Report

Assignment 1 Open Source Firewalls Jan 2024

Student Name	Student Number	email address
Sebastian Konefal	**b00168561**	**b00168561@mytudublin.ie**

TU765
Digital Forensics & Cyber Security

DFCS H3015 - Network Security [1114.VLE-BN]

Date

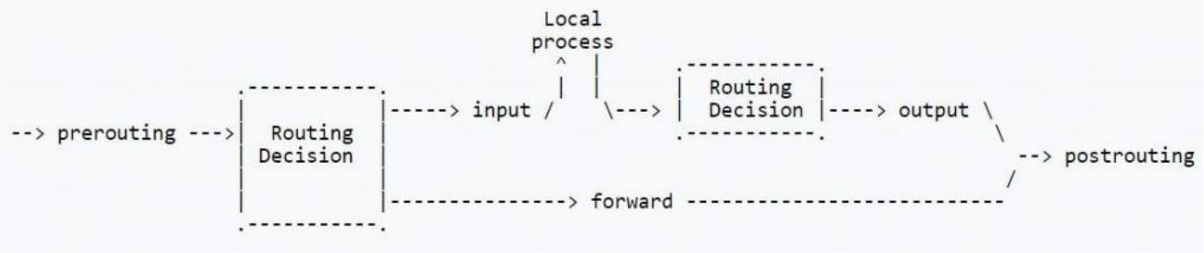
03/03/2024

Table of Contents

Introduction

Nftables is a subsystem of the Linux kernel that provides a set of functions for filtering and network manipulation packets. Nftables replaced the packet filtering system - iptables that was used by Linux for many years. Compared to iptables, nftables is engineered to use resources more efficiently, it can manage larger volumes of network traffic while putting a lower load on the system. Creating and maintaining packet filtering rules is made simpler by more flexible and easy-to-understand configuration syntax.[1] There are no pre-defined tables or chains when using nftables. Each table needs to be defined and only consists of the objects (chains, sets, maps, flowtables and stateful objects). The nftables rule is composed of zero or more expressions followed by one or more statements. Linear evaluation from left to right applies to multiple expressions - every expression is evaluated one after another. When we get to the final expression, the statements in the rule are executed as the packet matches all of the expressions in the rule. Linear evaluation from left to right also applies to multiple statements - this means that a single rule can take multiple actions by using multiple statements.[2]

Nftables has been available since Linux kernel 3.13. It supports a variety of address families such as ip - IPv4, ip6 - IPv6, arp - Address Resolution Protocol (ARP), bridge - processing for bridged packets and netdev - Netdev address family. The address family to which the rule will apply is determined by the nftables processing architecture overall. Nftables makes use of one or more tables, each of which has chains that carry the processing rules. Those processing rules consist of statements like drop, queue, and continue and they are created by expressions such as the address, interface, ports, or other data in the packet. In Linux, certain address families have hooks that allow nftables to inspect packets as they traverse the network stack.[3] Hooks are triggered when incoming or outgoing packet enters a node with active nftables. Rules linked to these hooks are allowed to interact with network traffic by the Linux kernel. Netfilter contains five hooks including prerouting, input, output, postrouting, forward, and ingress.[4]



```
                                            Local
                                           process
                                             ^  |      .-----------.
                    .-----------.            |  |      |  Routing  |
                    |           |-----> input /  \---> |  Decision |----> output \
--> prerouting ---> |  Routing  |                      .-----------.              \
                    |  Decision |                                                  --> postrouting
                    |           |                                                 /
                    |           |---------------> forward ---------------------------
                    .-----------.
```

Fig. 1 Easillyy (n.d.) A comprehensive guide to nftables [online graphic]

The main functions of nftables are:

- packet filtering - Nftables enables to set up rules to filter network packets according to parameters including IP addresses, ports, protocols, and network interfaces, allowing the management of packets that can enter, exit, or pass through the system.

[1] Shirvar A., (2020)

[2] Wiki, (2021)

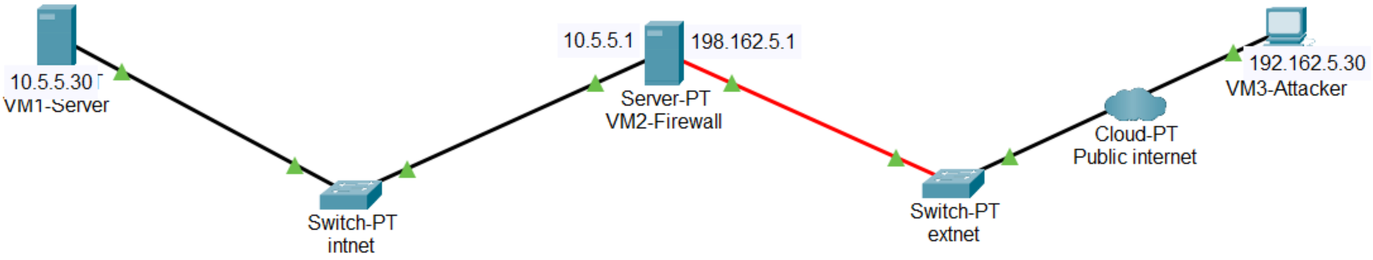
[3] Suehring S., 2015, 83-93

[4] Easillyy, (n.d.)

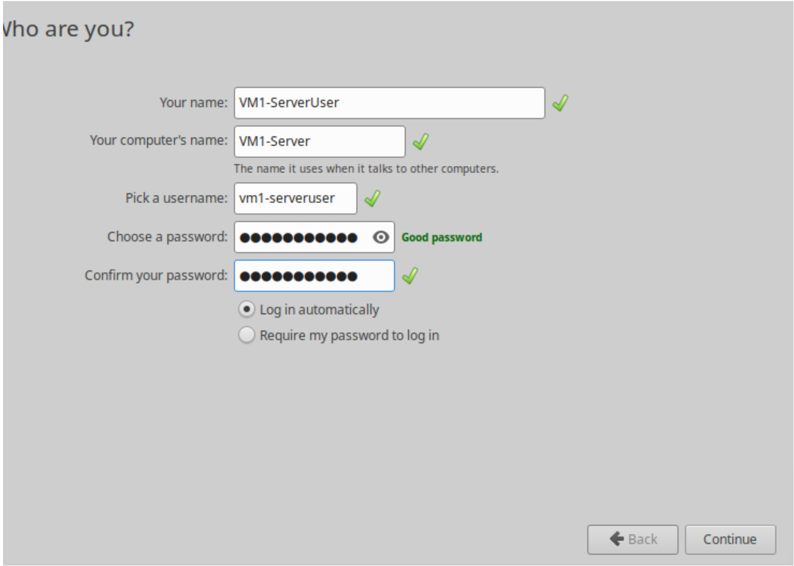
- Network Address Translation (NAT) - Network address translation (NAT) is a function of nftables that allows change of the IP addresses and ports of network packets which can be used for port translation (masquerading) and static network address translation (SNAT and DNAT).
- load balancing - It can improve the availability and scalability of network services by dividing incoming traffic across several backend servers providing load balancing.
- packet classification (QoS) - By implementing Quality of Service (QoS), nftables ensures adequate performance by prioritisation of some traffic types above others.
- packet logging - Nftables allows tracking traffic traversing the system and logging important events therefore it is very useful for network troubleshooting, auditing and monitoring.
- packet manipulation - Advanced packet manipulation features enable the modification of particular fields in network packets (such as IP addresses, ports, VLAN tags, and others) as they traverse the system.[5]

1. NFtables development

1.1. Topology



Installing xUbuntu on each VM, example below:



Vho are you?

Your name: VM1-ServerUser ✓

Your computer's name: VM1-Server ✓

The name it uses when it talks to other computers.

Pick a username: vm1-serveruser ✓

Choose a password: ●●●●●●●●●●● Good password

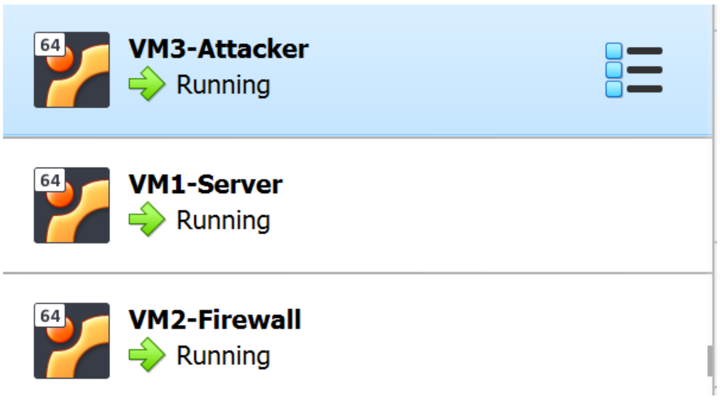
Confirm your password: ●●●●●●●●●●● ✓

● Log in automatically

○ Require my password to log in

[5] SW Team, (2024)

Created 3 VMs in total:



VM3-Attacker
Running

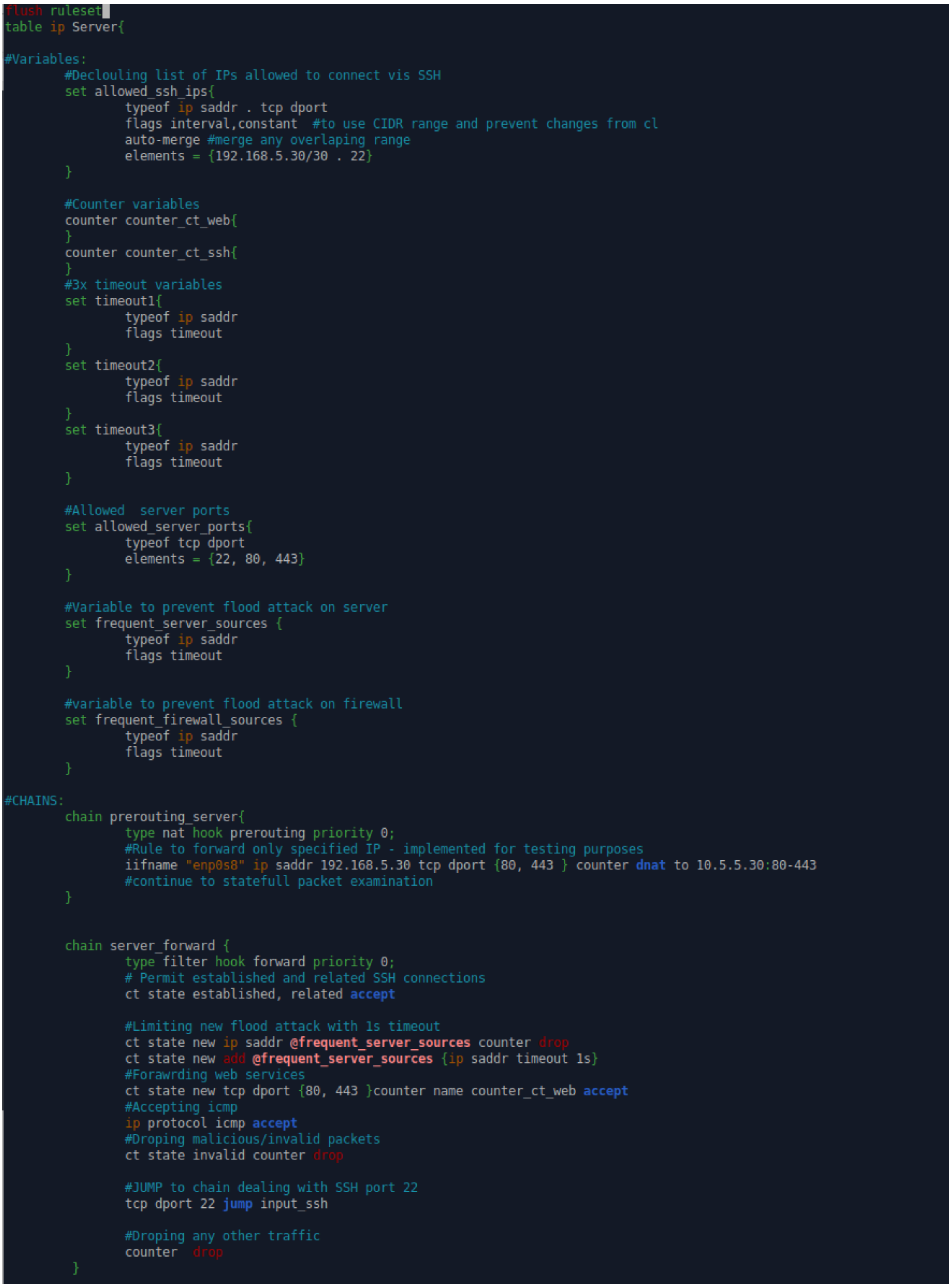
VM1-Server
Running

VM2-Firewall
Running

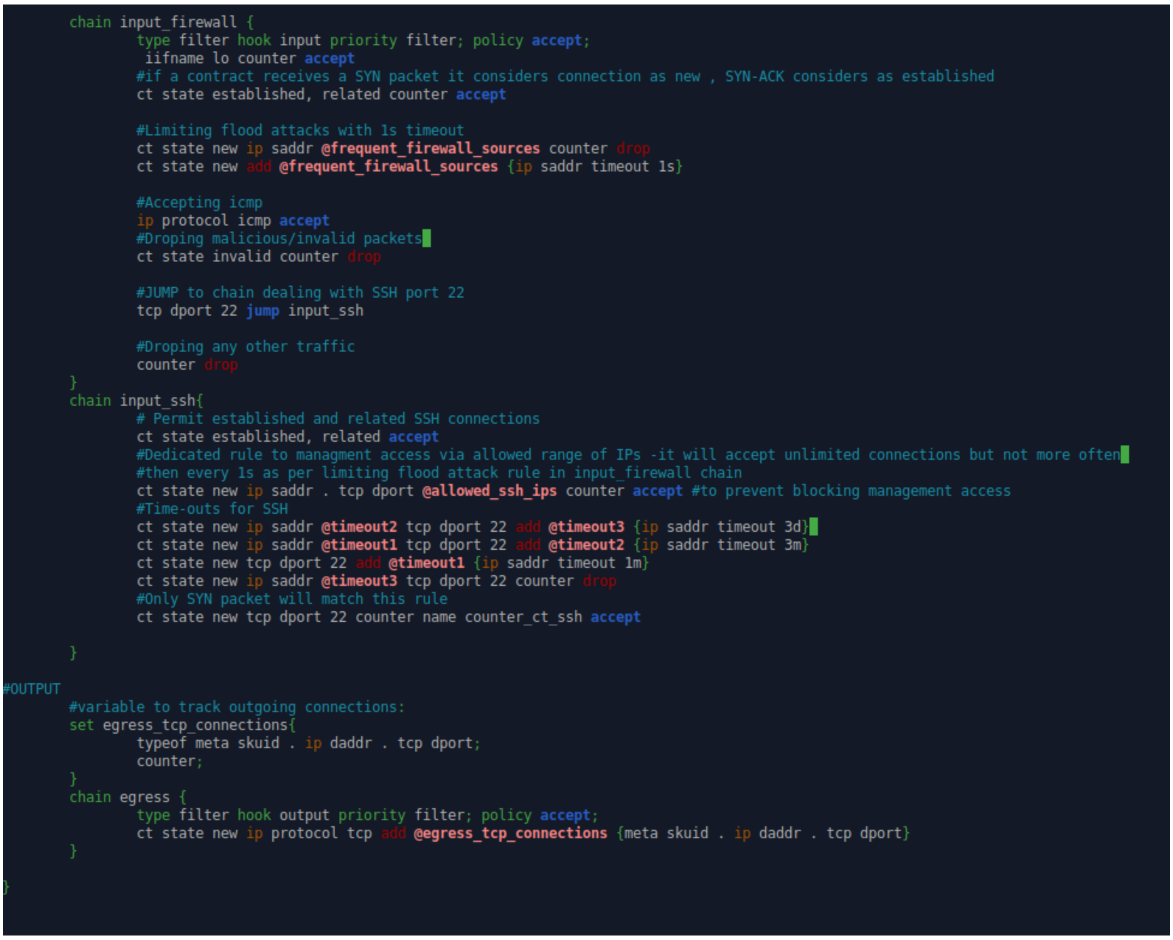
1.2. My NFTable

Editable text at the end of the report

Possible further development: Honeypots, Logging and Analysis, Geo-IP Filtering, Load Balancing



```
flush ruleset
table ip Server{

#Variables:
        #Declouling list of IPs allowed to connect vis SSH
        set allowed_ssh_ips{
                typeof ip saddr . tcp dport
                flags interval,constant  #to use CIDR range and prevent changes from cl
                auto-merge #merge any overlaping range
                elements = {192.168.5.30/30 . 22}
        }

        #Counter variables
        counter counter_ct_web{
        }
        counter counter_ct_ssh{
        }
        #3x timeout variables
        set timeout1{
                typeof ip saddr
                flags timeout
        }
        set timeout2{
                typeof ip saddr
                flags timeout
        }
        set timeout3{
                typeof ip saddr
                flags timeout
        }

        #Allowed  server ports
        set allowed_server_ports{
                typeof tcp dport
                elements = {22, 80, 443}
        }

        #Variable to prevent flood attack on server
        set frequent_server_sources {
                typeof ip saddr
                flags timeout
        }

        #variable to prevent flood attack on firewall
        set frequent_firewall_sources {
                typeof ip saddr
                flags timeout
        }

#CHAINS:
        chain prerouting_server{
                type nat hook prerouting priority 0;
                #Rule to forward only specified IP - implemented for testing purposes
                iifname "enp0s8" ip saddr 192.168.5.30 tcp dport {80, 443 } counter dnat to 10.5.5.30:80-443
                #continue to statefull packet examination
        }


        chain server_forward {
                type filter hook forward priority 0;
                # Permit established and related SSH connections
                ct state established, related accept

                #Limiting new flood attack with 1s timeout
                ct state new ip saddr @frequent_server_sources counter drop
                ct state new add @frequent_server_sources {ip saddr timeout 1s}
                #Forawrding web services
                ct state new tcp dport {80, 443 }counter name counter_ct_web accept
                #Accepting icmp
                ip protocol icmp accept
                #Droping malicious/invalid packets
                ct state invalid counter drop

                #JUMP to chain dealing with SSH port 22
                tcp dport 22 jump input_ssh

                #Droping any other traffic
                counter  drop
         }
```



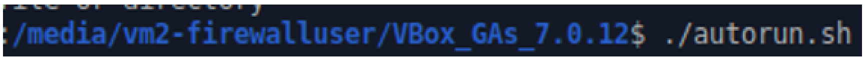
```
        chain input_firewall {
                type filter hook input priority filter; policy accept;
                 iifname lo counter accept
                #if a contract receives a SYN packet it considers connection as new , SYN-ACK considers as established
                ct state established, related counter accept

                #Limiting flood attacks with 1s timeout
                ct state new ip saddr @frequent_firewall_sources counter drop
                ct state new add @frequent_firewall_sources {ip saddr timeout 1s}

                #Accepting icmp
                ip protocol icmp accept
                #Droping malicious/invalid packets
                ct state invalid counter drop

                #JUMP to chain dealing with SSH port 22
                tcp dport 22 jump input_ssh

                #Droping any other traffic
                counter drop
        }
        chain input_ssh{
                # Permit established and related SSH connections
                ct state established, related accept
                #Dedicated rule to managment access via allowed range of IPs -it will accept unlimited connections but not more often
                #then every 1s as per limiting flood attack rule in input_firewall chain
                ct state new ip saddr . tcp dport @allowed_ssh_ips counter accept #to prevent blocking management access
                #Time-outs for SSH
                ct state new ip saddr @timeout2 tcp dport 22 add @timeout3 {ip saddr timeout 3d}
                ct state new ip saddr @timeout1 tcp dport 22 add @timeout2 {ip saddr timeout 3m}
                ct state new tcp dport 22 add @timeout1 {ip saddr timeout 1m}
                ct state new ip saddr @timeout3 tcp dport 22 counter drop
                #Only SYN packet will match this rule
                ct state new tcp dport 22 counter name counter_ct_ssh accept

        }

#OUTPUT
        #variable to track outgoing connections:
        set egress_tcp_connections{
                typeof meta skuid . ip daddr . tcp dport;
                counter;
        }
        chain egress {
                type filter hook output priority filter; policy accept;
                ct state new ip protocol tcp add @egress_tcp_connections {meta skuid . ip daddr . tcp dport}
        }

}
```

[6][7][8][9]

1.3. Adjusting software and environment

1.3.1. Global settings on all VMs:

Installed Guest Additions for better user experience:



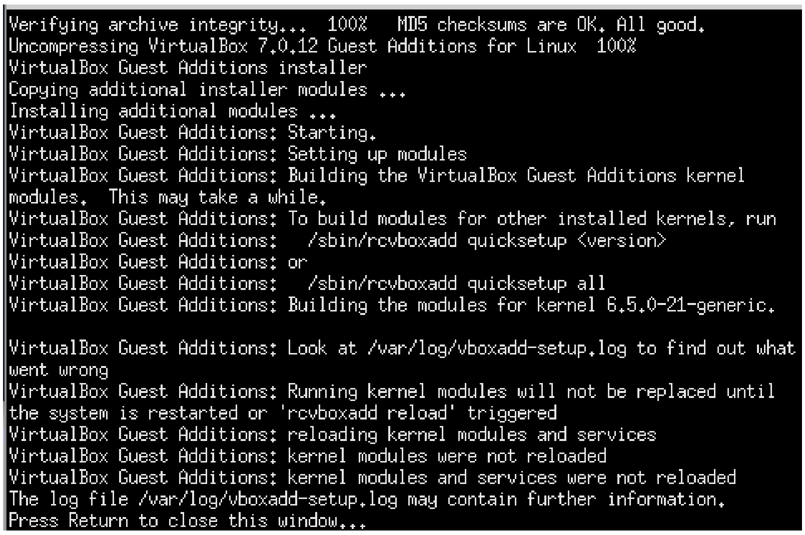
```
ile or directory
:/media/vm2-firewalluser/VBox_GAs_7.0.12$ ./autorun.sh
```

[6] LinuxCloudHacks (2023) Mastering Nftables Sets: A Comprehensive Guide

[7] LinuxCloudHacks (2023) Protecting Incoming Traffic with Nftables

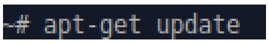
[8] LinuxCloudHacks (2023) Unleashing the Power of Nftables Chains and Verdict Maps

[9] Wiki (2018)

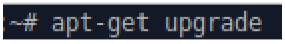


```
Verifying archive integrity...  100%   MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.0.12 Guest Additions for Linux  100%
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Setting up modules
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules.  This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 6.5.0-21-generic.

VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what
went wrong
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted or 'rcvboxadd reload' triggered
VirtualBox Guest Additions: reloading kernel modules and services
VirtualBox Guest Additions: kernel modules were not reloaded
VirtualBox Guest Additions: kernel modules and services were not reloaded
The log file /var/log/vboxadd-setup.log may contain further information.
Press Return to close this window...
```

Updating and upgrading using additional NAT interface:

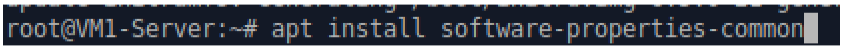


```
~# apt-get update
```

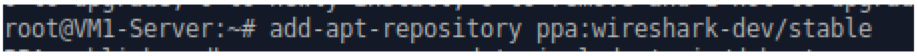


```
~# apt-get upgrade
```

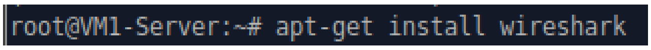
Installing Wireshark:



```
root@VM1-Server:~# apt install software-properties-common
```



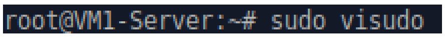
```
root@VM1-Server:~# add-apt-repository ppa:wireshark-dev/stable
```



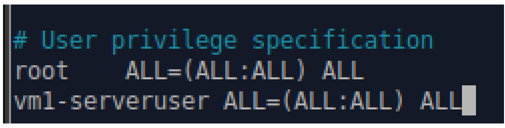
```
root@VM1-Server:~# apt-get install wireshark
```

[10]

Adding user to sudoers:

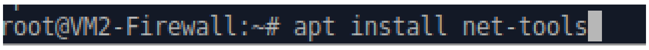


```
root@VM1-Server:~# sudo visudo
```



```
# User privilege specification
root     ALL=(ALL:ALL) ALL
vm1-serveruser ALL=(ALL:ALL) ALL
```

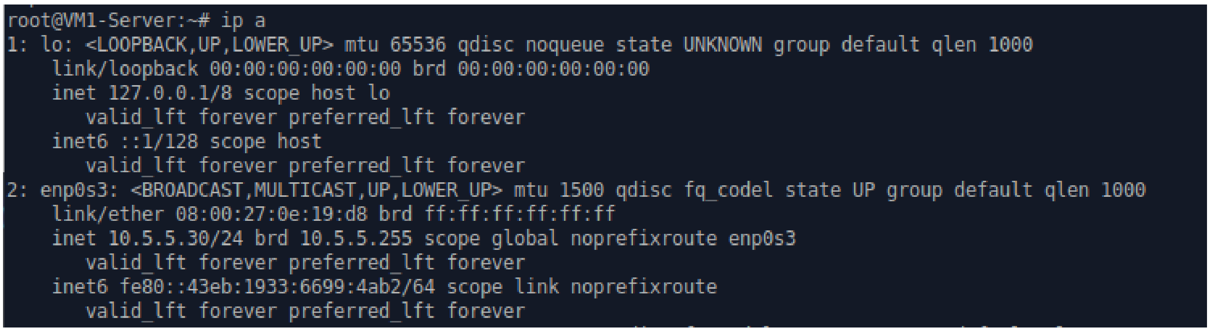
[11]



```
root@VM2-Firewall:~# apt install net-tools
```

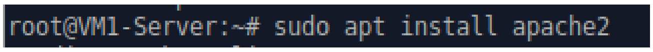
[10] ZacsTech (2022)
[11] ZacsTech (2022)

2.3.2. VM1-Server settings

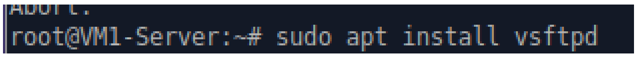


```
root@VM1-Server:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0e:19:d8 brd ff:ff:ff:ff:ff:ff
    inet 10.5.5.30/24 brd 10.5.5.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::43eb:1933:6699:4ab2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

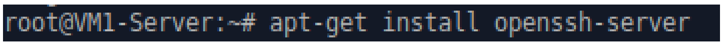
Installing services:



```
root@VM1-Server:~# sudo apt install apache2
```

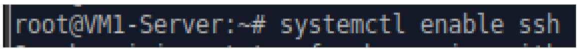


```
root@VM1-Server:~# sudo apt install vsftpd
```

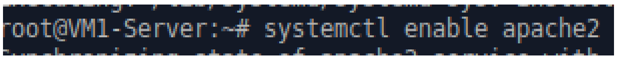


```
root@VM1-Server:~# apt-get install openssh-server
```

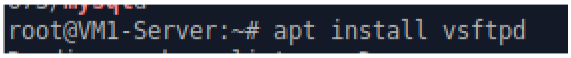
Enabling opening ports on boot up:



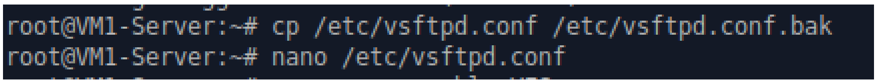
```
root@VM1-Server:~# systemctl enable ssh
```



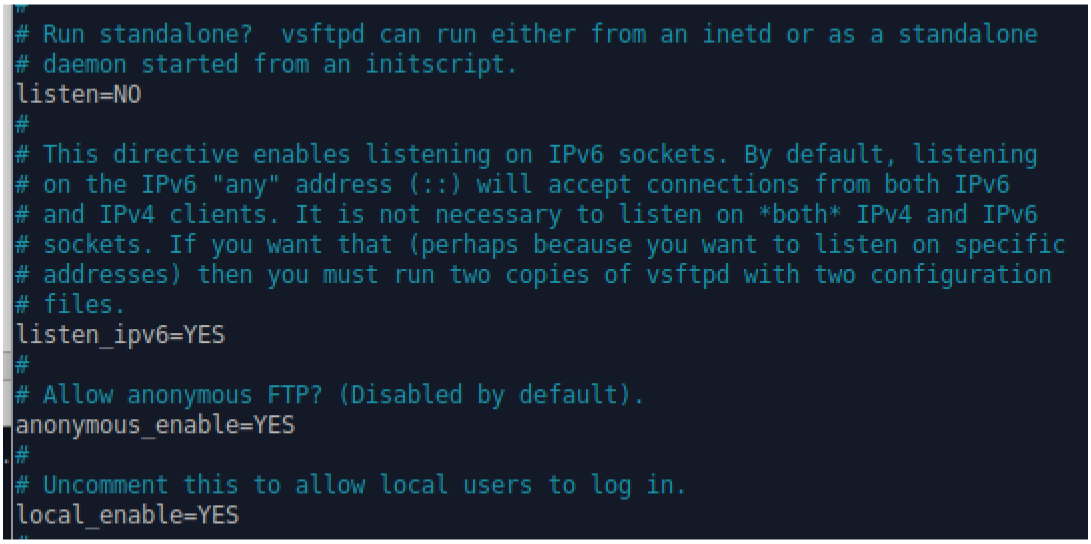
```
oot@VM1-Server:~# systemctl enable apache2
```



