Information Technology Security

Authors:
Edgar Duarte no. 2019216077
Tatiana Almeida, no. 2019219581

April 23, 2023

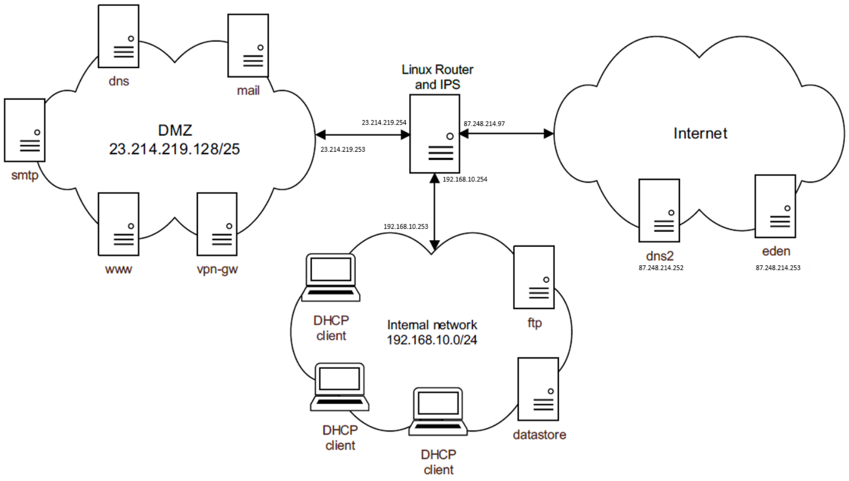
1 Introduction

Nowadays, the importance of information technology (IT) security has increased as a way of guaranteeing the security and protection of sensitive data and services. Given the rise in cyber threats and assaults, companies and organizations need to have robust security measures in place to guard against unauthorized access, data breaches, and system intrusions.Due to this, in this project a network firewall that is capable of identifying and responding to security attacks against services running on a protected network is implemented.

The instructions in this report describe how to set up a network firewall that can effectively filter incoming and outgoing traffic, provide network address translation (NAT), and identify and handle any security threats. The firewall will be a crucial defense tool to protect the protected network from attacks from outside hosts on the Internet and to maintain its security and integrity.

2 Network Scenario

Taking the scenario outlined in the project statement into consideration, we designed the project architecture with three networks: one for the internal network, 192.168.10.0, another for the DMZ, 23.214.219.128, and one to simulate the internet, 87.0.0.0. Based on the diagram below, the IP addresses for the interfaces were selected.



3 Router Protection

To ensure that no incoming communication is allowed into the router system, we executed the following command for the firewall configuration to drop all such communications.

```
iptables -P INPUT DROP
```

As stated in the assignment, we established a set of regulations to guarantee a specific type of communication.

3.1 DNS name resolution requests sent to outside servers.

Once the connections were declined, the acceptance rules were established. The next step enables the router to send a DNS request to the DNS server and DNS2 from the internet. The command grants access to both devices' service addresses as defined by the source (-s) and approves connections with the router's gateway as defined by the destination (-d).

```
iptables -A INPUT -p udp --sport domain -j ACCEPT
iptables -A INPUT -p tcp --sport domain -j ACCEPT
```

3.2 SSH connections to the router system, if originated at the internal network or at the VPN gateway

The SSH protocol is actively used to provide communication and data exchange between the two machines. This method makes it easier for your gateway to accept connections from the internal network, 192.168.10.0/24, and connections from a VPN gateway, 23.214.219.128/25.

```
iptables -A INPUT -s 192.168.10.0/24 -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -s 23.214.219.253 -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -s 192.168.10.0/24 -p udp --dport ssh -j ACCEPT
iptables -A INPUT -s 23.214.219.253 -p udp --dport ssh -j ACCEPT
```

4 Authorize direct communications

To guarantee that no unintended forwarding rules affect the behaviour of the network, initially, all rules are dropped:

```
iptables -P FORWARD DROP
iptables -A FORWARD -j NFQUEUE --queue-num 0
```

The second rule is for Snort to be able to capture the packets in a queue.

4.1 Domain name resolutions using the dns server

Next, the forwarding of DNS responses through the dns port are enabled through the following command:

```
iptables -A FORWARD -p udp -d 23.214.219.253 --dport domain -j
    ACCEPT
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport domain -j
    ACCEPT
```

4.2 The dns server should be able to resolve names using DNS servers on the Internet (dns2 and also others)

To allow the dns server to access the Internet in order to resolve names, an udp request is allowed with the following iptables command:

```
iptables -A FORWARD -p udp -s 23.214.219.253 -d 87.0.0.0/8 --dport
    domain -j ACCEPT
iptables -A FORWARD -p udp -d 23.214.219.253 -s 87.0.0.0/8 --dport
    domain -j ACCEPT
```

4.3 The dns and dns2 servers should be able to synchronize the contents of DNS zones

To synchronize the requests, the tcp protocol has to be allowed, seeing that this protocol allows for synchronization between two server. The rest of the command is equal to the last one:

```
iptables -A FORWARD -p tcp -s 23.214.219.128/25 -d 87.0.0.0/8 --
    dport domain -j ACCEPT
iptables -A FORWARD -p tcp -d 23.214.219.128/25 -s 87.0.0.0/8 --
    dport domain -j ACCEPT
```

4.4 SMTP connections to the smtp server

SMTP is an Internet standard protocol for mail transfer, being used to send and receive mail messages. For this, the following rule was defined:

```
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport smtp -j ACCEPT

iptables -A FORWARD -p udp -d 23.214.219.253 --dport smtp -j ACCEPT
```

4.5 POP and IMAP connections to the mail server

POP is one of, if not the most, used message request protocol in the Internet for transferring messages from a mail server to a client, while IMAP allows a user to access their email wherever they are. To allow for these connections:

```
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport pop3 -j ACCEPT
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport imap -j ACCEPT
iptables -A FORWARD -p udp -d 23.214.219.253 --dport pop3 -j ACCEPT
iptables -A FORWARD -p udp -d 23.214.219.253 --dport imap -j ACCEP
```

4.6 HTTP and HTTPS connections to the www server

Hypertext Transfer Protocol (HTTP) is used to encode and transport information over the web. HTTPS is a secure version of HTTP that uses the SSL/TLS protocol for encryption and authentication. To enable these protocols, the following rules were used:

```
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport http -j ACCEPT
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport https -j
    ACCEPT
iptables -A FORWARD -p udp -d 23.214.219.253 --dport http -j ACCEPT
iptables -A FORWARD -p udp -d 23.214.219.253 --dport https -j
    ACCEPT
```

