o tráfego de DNS no porto UDP 53. Adicionar duas regras na tabela *filter* do iptables para permitir a entrada e saída de tráfego DNS no porto 53.

```
# Allow DNS connections INPUT
iptables -A INPUT -p udp --dport 53 -j ACCEPT

# Allow DNS traffic for name resolution
iptables -A INPUT -p udp --sport 53 -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

• Conexões SSH com o sistema router, se originadas na rede interna ou no VPN Gateway (vpn_gw).

Configurámos as iptables de sejam permitidas conexões SSH, no porto 22, com origem na Internal Network ou do VPN Gateway (DMZ).

```
# Allow SSH connections INPUT
iptables -A INPUT -p tcp -s ${Internal_Network} --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -s ${vpn_gw} --dport 22 -j ACCEPT
```

Configuração da firewall para autorizar comunicações diretas (sem NAT)

A configuração da firewall deve descartar todas as comunicações entre as redes (descrito acima), exceto as necessárias para o normal funcionamento dos serviços indicados:

• Permitir resoluções de domínio de nomes usando o servidor DNS.

Para isso permitimos que o servidor envie pacotes UDP para o cliente no porto 53, porto padrão, e que pacotes UDP sejam recebidos pelo cliente na porta 53.

```
# Allow DNS resolutions using DNS server
iptables -A FORWARD -p udp --sport 53 -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -j ACCEPT
```

• O servidor DNS deve ser capaz de resolver nomes usando servidores DNS na Internet.

Para isso, permitimos que o servidor DNS receba os pacotes UDP vindos dos servidores DNS e DNS2.

```
# Allow DNS traffic to Internet DNS servers
iptables -t filter -A FORWARD -p udp --dport 53 -d ${dns2} -j ACCEPT
iptables -t filter -A FORWARD -p udp --dport 53 -d ${dns} -j ACCEPT
```

• Os servidores DNS e DNS2 devem ser capazes de sincronizar o conteúdo das zonas DNS.

Para isso, permitimos que os pacotes TCP com origem e destino no porto 53 e endereço DNS2 sejam encaminhados pela cadeia "FORWARD".

```
# The dns and dns2 servers synchronize the contents of DNS zones.
iptables -A FORWARD -p tcp --dport 53 -s ${dns2} -j ACCEPT
```

```
iptables -A FORWARD -p tcp --sport 53 -d ${dns2} -j ACCEPT
```

• Conexões SMTP com o servidor SMTP.

Permitimos que pacotes os TCP com destino no porto 25 e endereço smtp sejam encaminhados pela cadeia "FORWARD". Isto irá que clientes de e-mail se conectem ao servidor SMTP para enviar e-mails. Para além disso, permitimos que pacotes os TCP com origem no smtp e no porto 25, sejam encaminhados pela cadeia "FORWARD".Isto permitirá que o servidor SMTP envie e-mails para outros servidores SMTP.

```
 # SMTP connections to the smtp server.
iptables -A FORWARD -p tcp --dport 25 -d ${smtp} -j ACCEPT
iptables -A FORWARD -p tcp --sport 25 -s ${smtp} -j ACCEPT
```

• Conexões POP e IMAP com o servidor de correio

Permitimos que os pacotes TCP com destino no porto 110 e destino no endereço *mail* sejam encaminhados pela cadeia "FORWARD". Essa porta é usada para o protocolo POP3. Realizámos o mesmo procedimento para o endereço IP mail e para os portos 143, 993 e 995, que correspondem aos protocolos IMAP, SSL/TLS, SSL/TLS, respetivamente.

```
# POP and IMAP connections to the mail server.
iptables -A FORWARD -p tcp --dport 110 -d ${mail} -j ACCEPT
iptables -A FORWARD -p tcp --dport 143 -d ${mail} -j ACCEPT
iptables -A FORWARD -p tcp --dport 993 -d ${mail} -j ACCEPT
iptables -A FORWARD -p tcp --dport 995 -d ${mail} -j ACCEPT
```

• Conexões HTTP e HTTPS com o servidor www.

Permitimos que os pacotes TCP destinados ao endereço IP www e portos 80 e 443 sejam encaminhados pela cadeia “FORWARD”. As portas mencionadas correspondem aos protocolos HTTP e HTTPS, respetivamente.