```
root@VM1-Server:~# apt install vsftpd
```

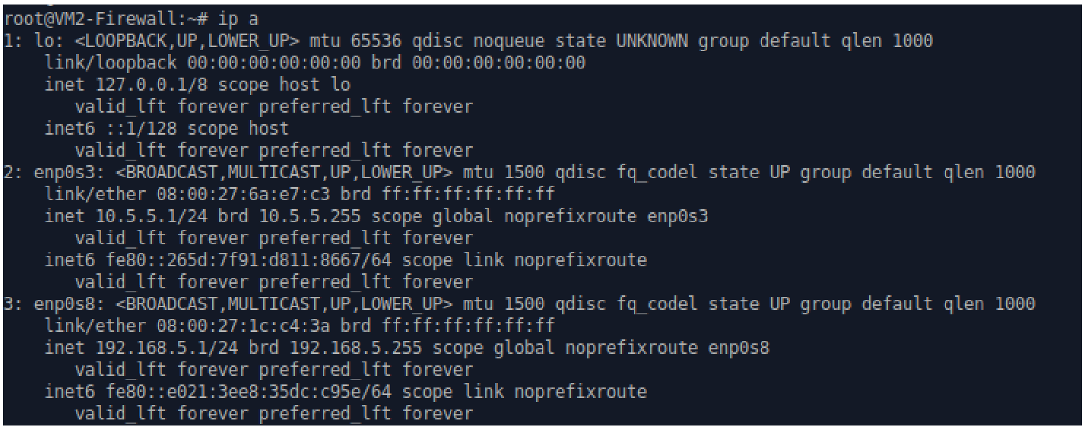


```
root@VM1-Server:~# cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
root@VM1-Server:~# nano /etc/vsftpd.conf
```



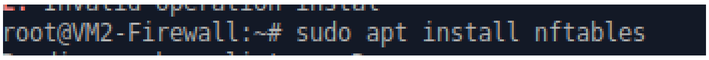
```
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
```

2.3.3. VM2- Firewall settings

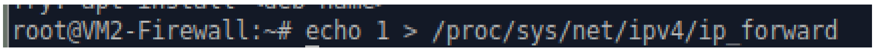


```
root@VM2-Firewall:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6a:e7:c3 brd ff:ff:ff:ff:ff:ff
    inet 10.5.5.1/24 brd 10.5.5.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::265d:7f91:d811:8667/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:c4:3a brd ff:ff:ff:ff:ff:ff
    inet 192.168.5.1/24 brd 192.168.5.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::e021:3ee8:35dc:c95e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

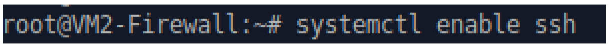
Installing and enabling nftables on VM-2Firewall and ssh, fpt services and port forwarding, examples below:



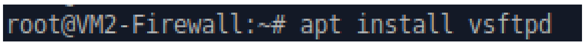
```
root@VM2-Firewall:~# sudo apt install nftables
```



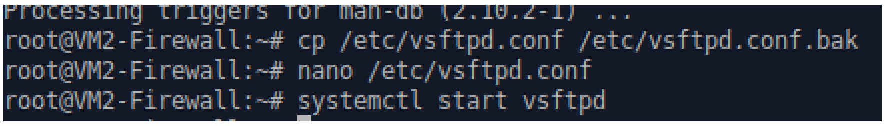
```
root@VM2-Firewall:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```



```
root@VM2-Firewall:~# systemctl enable ssh
```

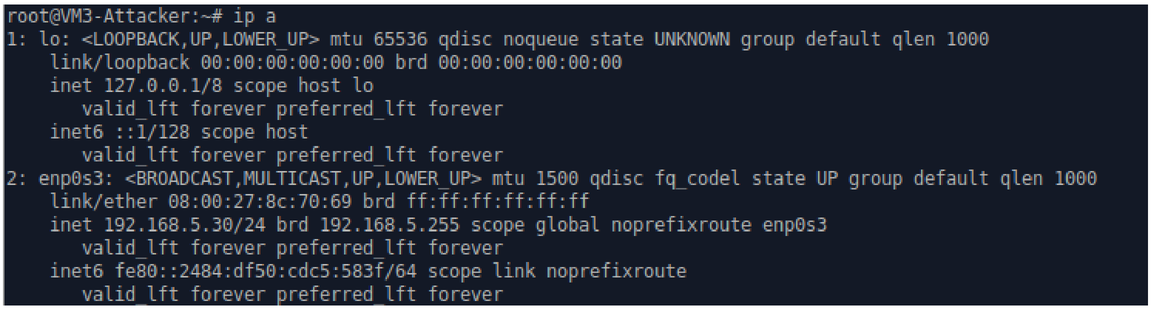


```
root@VM2-Firewall:~# apt install vsftpd
```



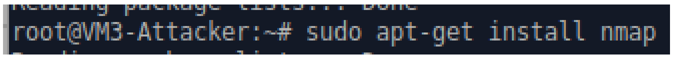
```
Processing triggers for man-db (2.10.2-1) ...
root@VM2-Firewall:~# cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
root@VM2-Firewall:~# nano /etc/vsftpd.conf
root@VM2-Firewall:~# systemctl start vsftpd
```

2.3.4. VM3 - Attacker settings

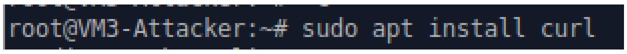


```
root@VM3-Attacker:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8c:70:69 brd ff:ff:ff:ff:ff:ff
    inet 192.168.5.30/24 brd 192.168.5.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::2484:df50:cdc5:583f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

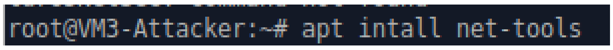
Tools:



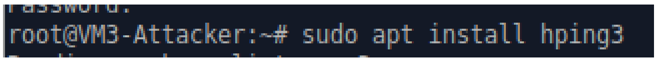
```
root@VM3-Attacker:~# sudo apt-get install nmap
```



```
root@VM3-Attacker:~# sudo apt install curl
```

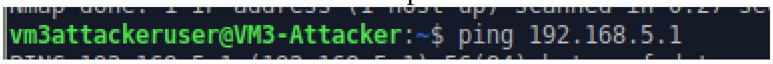


```
root@VM3-Attacker:~# apt intall net-tools
```

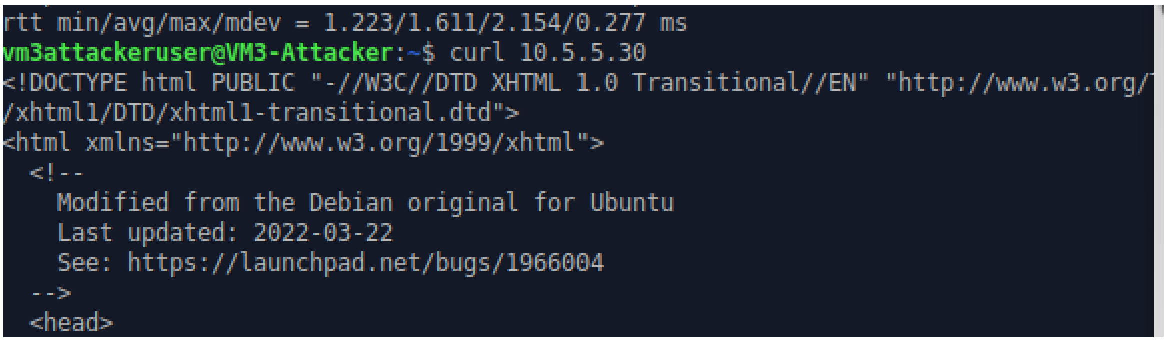


```
root@VM3-Attacker:~# sudo apt install hping3
```

Attacker can reach Firewall and Apache2 service in VM1-Server:



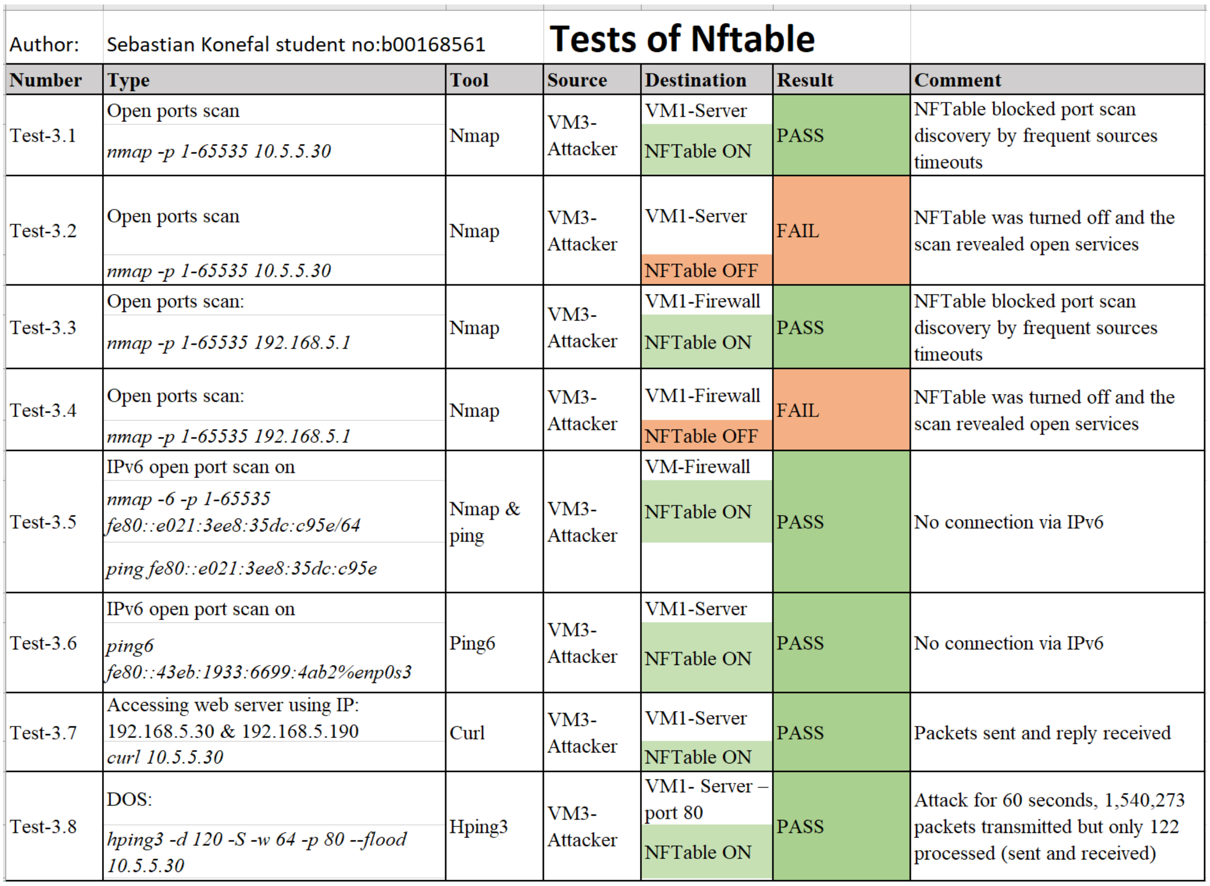
```
vm3attackeruser@VM3-Attacker:~$ ping 192.168.5.1
```



```
rtt min/avg/max/mdev = 1.223/1.611/2.154/0.277 ms
vm3attackeruser@VM3-Attacker:~$ curl 10.5.5.30
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/T
/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
```

3. Testing

Summary of tests:



Author: Sebastian Konefal student no:b00168561

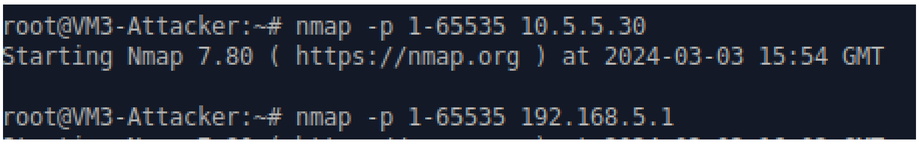
Tests of Nftable

Number	Type	Tool	Source	Destination	Result	Comment
Test-3.1	Open ports scan *nmap -p 1-65535 10.5.5.30*	Nmap	VM3-Attacker	VM1-Server NFTable ON	PASS	NFTable blocked port scan discovery by frequent sources timeouts
Test-3.2	Open ports scan *nmap -p 1-65535 10.5.5.30*	Nmap	VM3-Attacker	VM1-Server NFTable OFF	FAIL	NFTable was turned off and the scan revealed open services
Test-3.3	Open ports scan: *nmap -p 1-65535 192.168.5.1*	Nmap	VM3-Attacker	VM1-Firewall NFTable ON	PASS	NFTable blocked port scan discovery by frequent sources timeouts
Test-3.4	Open ports scan: *nmap -p 1-65535 192.168.5.1*	Nmap	VM3-Attacker	VM1-Firewall NFTable OFF	FAIL	NFTable was turned off and the scan revealed open services
Test-3.5	IPv6 open port scan on *nmap -6 -p 1-65535 fe80::e021:3ee8:35dc:c95e/64* *ping fe80::e021:3ee8:35dc:c95e*	Nmap & ping	VM3-Attacker	VM-Firewall NFTable ON	PASS	No connection via IPv6
Test-3.6	IPv6 open port scan on *ping6 fe80::43eb:1933:6699:4ab2%enp0s3*	Ping6	VM3-Attacker	VM1-Server NFTable ON	PASS	No connection via IPv6
Test-3.7	Accessing web server using IP: 192.168.5.30 & 192.168.5.190 *curl 10.5.5.30*	Curl	VM3-Attacker	VM1-Server NFTable ON	PASS	Packets sent and reply received
Test-3.8	DOS: *hping3 -d 120 -S -w 64 -p 80 --flood 10.5.5.30*	Hping3	VM3-Attacker	VM1- Server – port 80 NFTable ON	PASS	Attack for 60 seconds, 1,540,273 packets transmitted but only 122 processed (sent and received)