4.7 OpenVPN connections to the vpn-gw server

OpenVPN is an open-source virtual private network (VPN) software that allows users to create secure connections over the internet using encryption. To enable connections of this type the following rules were implemented:

```
iptables -A FORWARD -p udp -d 23.214.219.253 --dport openvpn -j
    ACCEPT
iptables -A FORWARD -p tcp -d 23.214.219.253 --dport openvpn -j
    ACCEPT
```

4.8 VPN clients connected to the gateway (vpn-gw) should be able to connect to all services in the Internal network

To achieve this the following rules were used:

```
iptables -A FORWARD -p udp -d 192.168.10.0/24 -s 23.214.219.253 --
    sport openvpn -j ACCEPT
iptables -A FORWARD -p tcp -d 192.168.10.0/24 -s 23.214.219.253 --
    sport openvpn -j ACCEPT

iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

5 Connections to the external IP address of the firewall(NAT)

5.1 FTP connections (in passive and active modes) to the ftp server

For this the router needs to accept ftp and ftp-data packets and needs to route these packets accordingly, using DNAT and SNAT. DNAT will be used when the Internet tries to send a ftp packet to the internal network, and SNAT will be used when the internal network replies with a ftp-data packet to the Internet.

```
iptables -A FORWARD -p tcp -s 87.0.0.0/8  --dport ftp -j ACCEPT
iptables -A FORWARD -p tcp -s 192.128.10.253 --sport ftp-data -j
    ACCEPT
iptables -t nat -A PREROUTING -s 87.0.0.0/8 -d 87.248.214.97 -p tcp
     --dport ftp -j DNAT --to-destination 192.168.10.253
iptables -t nat -A POSTROUTING -s 192.168.10.253 -p tcp --sport ftp
    -data -j SNAT --to-source 87.248.214.97
```

5.2 SSH connections to the datastore server, but only if originated at the eden or dns2 servers

To enable SSH connections, the router has to accept ssh connections that come from the internet and uses DNAT to allow machines in the Internet services to use the dmz services:

```
iptables -A FORWARD -p tcp -s 87.248.214.253 -d 192.168.10.253 --
    dport ssh -j ACCEPT
iptables -A FORWARD -p tcp -s 87.248.214.252 -d 192.168.10.253 --
    dport ssh -j ACCEPT

iptables -t nat -A PREROUTING -p tcp -s 87.248.214.253 -d
    87.248.214.254 --dport ssh -j DNAT --to-destination
    192.168.10.253
iptables -t nat -A PREROUTING -p tcp -s 87.248.214.252 -d
    87.248.214.254 --dport ssh -j DNAT --to-destination
    192.168.10.253
```

6 Firewall configuration for communications from the internal network to the outside (using NAT)

6.1 Domain name resolutions using DNS

The router needs to allow for dns requests from the internal server to the Internet and needs to enable SNAT for this type of connection.

```
iptables -A FORWARD -p udp -s 192.168.10.0/24 -d 87.0.0.0/8 --dport
     domain -j ACCEPT
iptables -t nat -A POSTROUTING -p udp -s 192.168.10.0/24 -d
    87.0.0.0/8 --dport domain -j SNAT --to-source 87.248.214.97

iptables -A FORWARD -p tcp -s 192.168.10.0/24 -d 87.0.0.0/8 --dport
     domain -j ACCEPT
iptables -t nat -A POSTROUTING -p tcp -s 192.168.10.0/24 -d
    87.0.0.0/8 --dport domain -j SNAT --to-source 87.248.214.97
```

6.2 HTTP, HTTPS and SSH connections

The router needs to accept http, https and ssh packets that come from the internal network and needs to allow for SNAT from the internal network to the Internet:

```
iptables -A FORWARD -p tcp -s 192.168.10.0/24 -d 87.248.214.253  --
    dport http -j ACCEPT
iptables -A FORWARD -p tcp -s 192.168.10.0/24 -d 87.248.214.253  --
    dport https -j ACCEPT
iptables -A FORWARD -p tcp -s 192.168.10.0/24 -d 87.248.214.253  --
    dport ssh -j ACCEPT

iptables -t nat -A POSTROUTING -p udp -s 192.168.10.0/24 -d
    87.0.0.0/8 --dport http -j SNAT --to-source 87.248.214.97
iptables -t nat -A POSTROUTING -p udp -s 192.168.10.0/24 -d
    87.0.0.0/8 --dport https -j SNAT --to-source 87.248.214.97
iptables -t nat -A POSTROUTING -p udp -s 192.168.10.0/24 -d
    87.0.0.0/8 --dport ssh -j SNAT --to-source 87.248.214.97
```

6.3 FTP connections (in passive and active modes) to external FTP servers

For this the router needs to accept ftp and ftp-data packets and needs to route these packets accordingly, using DNAT and SNAT. SNAT will be used when the internal network sends a ftp packet to the Internet and DNAT will be used when the Internet replies with a ftp-data packet to the internal network.

```
iptables -A FORWARD -s 192.168.10.253 -p tcp --dport ftp -j ACCEPT
iptables -A FORWARD -s 87.0.0.0/8 -p tcp --sport ftp-data -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.10.253 -p tcp --dport ftp
     -j SNAT --to-source 87.248.214.97
iptables -t nat -A PREROUTING -s 87.0.0.0/8 -p tcp --sport ftp-data
     -j DNAT --to-destination 192.168.10.253
```

7 Snort

Snort is an open source intrusion prevention and prevention system (IPS) that, through the use of rules defined by the users, is responsible for analyzing real-time network data and catching threats, such as attacks, exploits and code injections. In this project, we configured Snort in order to filter packets whose objective is:

- injecting SQL to exploit vulnerabilities in a database;
- causing DoS attacks with the objective of overwhelming a system;
- maliciously identify a user's network protocols, hardware devices and network topology (OS fingerprinting attempts).

To detect these attacks, a file named *snort.rules* in the folder */etc/snort/rules* was created hosts the rules necessary to prevent them. To make sure that snort detects packets, the following commands had to be activated:

```
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe nfnetlink_queue
```

7.1 SQL injection

Two SQL Injection rules were be implemented. The verifies if a ”drop table” query is in the user’s message, while the second verifies if the message contains a ”UNION SELECT” attack, a very common attack that attempts to extract otherwise hidden information, from a SQL query.