```
# HTTP and HTTPS connections to the www server.
iptables -A FORWARD -p tcp --dport 80 -d ${www} -j ACCEPT
iptables -A FORWARD -p tcp --dport 443 -d ${www} -j ACCEPT
```

• Conexões OpenVPN com o servidor vpn-gw.

Permitimos que os pacotes UDP com origem no endereço IP vpn_gw e destino destino à porta 1194 e 443 sejam encaminhados pela cadeia "FORWARD". Esses portos são utilizados para o protocolo OpenVPN.

```
# OpenVPN connections to the vpn-gw server
iptables -A FORWARD -p udp --dport 1194 -s ${vpn_gw} -j ACCEPT
iptables -A FORWARD -p udp --dport 443 -s ${vpn_gw} -j ACCEPT
```

• Os clientes VPN conectados ao gateway (vpn-gw) devem poder se conectar a todos os serviços na rede interna (assuma que o gateway faz SNAT/MASQUERADING para comunicações recebidas de clientes).

Permitimos que os pacotes enviados pelos clientes VPN e com destino ao endereço IP ftp ou datastore sejam encaminhados pela cadeia "FORWARD". Isso garantiu que os clientes VPN possam aceder aos serviços de FTP e datastore que estão na Internal Network.
Para além disso, garantimos que os pacotes enviados pelos clientes VPN pareçam ter sido enviados pelo servidor VPN quando chegam à rede interna, utilizando o NAT.

```
# VPN clients connect to all services in the Internal network
iptables -A FORWARD -s ${vpn_gw} -d ${ftp} -j ACCEPT
iptables -A FORWARD -s ${vpn_gw} -d ${datastore} -j ACCEPT

# SNAT/MASQUERADING for VPN clients
iptables -t nat -A POSTROUTING -s ${vpn_gw} -o ens36 -j MASQUERADE
```

Configuração da firewall para conexões com o endereço IP externo à firewall (com NAT)

Antes de efetuarmos as configurações da firewall instalámos os módulos FTP, de modo a que a firewall possa filtrar e monitorizar as conexões FTP e permitir que este serviço funcione corretamente através da firewall.

```
# Enable FTP service modules
modprobe ip_conntrack_ftp
modprobe nf_nat_ftp  # Translate from addresses (NAT) to FTP connections
```

Começamos por permitir o tráfego de FTP entre a Internal Network, o servidor FTP e a Internet. Para tal, definimos a aceitação de pacotes TCP com destino aos portos 20 e 21, com origem na Internet e Internal Network, respetivamente, para o servidor FTP. Além disso, permitimos que os pacotes com destino ao porto 20 e com origem no porto 21 sejam aceites. O porto 21 serve para estabelecer uma conexão de controlo entre o cliente e o servidor FTP, enquanto que o porto 20 serve para transferência de dados.

```
# FTP connections to the ftp server.
iptables -A FORWARD -p tcp -d ${Internal_Network}  -s ${Internet} --sport 21 -j ACCEPT
```

```
iptables -A FORWARD -p tcp -s ${Internal_Network}  -d ${Internet} --dport 20 -j ACCEPT
iptables -A FORWARD -p tcp -d ${ftp} --dport 21 -j ACCEPT
iptables -A FORWARD -p tcp -s ${ftp} --sport 20 -j ACCEPT

# Redirect traffic
iptables -t nat -A PREROUTING -p tcp -s ${Internet} -d ${i_Internet} --dport 21 -j DNAT --to-destination ${ftp}
iptables -t nat -A PREROUTING -p tcp -s ${Internet} -d ${i_Internet} --dport 20 -j DNAT --to-destination ${ftp}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 21 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 20 -j SNAT --to-source ${i_Internet}
```

Em seguida, permitimos as conexões SSH ao servidor, mas apenas aquelas com origem nos servidores eden ou dn2, ou em qualquer endereço da Internal Network

```
# SSH connections to the datastore sever, but only if originated at the eden or dns2 servers
iptables -A FORWARD -p tcp -s ${eden} -d ${datastore} --dport 22 -j ACCEPT
iptables -A FORWARD -p tcp -s ${dns2} -d ${datastore} --dport 22 -j ACCEPT
iptables -A FORWARD -s ${Internal_Network} -p tcp --dport 22 -j ACCEPT

# Network address translation (NAT) to allow SSH connections to be routed correctly
iptables -t nat -A PREROUTING -p tcp -s ${Internet} -d ${i_Internet} --dport 22 -j DNAT --to-destination ${datastore}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 22 -j SNAT --to-source ${i_Internet}
```

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

Começamos por permitir que a Internal Network aceda ao servidor DNS na Internet para resolução de nomes de domínio.