Test-3.9	DOS: *hping3 -d 120 -S -w 64 -p 80 --flood 10.5.5.30*	Hping3	VM3-Attacker	VM1-Server – port 80 NFTable OFF	FAIL	Attack for 60 seconds, 1,500,179 packet transmitted resulting in 794,546 packets received and total 1,377,327 packets processed
Test-3.10	DOS using IP of non-privileged range: *hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1*	Hping3	VM3-Attacker	VM2-Firewall – intnet port 22 NFTable ON	PASS	Attack for 60 seconds, 331,053 packet dropped out of 331,054 sent. Allowed 1 packet.
Test-3.11	DOS using IP of privileged range - IP 192.168.5.30 used: *hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1*	Hping3	VM3-Attacker	VM2-Firewall – extnet port 22 NFTable ON	FAIL – FIXED BY TEST 3.12	Attack for 60 seconds, 4,831,512 packets sent and 409,103 received resulting in total of 700,209 packets sent and received from VM2-Firewall
Test-3.12 FIX	DOS using IP of privileged range - IP 192.168.5.30 used: *hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1*	Hping3	VM3-Attacker	VM2-Firewall – extnet port 22 NFTable ON	PASS	Additional rule was added to fix Test-3.11. Attack for 60 seconds, 332,650 packets transmitted and processed only 62 packets
Test-3.13	DOS using IP of privileged range - IP 192.168.5.30 used: *hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1*	Hping3	VM3-Attacker	VM2-Firewall – extnet port NFTable OFF	FAIL	Attack for 60 seconds, 6,121,501 packets transmitted and 401,484 packets accepted by VM2-Firewall
Test-3.14	Triggering egress hook by accessing web server *curl 10.5.5.30*	Curl	VM2-Firewall	VM1-Server – port 80 NFTable ON	PASS	Egress hook triggered
Test-3.15	Connecting to FTP (hidden open port) & SSH *nc -w1 -vz 10.5.5.30 21* *nc -w1 -vz 10.5.5.30 22* *nc -w1 -vz 192.168.5.1 21* *nc -w1 -vz 192.168.5.1 22*	Netcat	VM2-Firewall	VM1-Server & VM2-Firewall – port 21 & 22 NFTable ON	PASS	NFTable blocked open FTP port and allowed SSH traffic

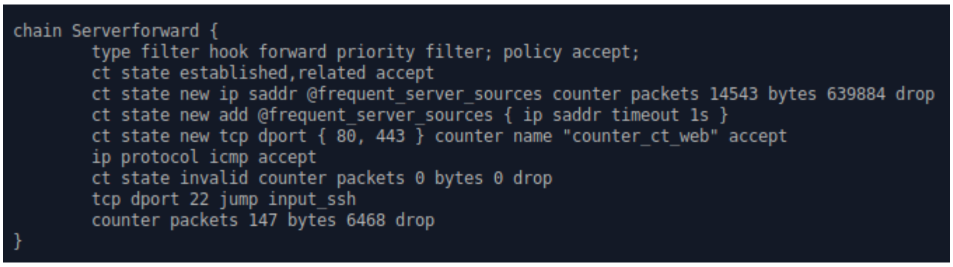
3.1. Open port scan on VM1-Server

Nmap search for all possible ports. VM-Attacker does not see FTP open port in VM3-Firewall, because VM-Firewall is blocking discovery by timeouts:



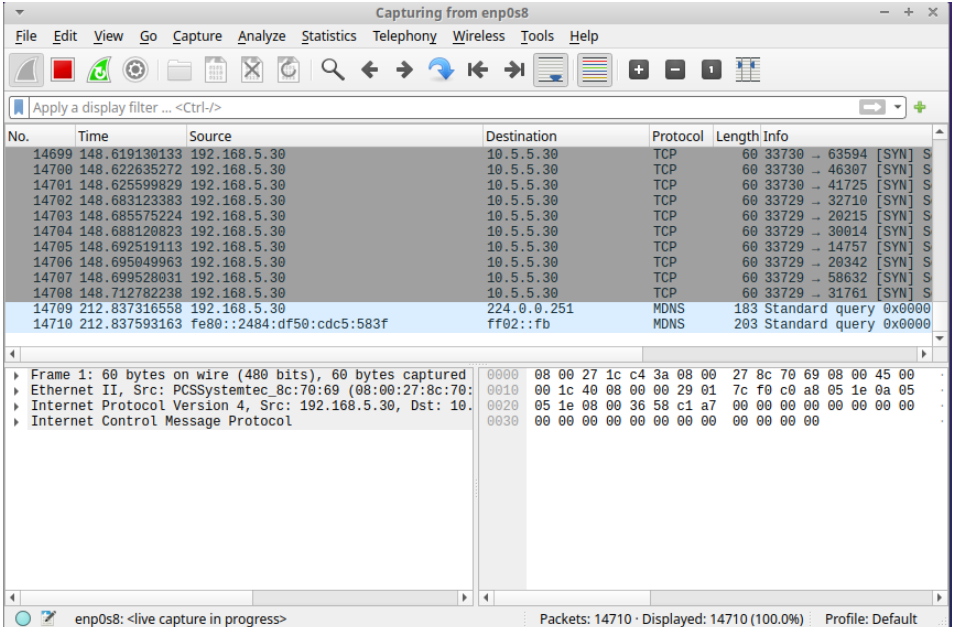
```
root@VM3-Attacker:~# nmap -p 1-65535 10.5.5.30
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-03 15:54 GMT

root@VM3-Attacker:~# nmap -p 1-65535 192.168.5.1
```

Confirmed block traffic in VM3-Firewall NFTable:



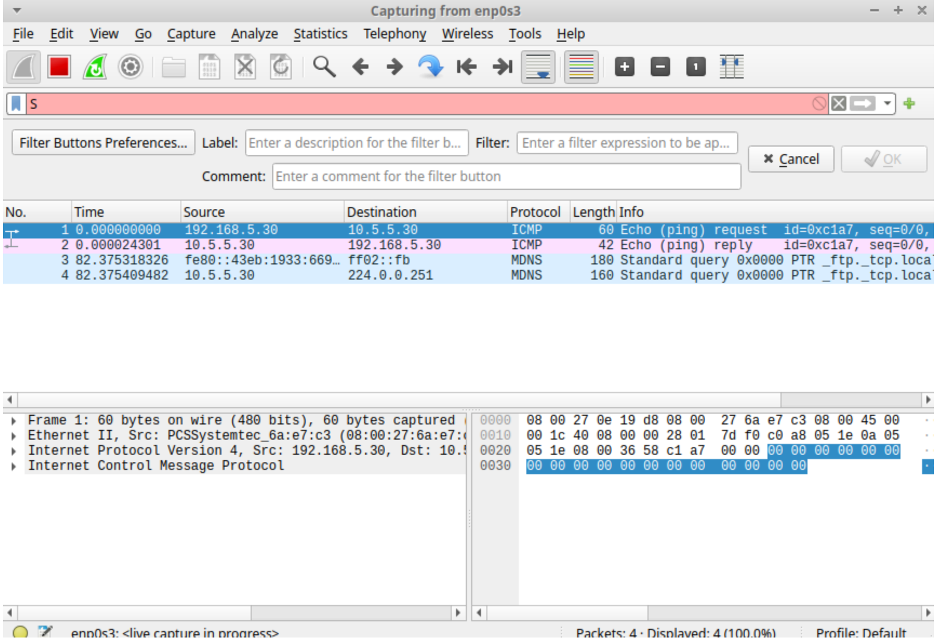
```
chain Serverforward {
        type filter hook forward priority filter; policy accept;
        ct state established,related accept
        ct state new ip saddr @frequent_server_sources counter packets 14543 bytes 639884 drop
        ct state new add @frequent_server_sources { ip saddr timeout 1s }
        ct state new tcp dport { 80, 443 } counter name "counter_ct_web" accept
        ip protocol icmp accept
        ct state invalid counter packets 0 bytes 0 drop
        tcp dport 22 jump input_ssh
        counter packets 147 bytes 6468 drop
}
```

Confirmed using Wireshark on VM2-Firewall:


Capturing from enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time        Source                       Destination   Protocol Length Info
14699 148.619130133 192.168.5.30               10.5.5.30     TCP         60 33730 → 63594 [SYN] S
14700 148.622635272 192.168.5.30               10.5.5.30     TCP         60 33730 → 46307 [SYN] S
14701 148.625599829 192.168.5.30               10.5.5.30     TCP         60 33730 → 41725 [SYN] S
14702 148.683123383 192.168.5.30               10.5.5.30     TCP         60 33729 → 32710 [SYN] S
14703 148.685575224 192.168.5.30               10.5.5.30     TCP         60 33729 → 20215 [SYN] S
14704 148.688120823 192.168.5.30               10.5.5.30     TCP         60 33729 → 30014 [SYN] S
14705 148.692519113 192.168.5.30               10.5.5.30     TCP         60 33729 → 14757 [SYN] S
14706 148.695049963 192.168.5.30               10.5.5.30     TCP         60 33729 → 20342 [SYN] S
14707 148.699528031 192.168.5.30               10.5.5.30     TCP         60 33729 → 58632 [SYN] S
14708 148.712782238 192.168.5.30               10.5.5.30     TCP         60 33729 → 31761 [SYN] S
14709 212.837316558 192.168.5.30               224.0.0.251   MDNS       183 Standard query 0x0000
14710 212.837593163 fe80::2484:df50:cdc5:583f  ff02::fb      MDNS       203 Standard query 0x0000

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured     0000  08 00 27 1c c4 3a 08 00  27 8c 70 69 08 00 45 00
▸ Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27:8c:70:    0010  00 1c 40 08 00 00 29 01  7c f0 c0 a8 05 1e 0a 05
▸ Internet Protocol Version 4, Src: 192.168.5.30, Dst: 10.    0020  05 1e 08 00 36 58 c1 a7  00 00 00 00 00 00 00 00
▸ Internet Control Message Protocol                           0030  00 00 00 00 00 00 00 00  00 00 00 00

enp0s8: <live capture in progress>      Packets: 14710 · Displayed: 14710 (100.0%)      Profile: Default
```

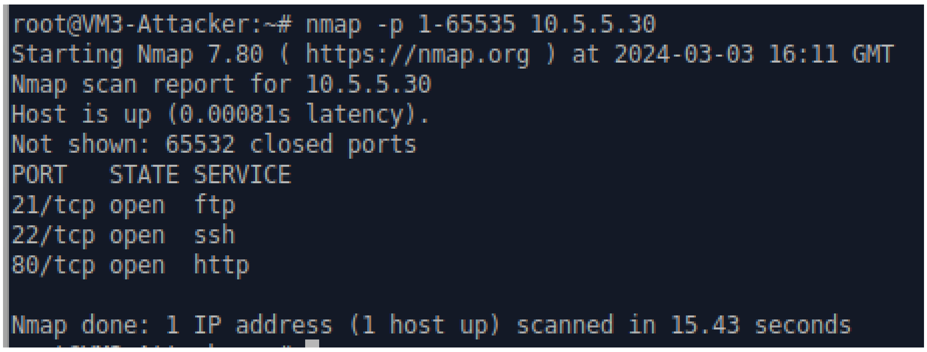
Confirmed using Wireshark on VM1-Server:


Capturing from enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
S
Filter Buttons Preferences…
Label:
Enter a description for the filter b…
Filter:
Enter a filter expression to be ap…
Comment:
Enter a comment for the filter button
✖ Cancel
✓ OK

```
No. Time          Source                 Destination   Protocol Length Info
  1 0.000000000   192.168.5.30           10.5.5.30     ICMP         60 Echo (ping) request  id=0xc1a7, seq=0/0,
  2 0.000024301   10.5.5.30              192.168.5.30  ICMP         42 Echo (ping) reply    id=0xc1a7, seq=0/0,
  3 82.375318326  fe80::43eb:1933:669…   ff02::fb      MDNS        180 Standard query 0x0000 PTR _ftp._tcp.loca
  4 82.375409482  10.5.5.30              224.0.0.251   MDNS        160 Standard query 0x0000 PTR _ftp._tcp.loca

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured    0000  08 00 27 0e 19 d8 08 00  27 6a e7 c3 08 00 45 00
▸ Ethernet II, Src: PCSSystemtec_6a:e7:c3 (08:00:27:6a:e7:   0010  00 1c 40 08 00 00 28 01  7d f0 c0 a8 05 1e 0a 05
▸ Internet Protocol Version 4, Src: 192.168.5.30, Dst: 10.   0020  05 1e 08 00 36 58 c1 a7  00 00 00 00 00 00 00 00
▸ Internet Control Message Protocol                          0030  00 00 00 00 00 00 00 00  00 00 00 00

enp0s3: <live capture in progress>      Packets: 4 · Displayed: 4 (100.0%)      Profile: Default
```

RESULT: PASS - NFTable blocked port scan discovery by frequent sources timeouts

3.2. Open port scan on VM1-Server – NFTable off

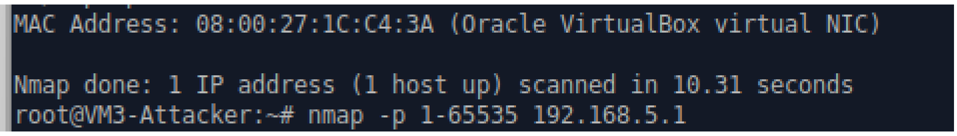