```
drop tcp any any -> 192.168.10.253 any (msg:"SQL injection detected
    "; content:"drop table"; sid:100001; rev:1;)
drop tcp any any -> 192.168.10.253 any (msg:"UNION SQL injection
    detected";content:"UNION SELECT";sid:100002; rev:1;)
```

7.2 Two types of DoS (Denial of Service) attacks.

A denial of service attack is one that makes the victim’s computer or device unavailable. To combat these types of attacks, the following two SNORT rules were implemented, one to detect UDP flood attacks and another to detect TCP flood attacks:

```
alert udp any any -> 192.168.10.253 any (msg:"ALERT UDP Flood";
    threshold:type both, track by_dst, count 15, seconds 3; sid
    :1000001; rev:1;)
alert tcp any any -> 192.168.10.253 any (msg:"ALERT TCP Flood";
    threshold:type both, track by_dst, count 15, seconds 3; sid
    :1000005; rev:1;)
```

7.3 OS fingerprinting attempts

An OS fingerprinting attack is one that attempts to identify and retrieve the network protocols and topology of a victim. To detect these type of attacks, the following Snort rule was implemented:

```
alert tcp any any -> any any (msg:"Nmap OS Fingerprinting Detected
    ";flags:S;reference:url,nmap.org; sid:1000003; rev:1;)
```

The rule verifies if an NMAP packet with the flag -sS is sent then an OS fingerprinting is occurring, prompting snort to intervene.

7.4 Snort Commands

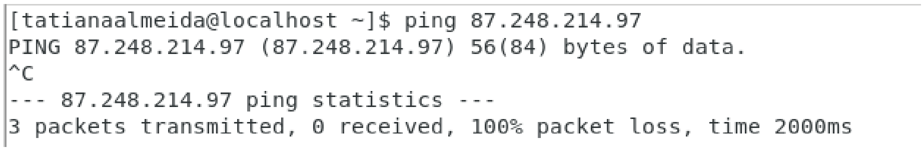
Between the three types of attacks the following parameters are common:

- **”drop”** - Snort rejects any packet that complies with the content of the rule;
- **”tcp”** - Protocol used;
- **”any”** - Source IP, in this case all IPs;
- **”any”** - Source Port, in this case all ports;
- **”->”** - Indicates the flow of the packets, in this case they are moving from the source to the destination;
- **”192.168.10.253”** - destination IP ;
- **any** - destination port;
- **”msg”** - Message that appears when Snort detects the defined rule;
- **”content”** - verifies if the string is in the message’s content;
- **”sid”** - identifies the rule;
- **”rev”** - revision number of the rule.

8 Tests

8.1 Drop all communications entering

It will be impossible to ping the router because we have blocked all incoming traffic to it.

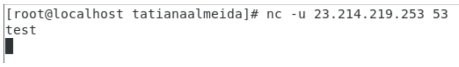


```
[tatianaalmeida@localhost ~]$ ping 87.248.214.97
PING 87.248.214.97 (87.248.214.97) 56(84) bytes of data.
^C
--- 87.248.214.97 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2000ms
```

Figure 1: Ping from internet to router’s external interface

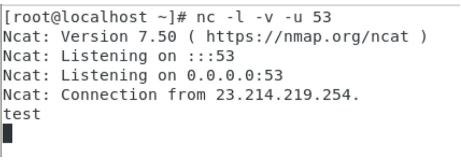
8.2 DNS name resolution requests sent to outside servers

After enabling DNS requests (port 53) we see that, using netcat, the messages are allowed to pass through the server.



```
[root@localhost tatianaalmeida]# nc -u 23.214.219.253 53
test
```

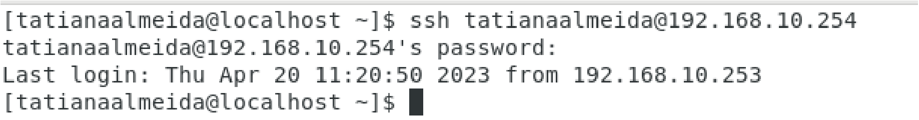
Figure 2: DNS request from router's interface to dns server



```
[root@localhost ~]# nc -l -v -u 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.254.
test
```

Figure 3: Request received in the dns server

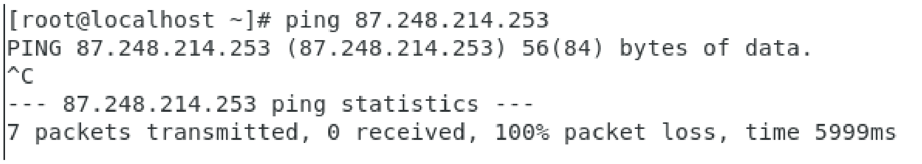
8.3 SSH connections to the router system, if originated at the internal network or at the VPN gateway



```
[tatianaalmeida@localhost ~]$ ssh tatianaalmeida@192.168.10.254
tatianaalmeida@192.168.10.254's password:
Last login: Thu Apr 20 11:20:50 2023 from 192.168.10.253
[tatianaalmeida@localhost ~]$
```

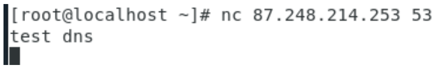
Figure 4: Internal connected via SSH to router

8.4 Drop all communications between networks



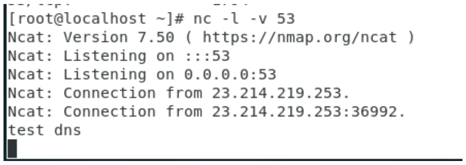
```
[root@localhost ~]# ping 87.248.214.253
PING 87.248.214.253 (87.248.214.253) 56(84) bytes of data.
^C
--- 87.248.214.253 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 5999ms
```

8.5 Domain name resolutions using the dns server



```
[root@localhost ~]# nc 87.248.214.253 53
test dns
```

Figure 5: Dns message from dns server

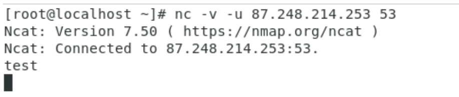


```
[root@localhost ~]# nc -l -v 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.253.
Ncat: Connection from 23.214.219.253:36992.
test dns
```

Figure 6: Message received in the router's interface

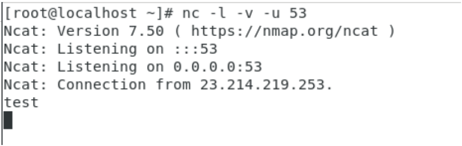
8.6 The dns server should be able to resolve names using DNS servers on the Internet (dns2 and also others)

EDEN