```
# Domain name resolutions using DNS
# Allow TCP and UDP packet traffic on port 53, from the Internal Network to the Internet
iptables -A FORWARD -p tcp -s ${Internal_Network} -d ${Internet} --dport 53 -j ACCEPT
iptables -A FORWARD -p udp -s ${Internal_Network} -d ${Internet} --dport 53 -j ACCEPT

# Translate the source address (SNAT) for packets leaving the Internal_Network towards the internet, through port 53
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 53 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p udp -s ${Internal_Network} -d ${Internet} --dport 53 -j SNAT --to-source ${i_Internet}
```

Em seguida, permitimos as conexões de rede para os protocolos HTTP (porto 80), HTTPS (porto 443) e SSH (porto 22), a partir da Internal Network em direção à Internet.

```
# HTTP, HTTPS and SSH connections
iptables -A FORWARD -p tcp -s ${Internal_Network} -d ${Internet} --dport 80 -j ACCEPT
iptables -A FORWARD -p udp -s ${Internal_Network} -d ${Internet} --dport 443 -j ACCEPT
iptables -A FORWARD -p tcp -s ${Internal_Network} -d ${Internet} --dport 443 -j ACCEPT
iptables -A FORWARD -p tcp -s ${Internal_Network} -d ${Internet} --dport 22 -j ACCEPT

iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 80 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p udp -s ${Internal_Network} -d ${Internet} --dport 443 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 443 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 22 -j SNAT --to-source ${i_Internet}
```

Por último, permitimos a conexão com servidores FTP externos, nomeadamente nos portos 20 e 21. O uso dos portos 20 e 21 é para suportar os modos de transferência de ficheiros ativo e passivo do FTP.

```
# FTP connections (passive/active modes) to external FTP servers.
iptables -A FORWARD -p tcp -s ${Internal_Network} --dport 21 -j ACCEPT
iptables -A FORWARD -p tcp -d ${Internal_Network} --dport 20 -j ACCEPT

iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 20 -j SNAT --to-source ${i_Internet}
iptables -t nat -A POSTROUTING -p tcp -s ${Internal_Network} -d ${Internet} --dport 21 -j SNAT --to-source ${i_Internet}
```

Deteção e Prevenção de Intrusão

De modo a detetar e prevenir intrusões, é necessário configurar a firewall para que esta consiga detetar e reagir a ataques de segurança, bloqueando-os.

Para tal, usámos o Snort e as IPtables.

Em primeiro lugar instalamos o Snort, seguindo o tutorial de instalação presente no livro Segurança Prática em Sistemas e Redes com Linux [2] . Foi incluída da linha “include local.RULES” no ficheiro “snort.conf”, e as regras do Snort foram adicionadas no ficheiro local.RULES

- **Snort**

Permitir o tráfego TCP no porto 1433, tanto para entrada quanto para saída, permitindo o acesso ao Microsoft SQL Server.

```
sudo iptables -A INPUT -p tcp --dport 1433 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 1433 -j ACCEPT
```

A firewall deve detetar e bloquear os seguintes ataques:

- **Dois tipos de injeção SQL.**

A injeção SQL é uma técnica de ataque em que um invasor tenta inserir código SQL malicioso num formulário web, de modo a comprometer a segurança do servidor e dos dados nele armazenados. São uma forma de ataque cibernético que exploram vulnerabilidades em aplicações da web que não validam adequadamente as entradas de utilizador. Com um ataque de injeção SQL bem-sucedido, um invasor pode executar comandos maliciosos na base de dados da aplicação, permitindo-lhe ler, modificar ou deletar dados.

Assim sendo, com o objetivo de impedir este tipo de ataques, pode descartar-se solicitações que contenham strings específicas que são comumente usadas em ataques desse tipo, tais como “or 1=1--” e “union”. Devemos selecionar o porto em que queremos implementar estas regras, 1433 por exemplo, e podemos definir a sensibilidade a maiúsculas e minúsculas através do parâmetro “nocase” .