```
root@VM3-Attacker:~# nmap -p 1-65535 10.5.5.30
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-03 16:11 GMT
Nmap scan report for 10.5.5.30
Host is up (0.00081s latency).
Not shown: 65532 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 15.43 seconds
```

RESULT: FAIL – NFTable was turned off and the scan revealed additional open services

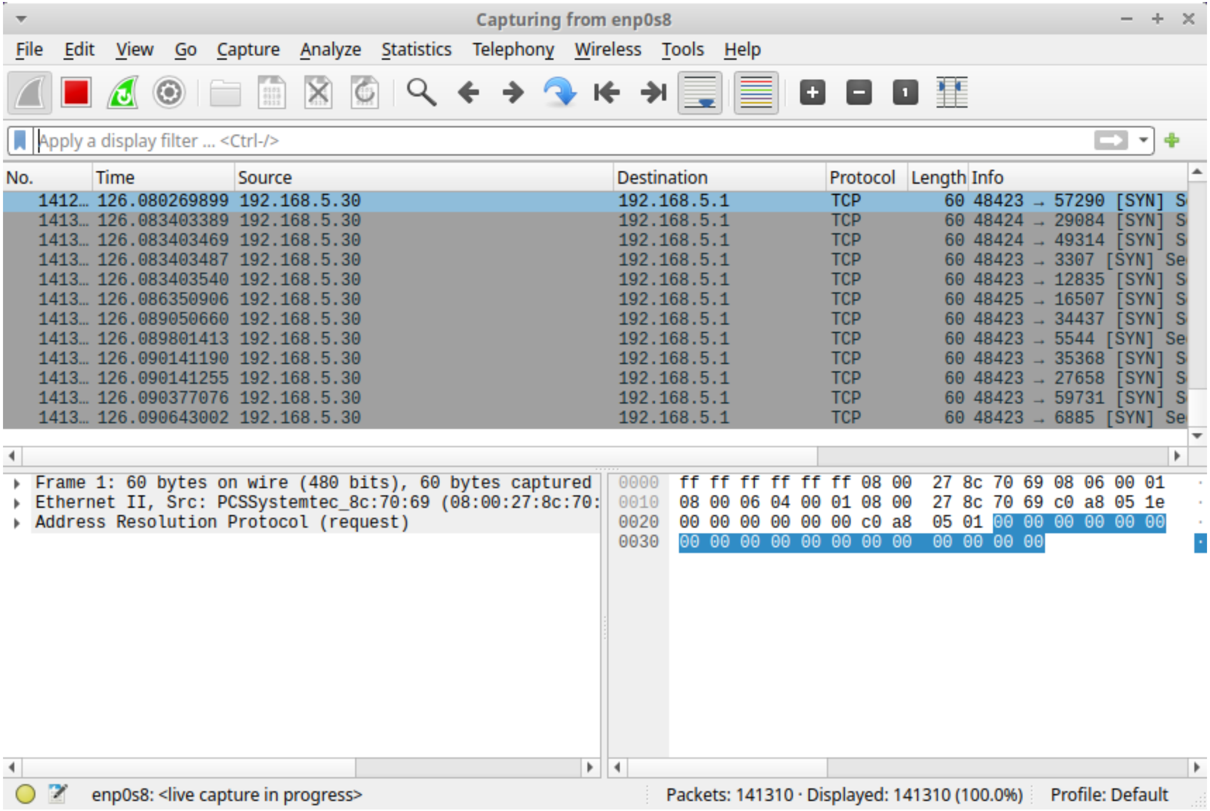
3.3. Open port scan on VM-Firewall

Nmap search for all possible ports. VM-Attacker does not see FTP open port in VM3-Firewall, because VM-Firewall is blocking discovery by timeouts:



```
MAC Address: 08:00:27:1C:C4:3A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 10.31 seconds
root@VM3-Attacker:~# nmap -p 1-65535 192.168.5.1
```

Confirmed traffic incoming VM2-Firewall by Wireshark:



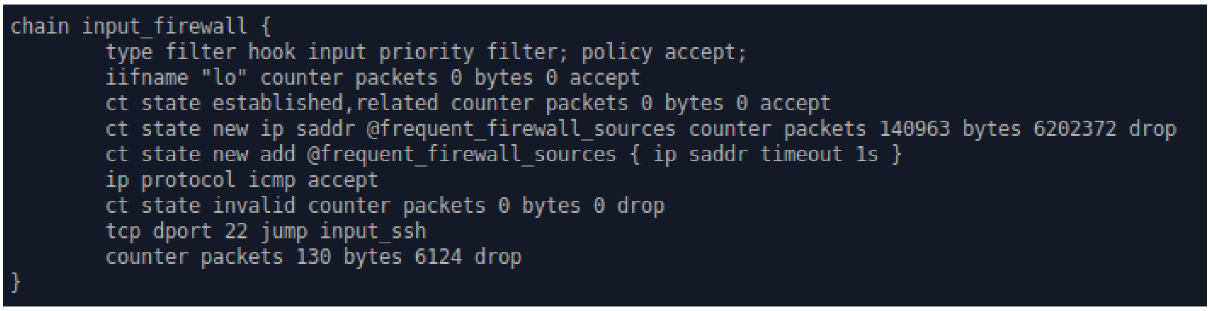
No.	Time	Source	Destination	Protocol	Length	Info
1412…	126.080269899	192.168.5.30	192.168.5.1	TCP	60	48423 → 57290 [SYN] S
1413…	126.083403389	192.168.5.30	192.168.5.1	TCP	60	48424 → 29084 [SYN] S
1413…	126.083403469	192.168.5.30	192.168.5.1	TCP	60	48424 → 49314 [SYN] S
1413…	126.083403487	192.168.5.30	192.168.5.1	TCP	60	48423 → 3307 [SYN] Se
1413…	126.083403540	192.168.5.30	192.168.5.1	TCP	60	48423 → 12835 [SYN] S
1413…	126.086350906	192.168.5.30	192.168.5.1	TCP	60	48425 → 16507 [SYN] S
1413…	126.089050660	192.168.5.30	192.168.5.1	TCP	60	48423 → 34437 [SYN] S
1413…	126.089801413	192.168.5.30	192.168.5.1	TCP	60	48423 → 5544 [SYN] Se
1413…	126.090141190	192.168.5.30	192.168.5.1	TCP	60	48423 → 35368 [SYN] S
1413…	126.090141255	192.168.5.30	192.168.5.1	TCP	60	48423 → 27658 [SYN] S
1413…	126.090377076	192.168.5.30	192.168.5.1	TCP	60	48423 → 59731 [SYN] S
1413…	126.090643002	192.168.5.30	192.168.5.1	TCP	60	48423 → 6885 [SYN] Se

```
▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured
▸ Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27:8c:70:
▸ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 08 00  27 8c 70 69 08 06 00 01
0010  08 00 06 04 00 01 08 00  27 8c 70 69 c0 a8 05 1e
0020  00 00 00 00 00 00 c0 a8  05 01 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 00 00
```

enp0s8: <live capture in progress>   Packets: 141310 · Displayed: 141310 (100.0%)   Profile: Default

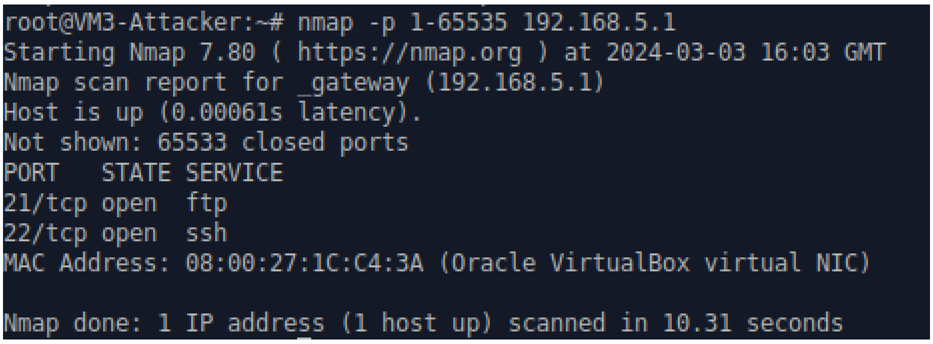
Confirmed block traffic in VM3-Firewall NFTable:



```
chain input_firewall {
        type filter hook input priority filter; policy accept;
        iifname "lo" counter packets 0 bytes 0 accept
        ct state established,related counter packets 0 bytes 0 accept
        ct state new ip saddr @frequent_firewall_sources counter packets 140963 bytes 6202372 drop
        ct state new add @frequent_firewall_sources { ip saddr timeout 1s }
        ip protocol icmp accept
        ct state invalid counter packets 0 bytes 0 drop
        tcp dport 22 jump input_ssh
        counter packets 130 bytes 6124 drop
}
```

RESULT: PASS - NFTable blocked port scan discovery by frequent sources timeouts

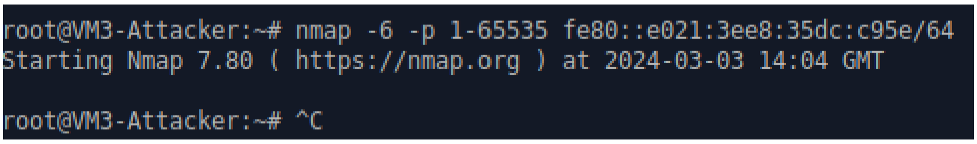
3.4. Open port scan on VM2-Firewall – NFTable off



```
root@VM3-Attacker:~# nmap -p 1-65535 192.168.5.1
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-03 16:03 GMT
Nmap scan report for _gateway (192.168.5.1)
Host is up (0.00061s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
MAC Address: 08:00:27:1C:C4:3A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 10.31 seconds
```

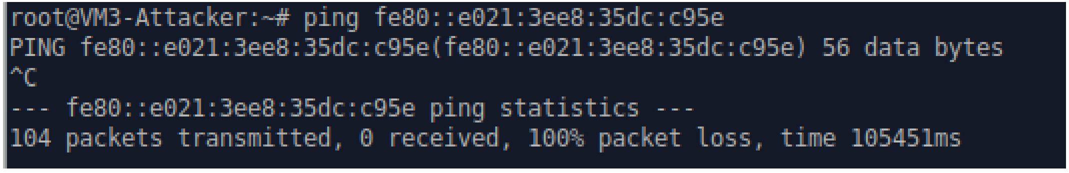
RESULT: FAIL – NFTable was turned off and the scan revealed additional open FTP service

3.5. IPv6 open port scan on VM-Firewall



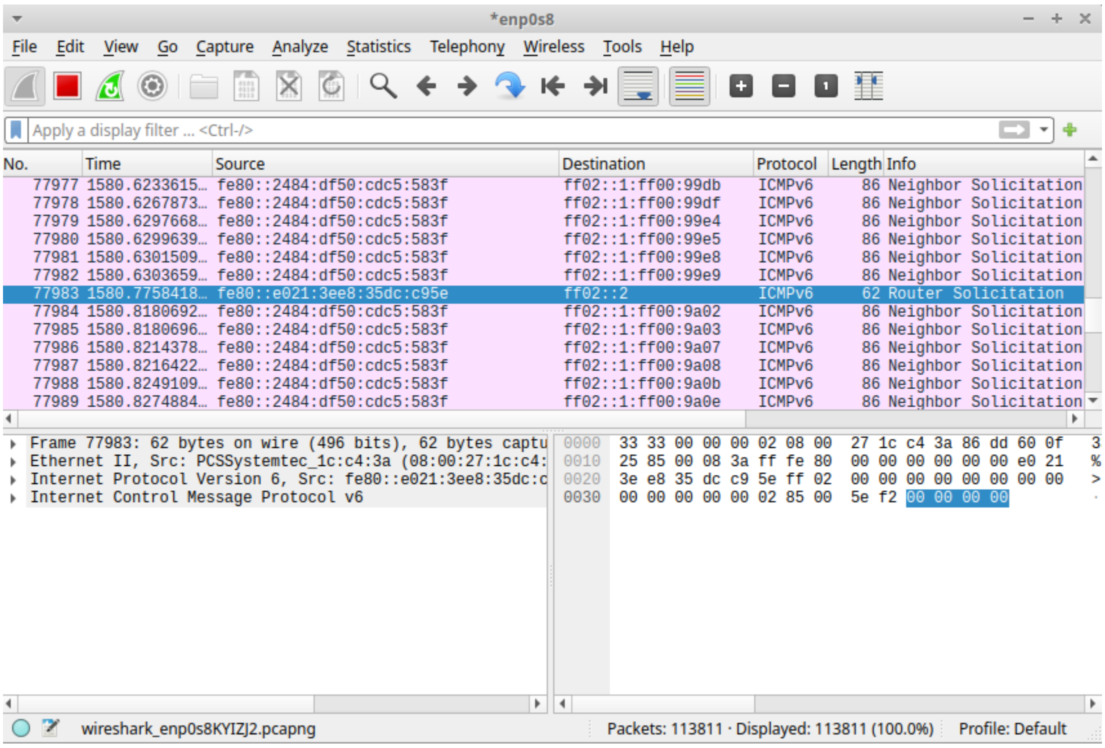
```
root@VM3-Attacker:~# nmap -6 -p 1-65535 fe80::e021:3ee8:35dc:c95e/64
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-03 14:04 GMT

root@VM3-Attacker:~# ^C
```

No connection via ipv6:



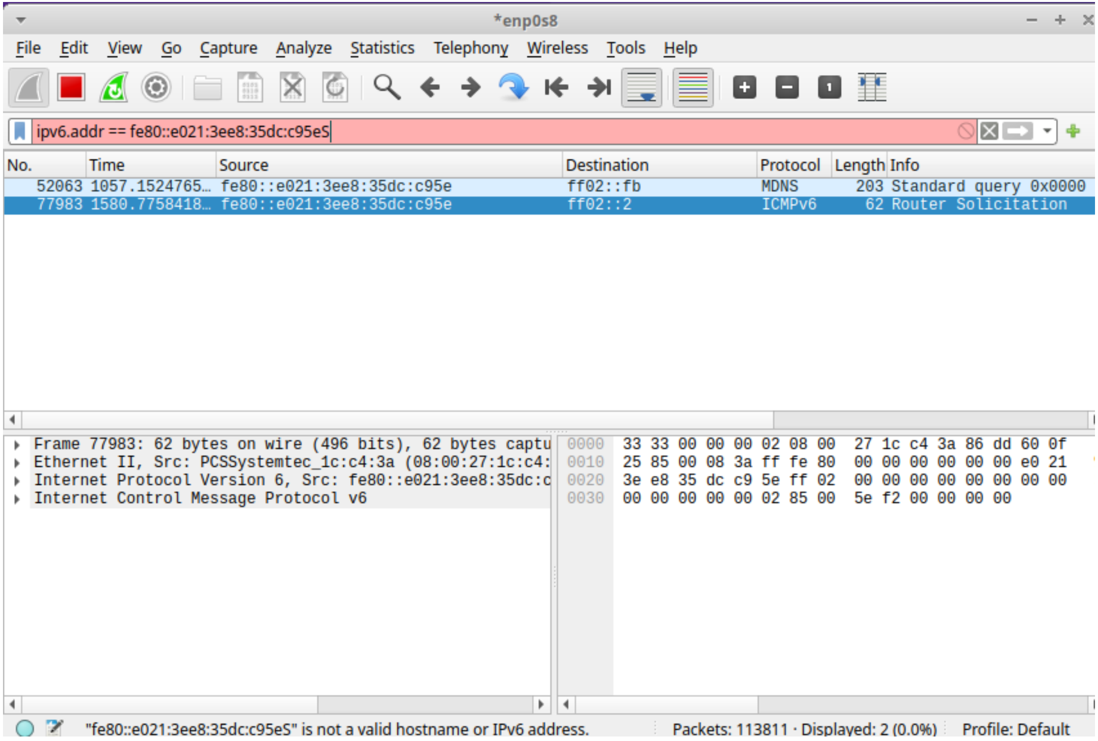
```
root@VM3-Attacker:~# ping fe80::e021:3ee8:35dc:c95e
PING fe80::e021:3ee8:35dc:c95e(fe80::e021:3ee8:35dc:c95e) 56 data bytes
^C
--- fe80::e021:3ee8:35dc:c95e ping statistics ---
104 packets transmitted, 0 received, 100% packet loss, time 105451ms
```

Packets transmitted via VM2-Firewall:



```
Apply a display filter ... <Ctrl-/>

No.   Time            Source                       Destination          Protocol  Length Info
77977 1580.6233615… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99db   ICMPv6      86 Neighbor Solicitation
77978 1580.6267873… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99df   ICMPv6      86 Neighbor Solicitation
77979 1580.6297668… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99e4   ICMPv6      86 Neighbor Solicitation
77980 1580.6299639… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99e5   ICMPv6      86 Neighbor Solicitation
77981 1580.6301509… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99e8   ICMPv6      86 Neighbor Solicitation
77982 1580.6303659… fe80::2484:df50:cdc5:583f   ff02::1:ff00:99e9   ICMPv6      86 Neighbor Solicitation
77983 1580.7758418… fe80::e021:3ee8:35dc:c95e   ff02::2             ICMPv6      62 Router Solicitation
77984 1580.8180692… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a02   ICMPv6      86 Neighbor Solicitation
77985 1580.8180696… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a03   ICMPv6      86 Neighbor Solicitation
77986 1580.8214378… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a07   ICMPv6      86 Neighbor Solicitation
77987 1580.8216422… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a08   ICMPv6      86 Neighbor Solicitation
77988 1580.8249109… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a0b   ICMPv6      86 Neighbor Solicitation
77989 1580.8274884… fe80::2484:df50:cdc5:583f   ff02::1:ff00:9a0e   ICMPv6      86 Neighbor Solicitation

Frame 77983: 62 bytes on wire (496 bits), 62 bytes captu   0000  33 33 00 00 00 02 08 00  27 1c c4 3a 86 dd 60 0f   3
Ethernet II, Src: PCSSystemtec_1c:c4:3a (08:00:27:1c:c4:   0010  25 85 00 08 3a ff fe 80  00 00 00 00 00 00 e0 21   %
Internet Protocol Version 6, Src: fe80::e021:3ee8:35dc:c   0020  3e e8 35 dc c9 5e ff 02  00 00 00 00 00 00 00 00   >
Internet Control Message Protocol v6                       0030  00 00 00 00 00 02 85 00  5e f2 00 00 00 00         .

wireshark_enp0s8KYIZJ2.pcapng          Packets: 113811 · Displayed: 113811 (100.0%)    Profile: Default
```

Packets replied by VM2-Firewall:

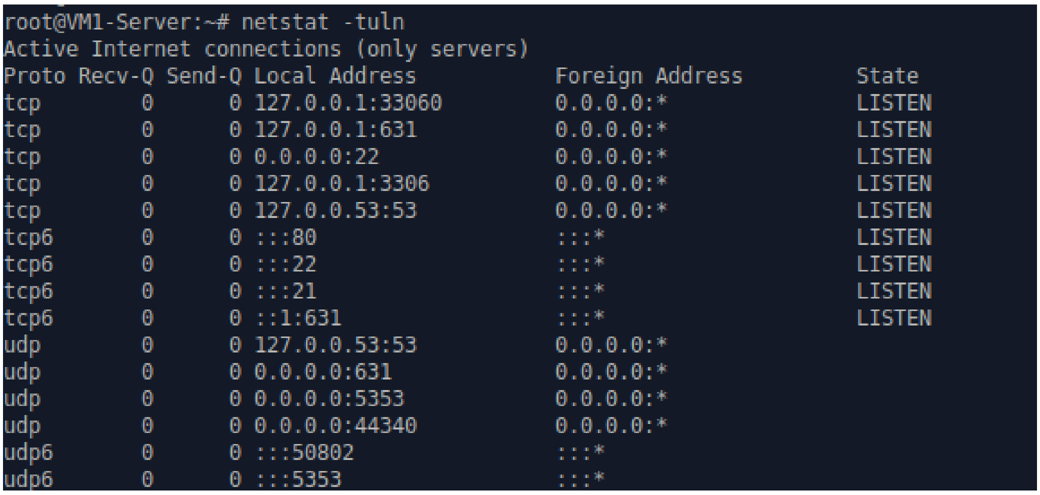


```
ipv6.addr == fe80::e021:3ee8:35dc:c95eS

No.   Time            Source                       Destination          Protocol  Length Info
52063 1057.1524765… fe80::e021:3ee8:35dc:c95e   ff02::fb            MDNS       203 Standard query 0x0000
77983 1580.7758418… fe80::e021:3ee8:35dc:c95e   ff02::2             ICMPv6      62 Router Solicitation

Frame 77983: 62 bytes on wire (496 bits), 62 bytes captu   0000  33 33 00 00 00 02 08 00  27 1c c4 3a 86 dd 60 0f
Ethernet II, Src: PCSSystemtec_1c:c4:3a (08:00:27:1c:c4:   0010  25 85 00 08 3a ff fe 80  00 00 00 00 00 00 e0 21
Internet Protocol Version 6, Src: fe80::e021:3ee8:35dc:c   0020  3e e8 35 dc c9 5e ff 02  00 00 00 00 00 00 00 00
Internet Control Message Protocol v6                       0030  00 00 00 00 00 02 85 00  5e f2 00 00 00 00

"fe80::e021:3ee8:35dc:c95eS" is not a valid hostname or IPv6 address.    Packets: 113811 · Displayed: 2 (0.0%)    Profile: Default
```

RESULT: PASS - No connection via IPv6

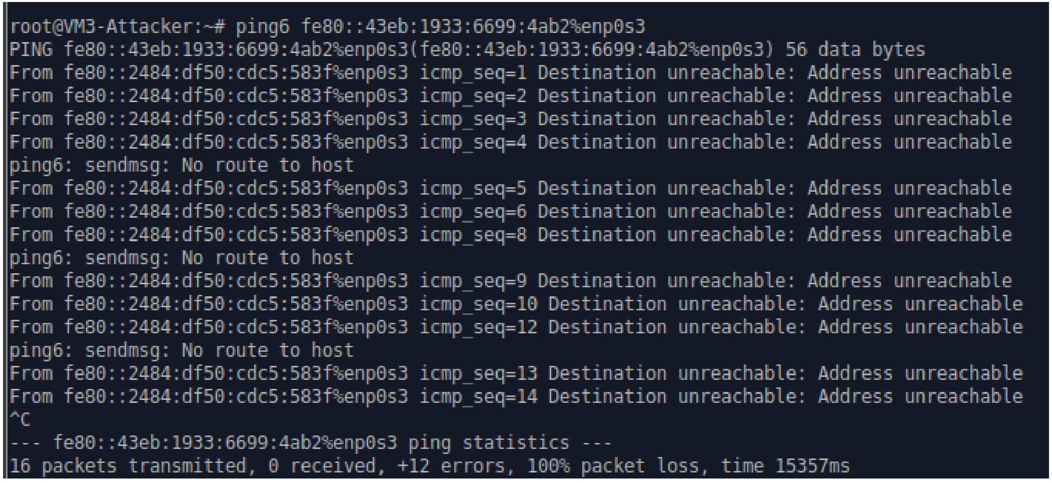
3.6. IPv6 open port scan on VM2-Server

Open ports:



```
root@VM1-Server:~# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 0.0.0.0:44340           0.0.0.0:*
udp6       0      0 :::50802                :::*
udp6       0      0 :::5353                 :::*
```

No connection via ipv6:

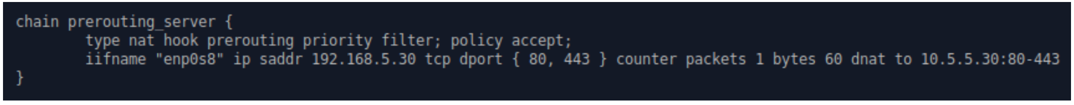


```
root@VM3-Attacker:~# ping6 fe80::43eb:1933:6699:4ab2%enp0s3
PING fe80::43eb:1933:6699:4ab2%enp0s3(fe80::43eb:1933:6699:4ab2%enp0s3) 56 data bytes
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=1 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=2 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=3 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=4 Destination unreachable: Address unreachable
ping6: sendmsg: No route to host
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=5 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=6 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=8 Destination unreachable: Address unreachable
ping6: sendmsg: No route to host
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=9 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=10 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=12 Destination unreachable: Address unreachable
ping6: sendmsg: No route to host
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=13 Destination unreachable: Address unreachable
From fe80::2484:df50:cdc5:583f%enp0s3 icmp_seq=14 Destination unreachable: Address unreachable
^C
--- fe80::43eb:1933:6699:4ab2%enp0s3 ping statistics ---
16 packets transmitted, 0 received, +12 errors, 100% packet loss, time 15357ms
```

RESULT: PASS - No connection via IPv6

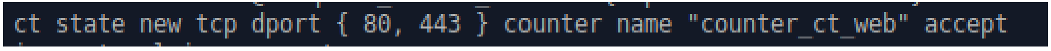
3.7. Accessing Web Server on VM1-Server

Tested with VM3-Attacker IP 192.168.5.30 match by this rule:



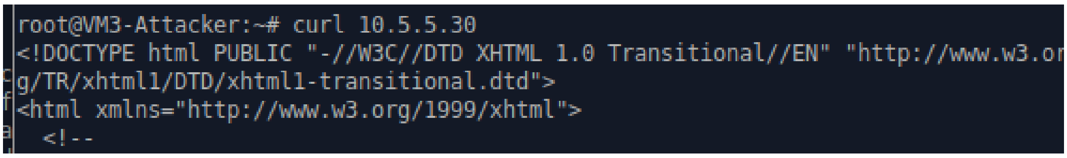
```
chain prerouting_server {
        type nat hook prerouting priority filter; policy accept;
        iifname "enp0s8" ip saddr 192.168.5.30 tcp dport { 80, 443 } counter packets 1 bytes 60 dnat to 10.5.5.30:80-443
}
```

And 192.168.5.190 matched by this rule in the forward hook:



```
ct state new tcp dport { 80, 443 } counter name "counter_ct_web" accept
```

Result for both IPs:

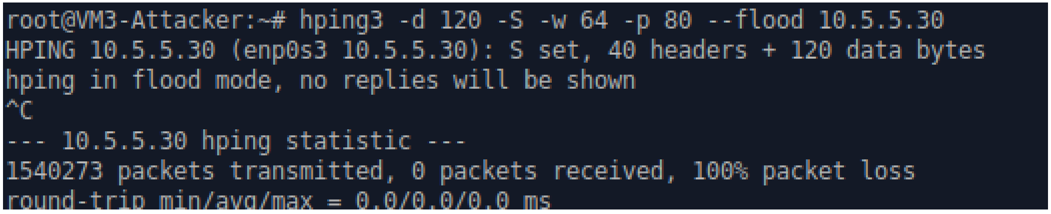