```
[root@localhost ~]# nc -v -u 87.248.214.253 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.253:53.
test
█
```

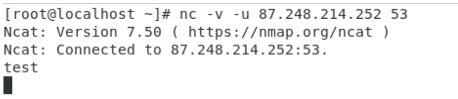
Figure 7: DMZ's interface sending messages to Eden



```
[root@localhost ~]# nc -l -v -u 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.253.
test
█
```

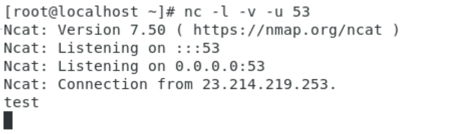
Figure 8: Internet's interface (eden) receiving messages from dmz

DNS2



```
[root@localhost ~]# nc -v -u 87.248.214.252 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.252:53.
test
█
```

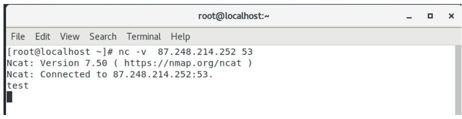
Figure 9: DMZ's interface sending messages to Dns2



```
[root@localhost ~]# nc -l -v -u 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.253.
test
█
```

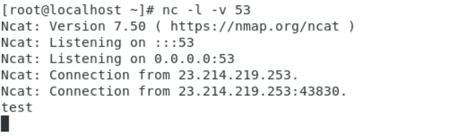
Figure 10: Internet's interface (dns2) receiving messages from DMZ

8.7 The dns and dns2 servers should be able to synchronize the contents of DNS zones



```
[root@localhost ~]# nc -v  87.248.214.252 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.252:53.
test
█
```

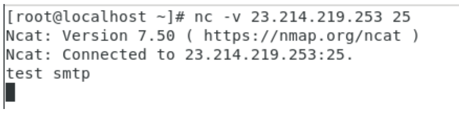
Figure 11: DMZ's interface sending messages to Dns2 using tcp



```
[root@localhost ~]# nc -l -v 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 23.214.219.253.
Ncat: Connection from 23.214.219.253:43830.
test
█
```

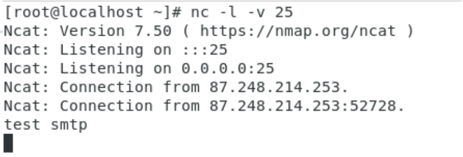
Figure 12: Internet's interface (dns2) receiving messages from dmz using tcp

8.8 SMTP connections to the smtp server



```
[root@localhost ~]# nc -v 23.214.219.253 25
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:25.
test smtp
█
```

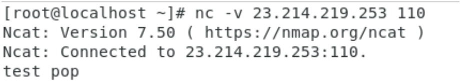
Figure 13: Internet's interface sends SMTP message



```
[root@localhost ~]# nc -l -v 25
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::25
Ncat: Listening on 0.0.0.0:25
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:52728.
test smtp
█
```

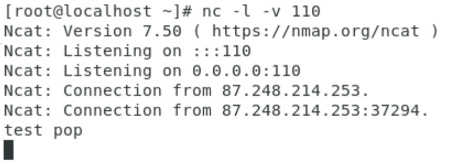
Figure 14: DMZ received SMTP message from Internet's interface

8.9 POP and IMAP connections to the mail server



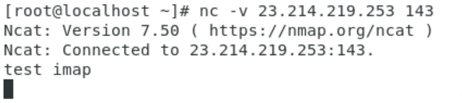
```
[root@localhost ~]# nc -v 23.214.219.253 110
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:110.
test pop
```

Figure 15: Internet's interface sends POP message



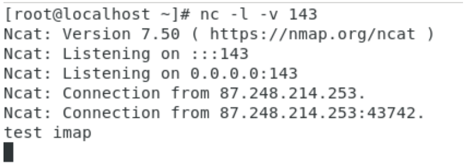
```
[root@localhost ~]# nc -l -v 110
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::110
Ncat: Listening on 0.0.0.0:110
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:37294.
test pop
```

Figure 16: DMZ received POP message from Internet's interface



```
[root@localhost ~]# nc -v 23.214.219.253 143
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:143.
test imap
```

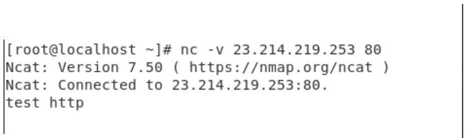
Figure 17: Internet's interface sends IMAP message



```
[root@localhost ~]# nc -l -v 143
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::143
Ncat: Listening on 0.0.0.0:143
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:43742.
test imap
```

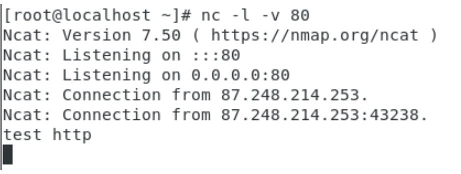
Figure 18: DMZ received IMAP message from internet's interface

8.10 HTTP and HTTPS connections to the www server



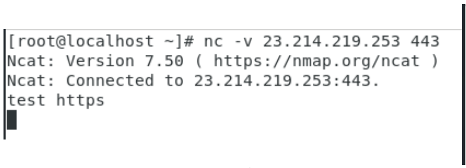
```
[root@localhost ~]# nc -v 23.214.219.253 80
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:80.
test http
```

Figure 19: Internet's interface sending message to DMZ interface



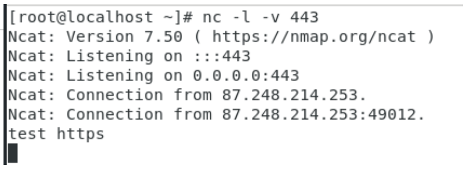
```
[root@localhost ~]# nc -l -v 80
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:43238.
test http
```

Figure 20: DMZ received message from Internet's interface



```
[root@localhost ~]# nc -v 23.214.219.253 443
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:443.
test https
```

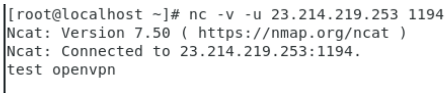
Figure 21: Internet's interface sending message to DMZ interface