```
drop tcp any any -> any 1433 (msg:"SQL Injection Attack Detected"; content:"' or 1=1--"; nocase; sid:100001;)
drop tcp any any -> any 1433 (msg:" Union-Based SQL Injection Detected"; content:"union"; nocase; content:"select"; nocase; sid:1000002; rev:1;)
```

- **Dois tipos de ataques DoS**

Ataques DoS são tentativas maliciosas de sobrecarregar um sistema ou rede com um grande volume de tráfego, pacotes ou solicitações, impedindo que os utilizadores autenticados acedam aos serviços e recursos do sistema. Esses ataques podem ser realizados de várias maneiras, incluindo o envio de tráfego excessivo, exploração de vulnerabilidades em serviços e aplicações, inundação de conexões, entre outras técnicas.

Uma das técnicas de prevenção contra este tipo de ataques é configurar a firewall de modo a bloquear grandes pacotes ICMP e com o bit de fragmentação definido, e ativar a fragmentação de pacotes nos routes para que os pacotes grandes possam ser fragmentados em pacotes menores, antes de serem enviados para o seu destino. Esta é uma técnica de prevenção contra o *Ping of Death.*

Outra técnica de prevenção contra este tipo de ataques é configurar a firewall de modo a limitar o número de conexões TCP simultâneas de uma única fonte TCP, para além de configurar o sistema operativo para aumentar o backlog de conexões TCP, para que possa lidar com várias solicitações de conexão TCP em simultâneo. Esta técnica previne o *SYN Flood.*

```
drop icmp any any -> any any (msg:"Ping of Death Detected"; fragbits:M; sid:100003;)
drop tcp any any -> any 80 (msg:"SYN Flood Detected"; flags:S; threshold: type both, track by_src, count 100, seconds 10; sid:100004;)
```

- **OS fingerprinting attempts**

OS fingerprinting é uma técnica utilizada por hackers para descobrir informações sobre o sistema operativo que está a ser executado num determinado alvo. Essa técnica geralmente envolve o uso de pacotes de rede para obter informações sobre a configuração do sistema operativo, como versão do kernel, tempo de atividade, tipo de dispositivo de rede, entre outras informações. Essas informações podem ser usadas pelos invasores para identificar vulnerabilidades específicas do sistema operativo e, assim, lançar ataques direcionados e bem-sucedidos.

Uma técnica para detetar este tipo de ataques que são feitas com ferramentas como o Nmap, é lançar um alerta sempre que qualquer pacote TCP contenha o conteúdo “Nmap” no seu payload.

```
drop tcp any any -> any any (flags:S,12; detection_filter:track by_src, count 1, seconds 60; sid:10000004; rev:1; msg:"Possible OS fingerprinting attempt";)
```

Testagem

Ferramentas de teste

Netcat

O netcat é uma ferramenta de linha de comando que permite que os usuários se comuniquem com outros computadores por meio de conexões de rede usando diferentes protocolos, como TCP e UDP.

Esta ferramenta permite realizar testes de conectividade, enviando uma solicitação de conexão no porto pretendido, de modo a verificar se o porto está aberto. Para que o porto esteja aberto é necessário colocar o porto à escuta.

Assim sendo, utilizámos os seguinte comando:

- **Servidor**

```
nc -l <porto> # add -u in case of a udp connection instead of tcp
```

Para estabelecer ligação ao servidor a partir do cliente foi utilizado o seguinte comando:

- **Cliente**

```
nc -v <endereço_IP>  <porto> # add -u in case of a udp connection instead of tcp
```

VSFTPD

O VSFTPD (Very Secure FTP Daemon) é um servidor FTP de código aberto que é usado para transferir arquivos entre sistemas através da rede.

Para testar o VSFTPD, é preciso ter acesso a um servidor que esteja a executar o software e ter as credenciais de login corretas.

Em primeiro lugar é necessário instalar o servidor FTP