```
root@VM3-Attacker:~# curl 10.5.5.30
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
```

RESULT: PASS - Packet sent and reply received

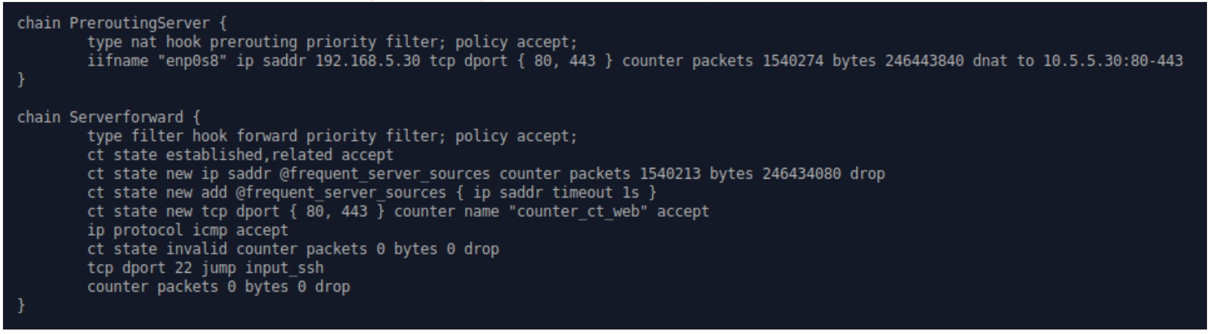
3.8. DOS on VM2-Server port 80

VM3-Attacker sent 1,540,273 packets:



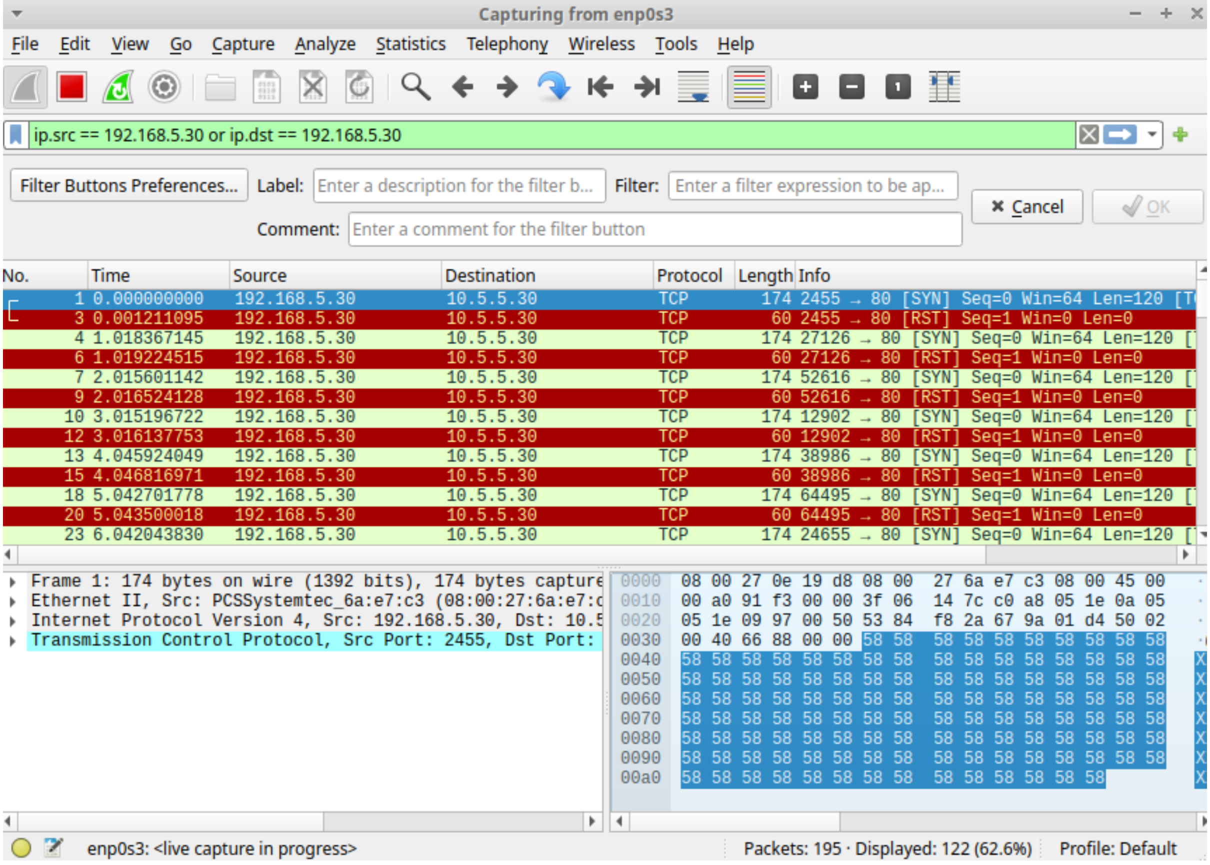
```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 80 --flood 10.5.5.30
HPING 10.5.5.30 (enp0s3 10.5.5.30): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.5.5.30 hping statistic ---
1540273 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

VM2-Firewall blocked flooding attack by this rule of @frequent_server_sources:



```
chain PreroutingServer {
        type nat hook prerouting priority filter; policy accept;
        iifname "enp0s8" ip saddr 192.168.5.30 tcp dport { 80, 443 } counter packets 1540274 bytes 246443840 dnat to 10.5.5.30:80-443
}

chain Serverforward {
        type filter hook forward priority filter; policy accept;
        ct state established,related accept
        ct state new ip saddr @frequent_server_sources counter packets 1540213 bytes 246434080 drop
        ct state new add @frequent_server_sources { ip saddr timeout 1s }
        ct state new tcp dport { 80, 443 } counter name "counter_ct_web" accept
        ip protocol icmp accept
        ct state invalid counter packets 0 bytes 0 drop
        tcp dport 22 jump input_ssh
        counter packets 0 bytes 0 drop
}
```

VM1-Server processed (sent and received) **only 122 packets**:



Capturing from enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Filter Buttons Preferences...
Label:
Filter:
Comment:
✖ Cancel
✔ OK

```
ip.src == 192.168.5.30 or ip.dst == 192.168.5.30
```

Label: Enter a description for the filter b...
Filter: Enter a filter expression to be ap...
Comment: Enter a comment for the filter button

```
No.  Time          Source         Destination   Protocol Length Info
   1 0.000000000   192.168.5.30   10.5.5.30     TCP      174 2455 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   3 0.001211095   192.168.5.30   10.5.5.30     TCP       60 2455 → 80 [RST] Seq=1 Win=0 Len=0
   4 1.018367145   192.168.5.30   10.5.5.30     TCP      174 27126 → 80 [SYN] Seq=0 Win=64 Len=120 [
   6 1.019224515   192.168.5.30   10.5.5.30     TCP       60 27126 → 80 [RST] Seq=1 Win=0 Len=0
   7 2.015601142   192.168.5.30   10.5.5.30     TCP      174 52616 → 80 [SYN] Seq=0 Win=64 Len=120 [
   9 2.016524128   192.168.5.30   10.5.5.30     TCP       60 52616 → 80 [RST] Seq=1 Win=0 Len=0
  10 3.015196722   192.168.5.30   10.5.5.30     TCP      174 12902 → 80 [SYN] Seq=0 Win=64 Len=120 [
  12 3.016137753   192.168.5.30   10.5.5.30     TCP       60 12902 → 80 [RST] Seq=1 Win=0 Len=0
  13 4.045924049   192.168.5.30   10.5.5.30     TCP      174 38986 → 80 [SYN] Seq=0 Win=64 Len=120 [
  15 4.046816971   192.168.5.30   10.5.5.30     TCP       60 38986 → 80 [RST] Seq=1 Win=0 Len=0
  18 5.042701778   192.168.5.30   10.5.5.30     TCP      174 64495 → 80 [SYN] Seq=0 Win=64 Len=120 [
  20 5.043500018   192.168.5.30   10.5.5.30     TCP       60 64495 → 80 [RST] Seq=1 Win=0 Len=0
  23 6.042043830   192.168.5.30   10.5.5.30     TCP      174 24655 → 80 [SYN] Seq=0 Win=64 Len=120 [
```

```
▸ Frame 1: 174 bytes on wire (1392 bits), 174 bytes capture
▸ Ethernet II, Src: PCSSystemtec_6a:e7:c3 (08:00:27:6a:e7:c
▸ Internet Protocol Version 4, Src: 192.168.5.30, Dst: 10.5
▸ Transmission Control Protocol, Src Port: 2455, Dst Port:
```

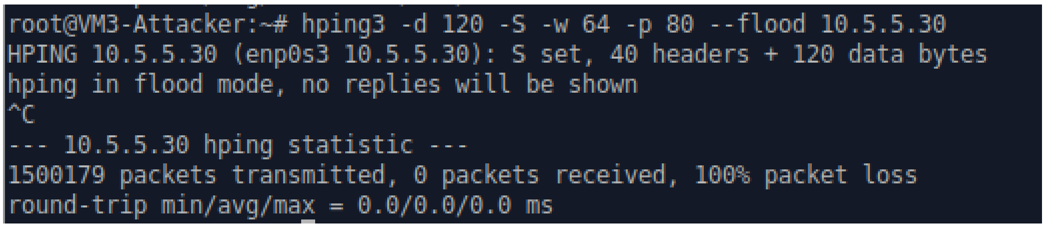
```
0000  08 00 27 0e 19 d8 08 00  27 6a e7 c3 08 00 45 00
0010  00 a0 91 f3 00 00 3f 06  14 7c c0 a8 05 1e 0a 05
0020  05 1e 09 97 00 50 53 84  f8 2a 67 9a 01 d4 50 02
0030  00 40 66 88 00 00 58 58  58 58 58 58 58 58 58 58
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58         X
```

enp0s3: <live capture in progress> Packets: 195 · Displayed: 122 (62.6%) Profile: Default

RESULT: PASS - Ttransmitted 1,540,273 packets but only 122 processed (sent and received)

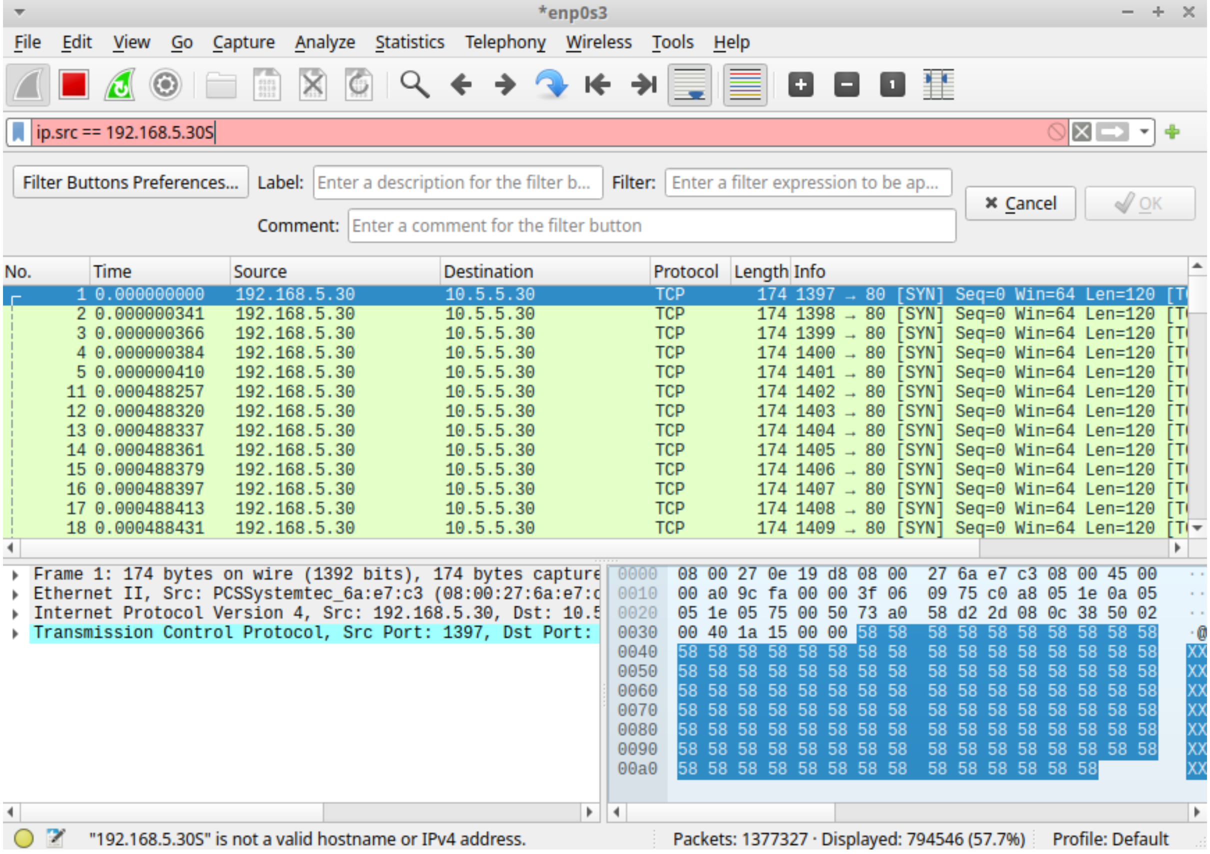
3.9. DOS on VM2-Server port 80 – NFTable off

VM3-Attacker sent 1,500,179 packets:



```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 80 --flood 10.5.5.30
HPING 10.5.5.30 (enp0s3 10.5.5.30): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.5.5.30 hping statistic ---
1500179 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

VM1-Server received 794,546 packets and processed (sent and received) 1,377,327 packets:



*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Filter Buttons Preferences...
Label: Enter a description for the filter b...
Filter: Enter a filter expression to be ap...
✖ Cancel
✔ OK
Comment: Enter a comment for the filter button

```
ip.src == 192.168.5.30S
```

```
No.   Time         Source          Destination   Protocol Length Info
    1 0.000000000  192.168.5.30    10.5.5.30     TCP      174 1397 → 80 [SYN] Seq=0 Win=64 Len=120 [T
    2 0.000000341  192.168.5.30    10.5.5.30     TCP      174 1398 → 80 [SYN] Seq=0 Win=64 Len=120 [T
    3 0.000000366  192.168.5.30    10.5.5.30     TCP      174 1399 → 80 [SYN] Seq=0 Win=64 Len=120 [T
    4 0.000000384  192.168.5.30    10.5.5.30     TCP      174 1400 → 80 [SYN] Seq=0 Win=64 Len=120 [T
    5 0.000000410  192.168.5.30    10.5.5.30     TCP      174 1401 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   11 0.000488257  192.168.5.30    10.5.5.30     TCP      174 1402 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   12 0.000488320  192.168.5.30    10.5.5.30     TCP      174 1403 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   13 0.000488337  192.168.5.30    10.5.5.30     TCP      174 1404 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   14 0.000488361  192.168.5.30    10.5.5.30     TCP      174 1405 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   15 0.000488379  192.168.5.30    10.5.5.30     TCP      174 1406 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   16 0.000488397  192.168.5.30    10.5.5.30     TCP      174 1407 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   17 0.000488413  192.168.5.30    10.5.5.30     TCP      174 1408 → 80 [SYN] Seq=0 Win=64 Len=120 [T
   18 0.000488431  192.168.5.30    10.5.5.30     TCP      174 1409 → 80 [SYN] Seq=0 Win=64 Len=120 [T
```