```
[root@localhost ~]# nc -l -v 443
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:49012.
test https
```

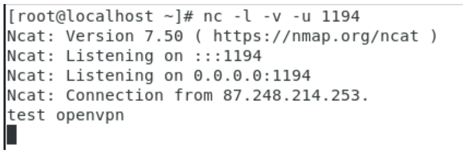
Figure 22: DMZ received message from Internet's interface

8.11 OpenVPN connections to the vpn-gw server



```
[root@localhost ~]# nc -v -u 23.214.219.253 1194
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:1194.
test openvpn
```

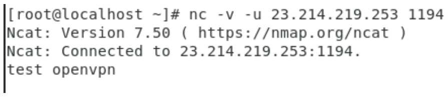
Figure 23: Internet's interface sending message to DMZ interface



```
[root@localhost ~]# nc -l -v -u 1194
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::1194
Ncat: Listening on 0.0.0.0:1194
Ncat: Connection from 87.248.214.253.
test openvpn
```

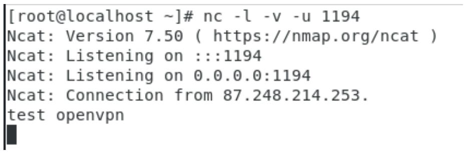
Figure 24: DMZ received message from Internet's interface

8.12 VPN clients connected to the gateway (vpn-gw) should be able to connect to all services in the Internal network



```
[root@localhost ~]# nc -v -u 23.214.219.253 1194
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 23.214.219.253:1194.
test openvpn
```

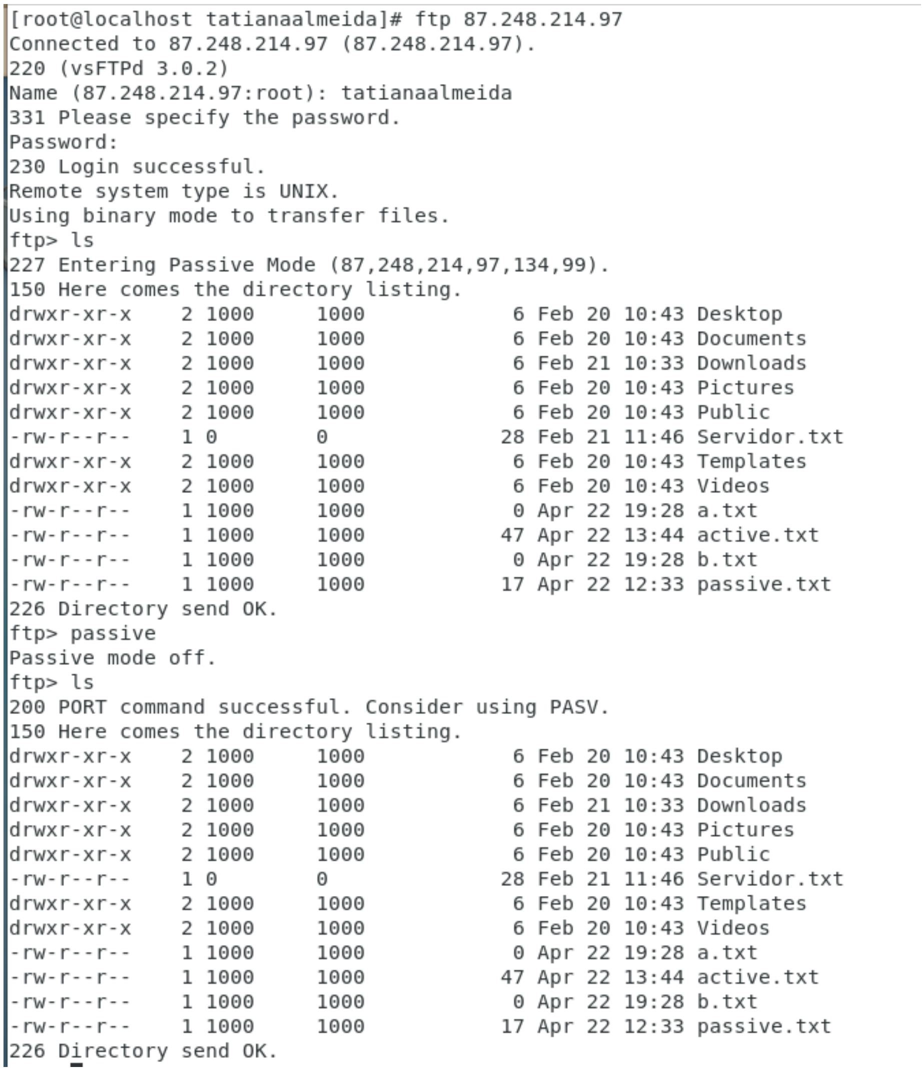
Figure 25: Internet's interface sending message to DMZ interface



```
[root@localhost ~]# nc -l -v -u 1194
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::1194
Ncat: Listening on 0.0.0.0:1194
Ncat: Connection from 87.248.214.253.
test openvpn
```

Figure 26: DMZ received message from Internet's interface

8.13 FTP connections (in passive and active modes) to the ftp server.

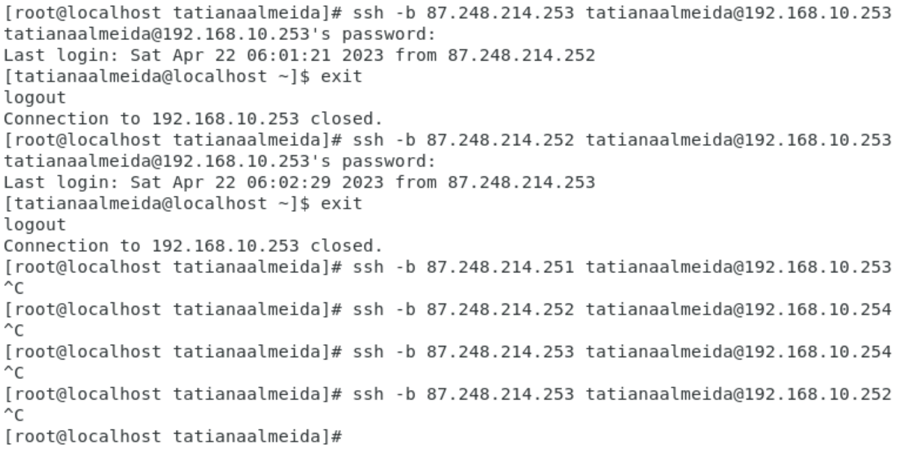