```
yum install vsftpd
sudo chown jopedro:jopedro /home/jopedro
sudo chmod 755 /home/jopedro
```

De seguida é necessário conectar o cliente ao servidor, inserindo o endereço IP seguido das credenciais de login. Ao ser estabelecida a conexão podemos alterar o modo de conexão, passivo ou ativo, e enviar ficheiros através do FTP.

Os comandos utilizados são os seguintes:

- **Servidor**

```
systemctl start  vsftpd
```

- **Cliente FTP**

```
ftp
passive (para mudar o modo)
open <ip do server>
# insert credentials
ls # to display elements in directory
```

Tabela de resultados

Com o intuito de verificar se todos os pontos do projeto foram cumpridos, realizamos diversos testes, de modo a comprovar o funcionamento das diversas conexões estabelecidas.

Assim sendo, anotámos na tabela seguinte os testes efetuados entre as diversas máquinas e as ferramentas e serviços utilizados, bem como o resultado obtido com cada teste.

Origem	**Destino**	**Serviços**	**Estado**	**Ferramenta**
Router	DMZ - dns	DNS	**OK**	Netcat
Internal Network - dhcp_client	Router	SSH	**OK**	Netcat
DMZ - vpn_gw	Router	SSH	**OK**	Netcat
Internet - dns2	DMZ - dns	DNS	**OK**	Netcat
Internet - dns2	DMZ	DNS	**OK**	Netcat
DMZ - dns	Internet - dn2	DNS	**OK**	Netcat
Internet	DMZ - smtp	SMTP	**OK**	Netcat
Internal	DMZ - smtp	SMTP	**OK**	Netcat
Internet	DMZ - mail	POP3	**OK**	Netcat
Internal	DMZ - mail	POP3	**OK**	Netcat
Internet	DMZ - mail	IMAP	**OK**	Netcat
Internal	DMZ - mail	IMAP	**OK**	Netcat
Internet	DMZ - www	HTTP	**OK**	Netcat
Internal	DMZ - www	HTTP	**OK**	Netcat
Internet	DMZ - www	HTTPS	**OK**	Netcat
Internal	DMZ - www	HTTPS	**OK**	Netcat
Internet	DMZ - vpn-gw	OpenVPN	**OK**	Netcat
Internal	DMZ - vpn-gw	OpenVPN	**OK**	Netcat
DMZ - vpn-gw	Internal - datastore	SSH	**OK**	Netcat

DMZ - vpn-gw	Internal - ftp	FTP	**OK**	vsftp
Internal	Internet	FTP	**OK**	Netcat
Internet	Internal - ftp	FTP	**OK**	vsftp
Internet - eden	Internal - datastore	SSH	**OK**	Netcat
Internet - dns2	Internal - datastore	SSH	**OK**	Netcat
Internal	Internet	DNS	**OK**	Netcat
Internal	Internet	HTTP	**OK**	Netcat
Internal	Internet	HTTPS	**OK**	Netcat
Internal	Internet	SSH	**OK**	Netcat
Internal	Internet	FTP	**OK**	Netcat

No caso do Snort, como não utilizamos uma web application, na qual os ataques possam ser testados, recorremos ao netcat usando os seguintes comandos:

SQL Injection

- **Router**

```
nc -lvp 1433
```

- **Atacante**

```
echo "' or 1=1--" | nc <IP do Servidor> 1433   # test SQL injection (1)
echo "union" | nc <IP do Servidor> 1433   # test SQL injection (1)
```

DoS

- **DMZ**

```
nc -l 80
```

- **Atacante**

```
while true; do nc -zvw1 23.214.219.132 80; done
```

OS Fingerprint

- **Atacante**

```
nmap -O <endereço_IP>
```

É importante relembrar que antes de testar a ligação corremos o snort num terminal à parte, usamos o seguinte comando:

```
snort -Q --daq nfq --daq-var queue=0 -c /etc/snort/snort.conf -v -l /var/log/snort
tail /var/log/snort/alert -v
```

Referências

[1] Slides de apoio às aulas teóricas e práticas-laboratoriais da edição da cadeira de Segurança em Tecnologias de Informação de 2022/2023

[2] Segurança Prática em Sistemas e Redes com Linux, Jorge Granjal, editor:
FCA, isbn: 9789727228652

[3] https://www.snort.org/