```
[root@localhost tatianaalmeida]# ftp 87.248.214.97
Connected to 87.248.214.97 (87.248.214.97).
220 (vsFTPd 3.0.2)
Name (87.248.214.97:root): tatianaalmeida
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (87,248,214,97,134,99).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Desktop
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Documents
drwxr-xr-x    2 1000     1000            6 Feb 21 10:33 Downloads
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Pictures
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Public
-rw-r--r--    1 0        0              28 Feb 21 11:46 Servidor.txt
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Templates
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Videos
-rw-r--r--    1 1000     1000            0 Apr 22 19:28 a.txt
-rw-r--r--    1 1000     1000           47 Apr 22 13:44 active.txt
-rw-r--r--    1 1000     1000            0 Apr 22 19:28 b.txt
-rw-r--r--    1 1000     1000           17 Apr 22 12:33 passive.txt
226 Directory send OK.
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 Feb 20 10:43 Desktop
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Documents
drwxr-xr-x    2 1000     1000            6 Feb 21 10:33 Downloads
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Pictures
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Public
-rw-r--r--    1 0        0              28 Feb 21 11:46 Servidor.txt
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Templates
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Videos
-rw-r--r--    1 1000     1000            0 Apr 22 19:28 a.txt
-rw-r--r--    1 1000     1000           47 Apr 22 13:44 active.txt
-rw-r--r--    1 1000     1000            0 Apr 22 19:28 b.txt
-rw-r--r--    1 1000     1000           17 Apr 22 12:33 passive.txt
226 Directory send OK.
```

Figure 27: figure

FTP connection from internet to router's external interface

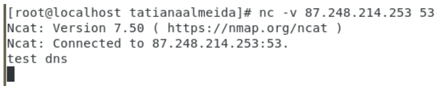
8.14 SSH connections to the datastore server, but only if originated at the eden or dns2 servers.



```
[root@localhost tatianaalmeida]# ssh -b 87.248.214.253 tatianaalmeida@192.168.10.253
tatianaalmeida@192.168.10.253's password:
Last login: Sat Apr 22 06:01:21 2023 from 87.248.214.252
[tatianaalmeida@localhost ~]$ exit
logout
Connection to 192.168.10.253 closed.
[root@localhost tatianaalmeida]# ssh -b 87.248.214.252 tatianaalmeida@192.168.10.253
tatianaalmeida@192.168.10.253's password:
Last login: Sat Apr 22 06:02:29 2023 from 87.248.214.253
[tatianaalmeida@localhost ~]$ exit
logout
Connection to 192.168.10.253 closed.
[root@localhost tatianaalmeida]# ssh -b 87.248.214.251 tatianaalmeida@192.168.10.253
^C
[root@localhost tatianaalmeida]# ssh -b 87.248.214.252 tatianaalmeida@192.168.10.254
^C
[root@localhost tatianaalmeida]# ssh -b 87.248.214.253 tatianaalmeida@192.168.10.254
^C
[root@localhost tatianaalmeida]# ssh -b 87.248.214.253 tatianaalmeida@192.168.10.252
^C
[root@localhost tatianaalmeida]#
```

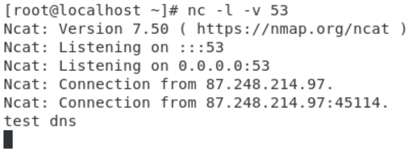
Figure 28: Internal connected via SSH to router

8.15 Domain name resolutions using DNS



```
[root@localhost tatianaalmeida]# nc -v 87.248.214.253 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.253:53.
test dns
```

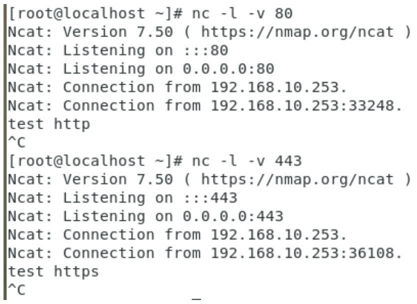
Figure 29: Internal network's interface sending messages to external network



```
[root@localhost ~]# nc -l -v 53
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::53
Ncat: Listening on 0.0.0.0:53
Ncat: Connection from 87.248.214.97.
Ncat: Connection from 87.248.214.97:45114.
test dns
```

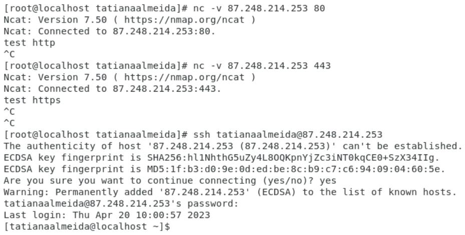
Figure 30: External network's interface receiving messages from external network

8.16 HTTP, HTTPS and SSH connections



```
[root@localhost ~]# nc -l -v 80
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 192.168.10.253.
Ncat: Connection from 192.168.10.253:33248.
test http
^C
[root@localhost ~]# nc -l -v 443
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 192.168.10.253.
Ncat: Connection from 192.168.10.253:36108.
test https
^C
```

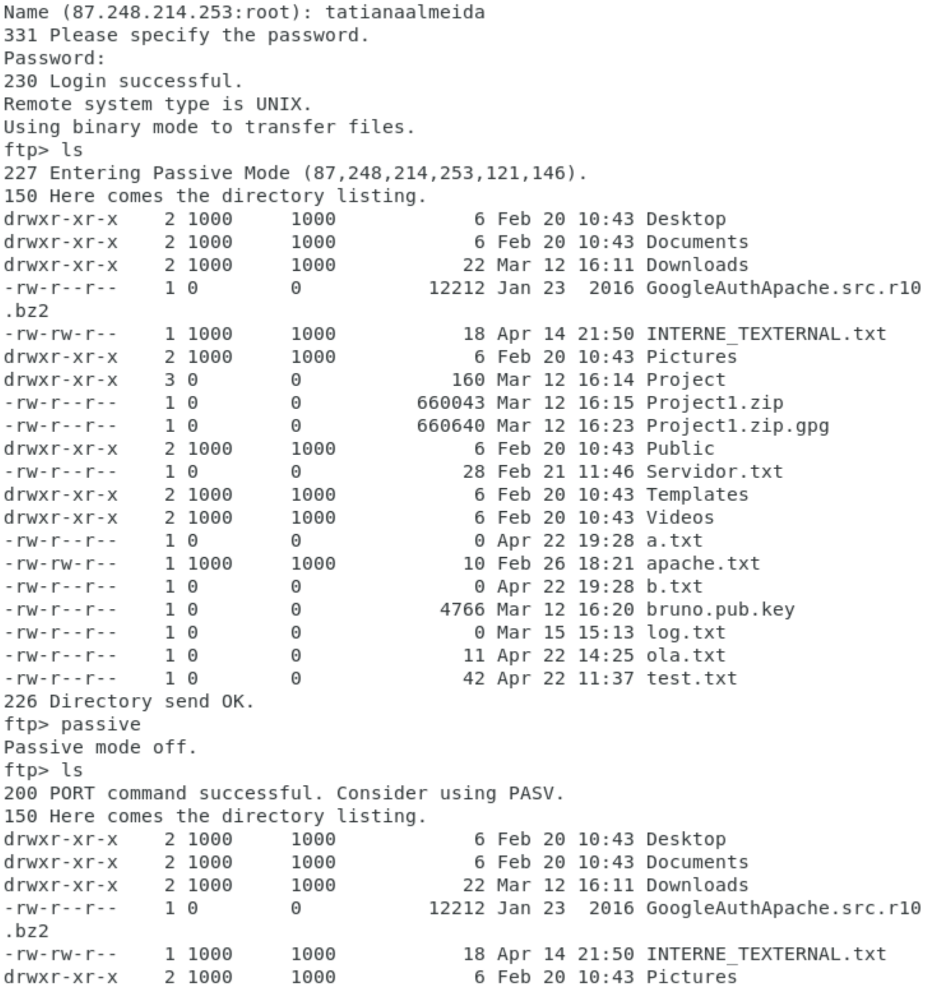
Figure 31: Connections http, https from external network



```
[root@localhost tatianaalmeida]# nc -v 87.248.214.253 80
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.253:80.
test http
^C
[root@localhost tatianaalmeida]# nc -v 87.248.214.253 443
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Connected to 87.248.214.253:443.
test https
^C
^C
[root@localhost tatianaalmeida]# ssh tatianaalmeida@87.248.214.253
The authenticity of host '87.248.214.253 (87.248.214.253)' can't be established.
ECDSA key fingerprint is SHA256:hl1NhthG5uZy4L8OQKpnYjZc3iNT0kqCE0+SzX34IIg.
ECDSA key fingerprint is MD5:1f:b3:d0:9e:0d:ed:be:8c:b9:c7:c6:94:09:04:60:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '87.248.214.253' (ECDSA) to the list of known hosts.
tatianaalmeida@87.248.214.253's password:
Last login: Thu Apr 20 10:00:57 2023
[tatianaalmeida@localhost ~]$
```

Figure 32: Connections http, https and ssh from internal network

8.17 FTP connections (in passive and active modes) to external FTP servers

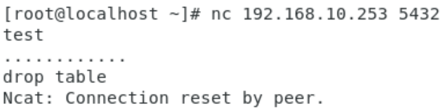


```
Name (87.248.214.253:root): tatianaalmeida
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (87,248,214,253,121,146).
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Desktop
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Documents
drwxr-xr-x    2 1000     1000           22 Mar 12 16:11 Downloads
-rw-r--r--    1 0        0           12212 Jan 23  2016 GoogleAuthApache.src.r10
.bz2
-rw-rw-r--    1 1000     1000           18 Apr 14 21:50 INTERNE_TEXTERNAL.txt
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Pictures
drwxr-xr-x    3 0        0             160 Mar 12 16:14 Project
-rw-r--r--    1 0        0          660043 Mar 12 16:15 Project1.zip
-rw-r--r--    1 0        0          660640 Mar 12 16:23 Project1.zip.gpg
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Public
-rw-r--r--    1 0        0              28 Feb 21 11:46 Servidor.txt
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Templates
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Videos
-rw-r--r--    1 0        0               0 Apr 22 19:28 a.txt
-rw-rw-r--    1 1000     1000           10 Feb 26 18:21 apache.txt
-rw-r--r--    1 0        0               0 Apr 22 19:28 b.txt
-rw-r--r--    1 0        0            4766 Mar 12 16:20 bruno.pub.key
-rw-r--r--    1 0        0               0 Mar 15 15:13 log.txt
-rw-r--r--    1 0        0              11 Apr 22 14:25 ola.txt
-rw-r--r--    1 0        0              42 Apr 22 11:37 test.txt
226 Directory send OK.
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 Feb 20 10:43 Desktop
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Documents
drwxr-xr-x    2 1000     1000           22 Mar 12 16:11 Downloads
-rw-r--r--    1 0        0           12212 Jan 23  2016 GoogleAuthApache.src.r10
.bz2
-rw-rw-r--    1 1000     1000           18 Apr 14 21:50 INTERNE_TEXTERNAL.txt
drwxr-xr-x    2 1000     1000            6 Feb 20 10:43 Pictures
```

Figure 33: FTP connection from internet to router’s external interface

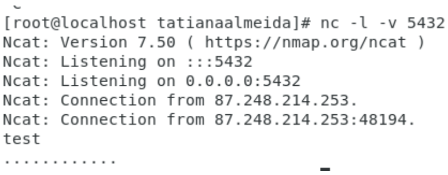
8.18 SQL injection

Rule: drop tcp any any -> 192.168.10.253 any (msg:”SQL injection detected”; content:”drop table”; sid:100001; rev:1;)



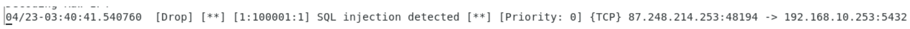
```
[root@localhost ~]# nc 192.168.10.253 5432
test
............
drop table
Ncat: Connection reset by peer.
```

Figure 34: Sending sql attack from internet to internal network



```
[root@localhost tatianaalmeida]# nc -l -v 5432
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::5432
Ncat: Listening on 0.0.0.0:5432
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:48194.
test
............
```

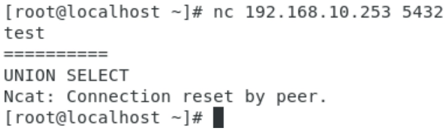
Figure 35: Internal network receiving sql attack from internet



```
04/23-03:40:41.540760  [Drop] [**] [1:100001:1] SQL injection detected [**] [Priority: 0] {TCP} 87.248.214.253:48194 -> 192.168.10.253:5432
```

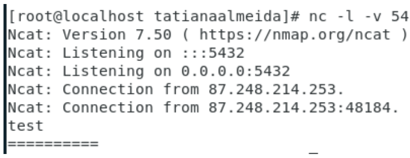
Figure 36: Snort message

Rule: drop tcp any any -> 192.168.10.253 any (msg:"UNION SQL injection detected"; content:"UNION SELECT"; sid:100002; rev:1;)



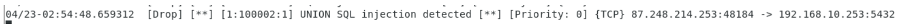
```
[root@localhost ~]# nc 192.168.10.253 5432
test
==========
UNION SELECT
Ncat: Connection reset by peer.
[root@localhost ~]# 
```

Figure 37: Sending sql attack from internet to internal network



```
[root@localhost tatianaalmeida]# nc -l -v 5432
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::5432
Ncat: Listening on 0.0.0.0:5432
Ncat: Connection from 87.248.214.253.
Ncat: Connection from 87.248.214.253:48184.
test
==========
```

Figure 38: Internal network receiving sql attack from internet

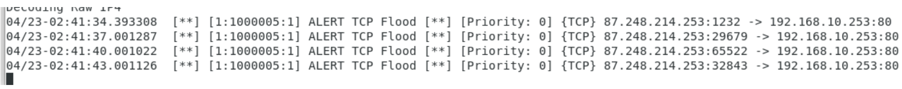


```
04/23-02:54:48.659312  [Drop] [**] [1:100002:1] UNION SQL injection detected [**] [Priority: 0] {TCP} 87.248.214.253:48184 -> 192.168.10.253:5432
```

Figure 39: Snort message

8.19 DoS (Denial of Service) attack

Rule: alert tcp any any -> 192.168.10.253 any (msg:"ALERT TCP Flood"; threshold:type both, track by_dst, count 15, seconds 3; sid:1000005; rev:1;) To test, the command "hping3 -S -p 80 –flood 192.168.10.253" was ran on Internet's terminal

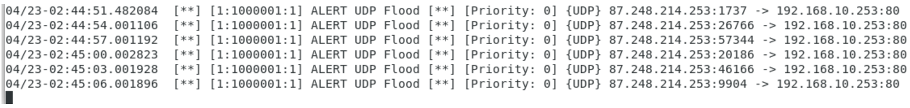


```
Decoding Raw IP4
04/23-02:41:34.393308  [**] [1:1000005:1] ALERT TCP Flood [**] [Priority: 0] {TCP} 87.248.214.253:1232 -> 192.168.10.253:80
04/23-02:41:37.001287  [**] [1:1000005:1] ALERT TCP Flood [**] [Priority: 0] {TCP} 87.248.214.253:29679 -> 192.168.10.253:80
04/23-02:41:40.001022  [**] [1:1000005:1] ALERT TCP Flood [**] [Priority: 0] {TCP} 87.248.214.253:65522 -> 192.168.10.253:80
04/23-02:41:43.001126  [**] [1:1000005:1] ALERT TCP Flood [**] [Priority: 0] {TCP} 87.248.214.253:32843 -> 192.168.10.253:80
```

Figure 40: Snort message

Rule: alert udp any any -> 192.168.10.253 any (msg:"ALERT UDP Flood"; threshold:type both, track by_dst, count 15, seconds 3; sid:1000001; rev:1;)

To test, the command "hping3 -S -p 80 -2 –flood 192.168.10.253" was ran on Internet's terminal

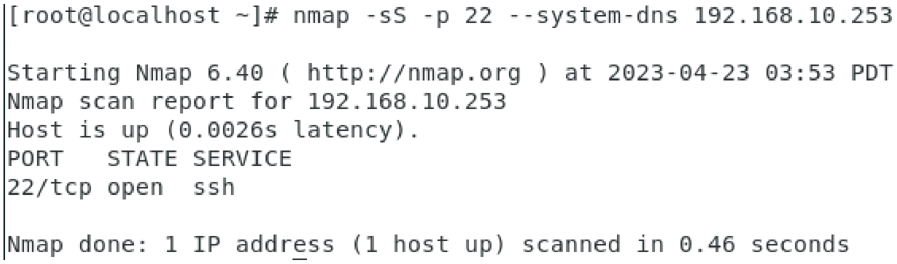


```
04/23-02:44:51.482084  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:1737 -> 192.168.10.253:80
04/23-02:44:54.001106  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:26766 -> 192.168.10.253:80
04/23-02:44:57.001192  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:57344 -> 192.168.10.253:80
04/23-02:45:00.002823  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:20186 -> 192.168.10.253:80
04/23-02:45:03.001928  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:46166 -> 192.168.10.253:80
04/23-02:45:06.001896  [**] [1:1000001:1] ALERT UDP Flood [**] [Priority: 0] {UDP} 87.248.214.253:9904 -> 192.168.10.253:80
```

Figure 41: Snort message

8.20 OS fingerprinting attempts

Rule: alert tcp any any -> any any (msg:"Nmap OS Fingerprinting Detected"; flags:S;reference:url,nmap.org; sid:1000003; rev:1;)

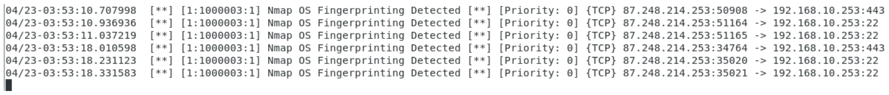


```
[root@localhost ~]# nmap -sS -p 22 --system-dns 192.168.10.253

Starting Nmap 6.40 ( http://nmap.org ) at 2023-04-23 03:53 PDT
Nmap scan report for 192.168.10.253
Host is up (0.0026s latency).
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

Figure 42: Sending OS fingerprinting attack



```
04/23-03:53:10.707998  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:50908 -> 192.168.10.253:443
04/23-03:53:10.936936  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:51164 -> 192.168.10.253:22
04/23-03:53:11.037219  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:51165 -> 192.168.10.253:22
04/23-03:53:18.010598  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:34764 -> 192.168.10.253:443
04/23-03:53:18.231123  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:35020 -> 192.168.10.253:22
04/23-03:53:18.331583  [**] [1:1000003:1] Nmap OS Fingerprinting Detected [**] [Priority: 0] {TCP} 87.248.214.253:35021 -> 192.168.10.253:22
```

Figure 43: Snort message

9 Conclusion

With this project we learned to work with ip table rules and learned how to detect attacks and sql injections with the use of Snort. As was shown, a series of tests validated our implementation, meaning we successfully implemented a basic firewall.