```
▸ Frame 1: 174 bytes on wire (1392 bits), 174 bytes capture
▸ Ethernet II, Src: PCSSystemtec_6a:e7:c3 (08:00:27:6a:e7:c
▸ Internet Protocol Version 4, Src: 192.168.5.30, Dst: 10.5
▸ Transmission Control Protocol, Src Port: 1397, Dst Port:
```

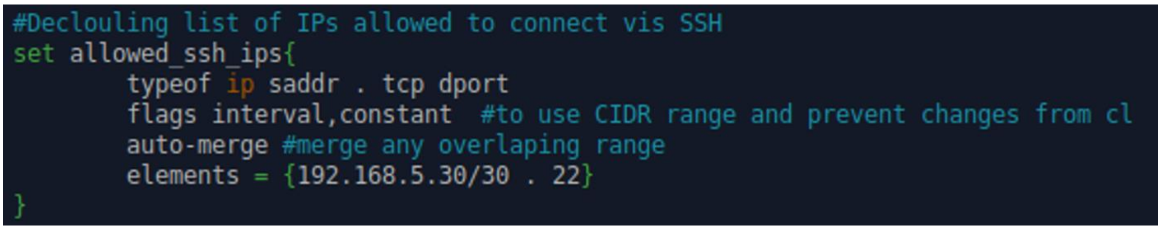
```
0000  08 00 27 0e 19 d8 08 00  27 6a e7 c3 08 00 45 00   ··
0010  00 a0 9c fa 00 00 3f 06  09 75 c0 a8 05 1e 0a 05   ··
0020  05 1e 05 75 00 50 73 a0  58 d2 2d 08 0c 38 50 02   ··
0030  00 40 1a 15 00 00 58 58  58 58 58 58 58 58 58 58   ·@
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XX
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58         XX
```

"192.168.5.30S" is not a valid hostname or IPv4 address. Packets: 1377327 · Displayed: 794546 (57.7%) Profile: Default

RESULT: FAIL - 1,500,179 packet transmitted resulting in 794,546 packets received and total 1,377,327 packets processed.

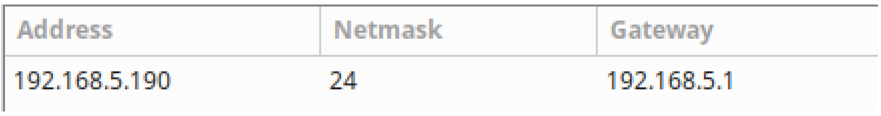
3.10. DOS on VM2-Firewall intnet on port 22 using IP of non-privileged range

Privileged range is:



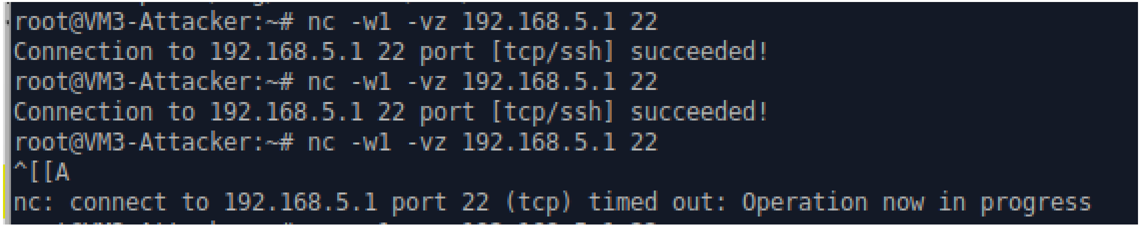
```
#Declouling list of IPs allowed to connect vis SSH
set allowed_ssh_ips{
        typeof ip saddr . tcp dport
        flags interval,constant  #to use CIDR range and prevent changes from cl
        auto-merge #merge any overlaping range
        elements = {192.168.5.30/30 . 22}
}
```

I changed IP on VM3-Attacker to range outside of privileged range:



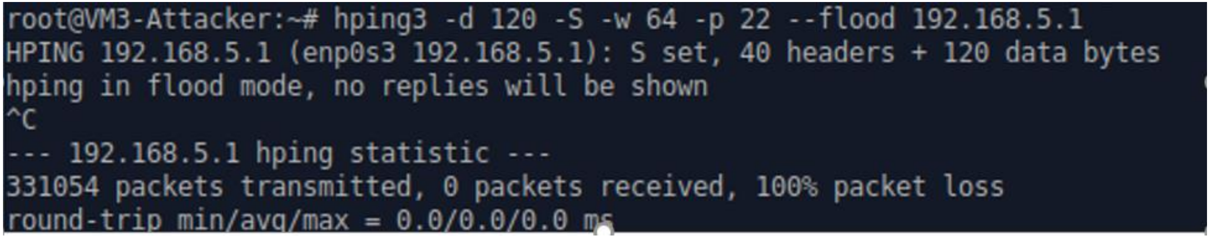
Address	Netmask	Gateway
192.168.5.190	24	192.168.5.1

Verified that connection is possible from VM3-Attacker but blocked on 3rd connection:



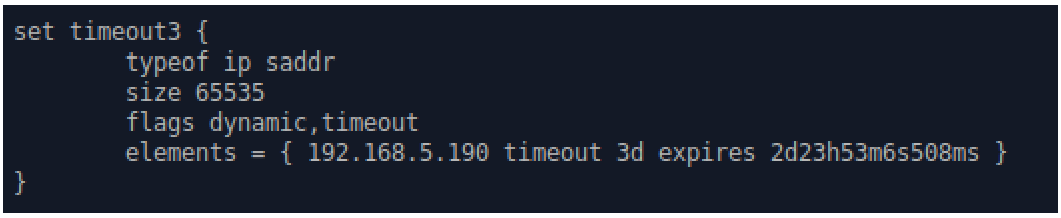
```
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
Connection to 192.168.5.1 22 port [tcp/ssh] succeeded!
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
Connection to 192.168.5.1 22 port [tcp/ssh] succeeded!
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
^[[A
nc: connect to 192.168.5.1 port 22 (tcp) timed out: Operation now in progress
```

I reset the timeouts and VM3-Attacker sent 331,054 packets:

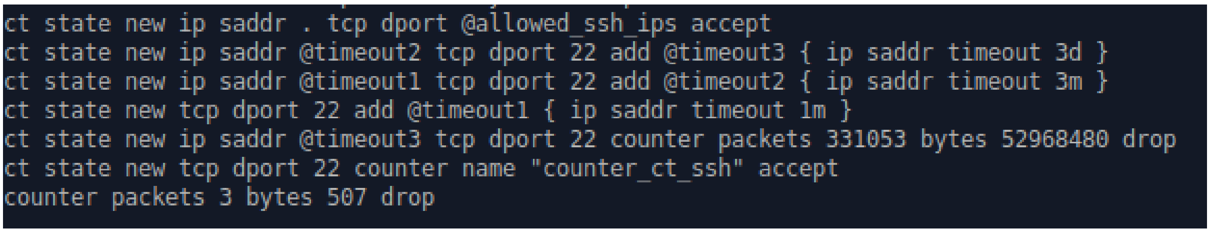


```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1
HPING 192.168.5.1 (enp0s3 192.168.5.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.5.1 hping statistic ---
331054 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Received on VM2-Firewall and blocked by rule:

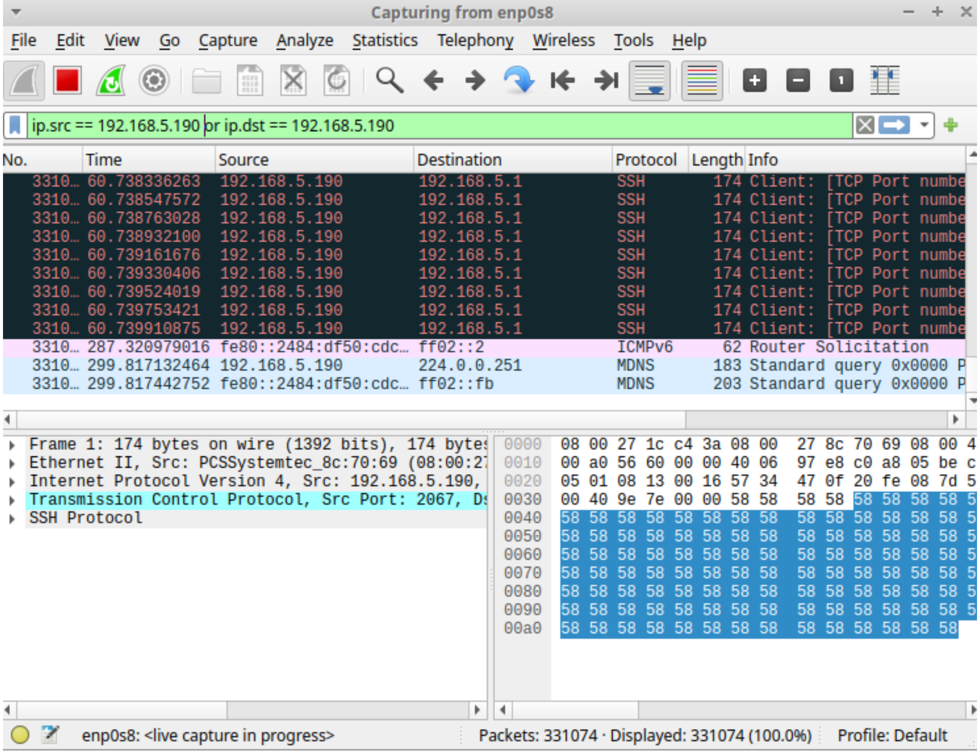


```
set timeout3 {
        typeof ip saddr
        size 65535
        flags dynamic,timeout
        elements = { 192.168.5.190 timeout 3d expires 2d23h53m6s508ms }
}
```



```
ct state new ip saddr . tcp dport @allowed_ssh_ips accept
ct state new ip saddr @timeout2 tcp dport 22 add @timeout3 { ip saddr timeout 3d }
ct state new ip saddr @timeout1 tcp dport 22 add @timeout2 { ip saddr timeout 3m }
ct state new tcp dport 22 add @timeout1 { ip saddr timeout 1m }
ct state new ip saddr @timeout3 tcp dport 22 counter packets 331053 bytes 52968480 drop
ct state new tcp dport 22 counter name "counter_ct_ssh" accept
counter packets 3 bytes 507 drop
```

Wireshark on VM2-Firewall:



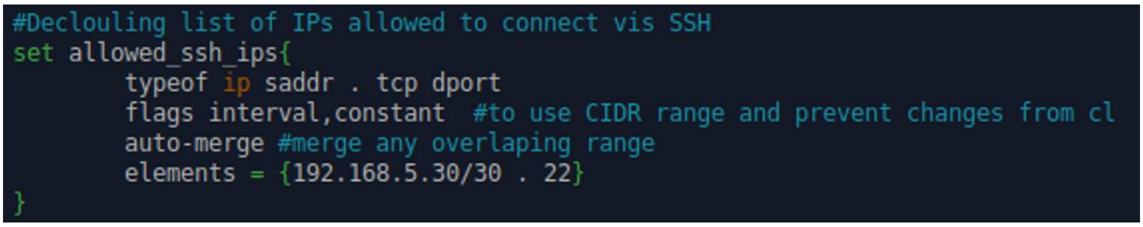
Capturing from enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.src == 192.168.5.190 or ip.dst == 192.168.5.190

No.     Time            Source                 Destination   Protocol Length Info
  3310… 60.738336263    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.738547572    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.738763028    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.738932100    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.739161676    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.739330406    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.739524019    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.739753421    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 60.739910875    192.168.5.190          192.168.5.1   SSH        174 Client: [TCP Port numbe
  3310… 287.320979016   fe80::2484:df50:cdc…   ff02::2       ICMPv6      62 Router Solicitation
  3310… 299.817132464   192.168.5.190          224.0.0.251   MDNS       183 Standard query 0x0000 P
  3310… 299.817442752   fe80::2484:df50:cdc…   ff02::fb      MDNS       203 Standard query 0x0000 P

▸ Frame 1: 174 bytes on wire (1392 bits), 174 bytes
▸ Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27
▸ Internet Protocol Version 4, Src: 192.168.5.190,
▸ Transmission Control Protocol, Src Port: 2067, Ds
▸ SSH Protocol

0000  08 00 27 1c c4 3a 08 00  27 8c 70 69 08 00 4
0010  00 a0 56 60 00 00 40 06  97 e8 c0 a8 05 be c
0020  05 01 08 13 00 16 57 34  47 0f 20 fe 08 7d 5
0030  00 40 9e 7e 00 00 58 58  58 58 58 58 58 58 5
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 5
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58

enp0s8: <live capture in progress>    Packets: 331074 · Displayed: 331074 (100.0%)    Profile: Default
```

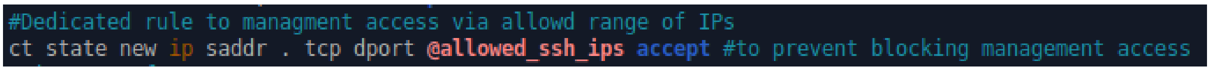
RESULT: PASS - 331053 packet dropped out of 331054 send by VM3-Attacker. Allowed 1 packet.

3.11. DOS on VM2-Firewall on port 22 using IP of privileged range - IP 192.168.5.30 used

Privileged range is:

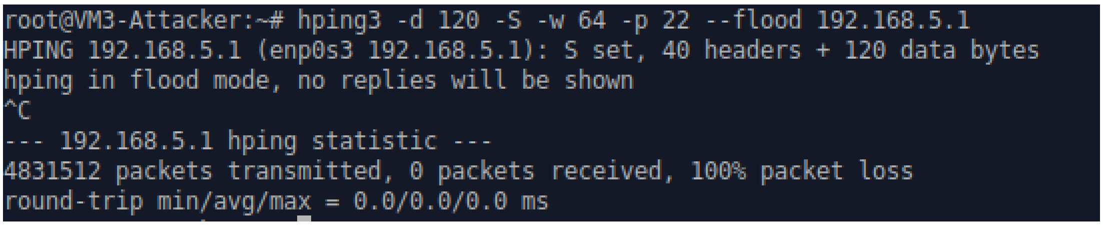


```
#Declouling list of IPs allowed to connect vis SSH
set allowed_ssh_ips{
        typeof ip saddr . tcp dport
        flags interval,constant  #to use CIDR range and prevent changes from cl
        auto-merge #merge any overlaping range
        elements = {192.168.5.30/30 . 22}
}
```



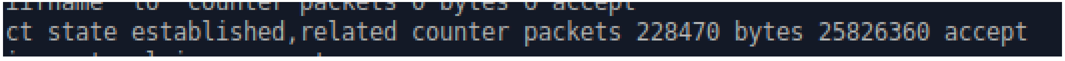
```
#Dedicated rule to managment access via allowd range of IPs
ct state new ip saddr . tcp dport @allowed_ssh_ips accept #to prevent blocking management access
```

Flood attack from VM3-Attacker sending 4,831,512packets:



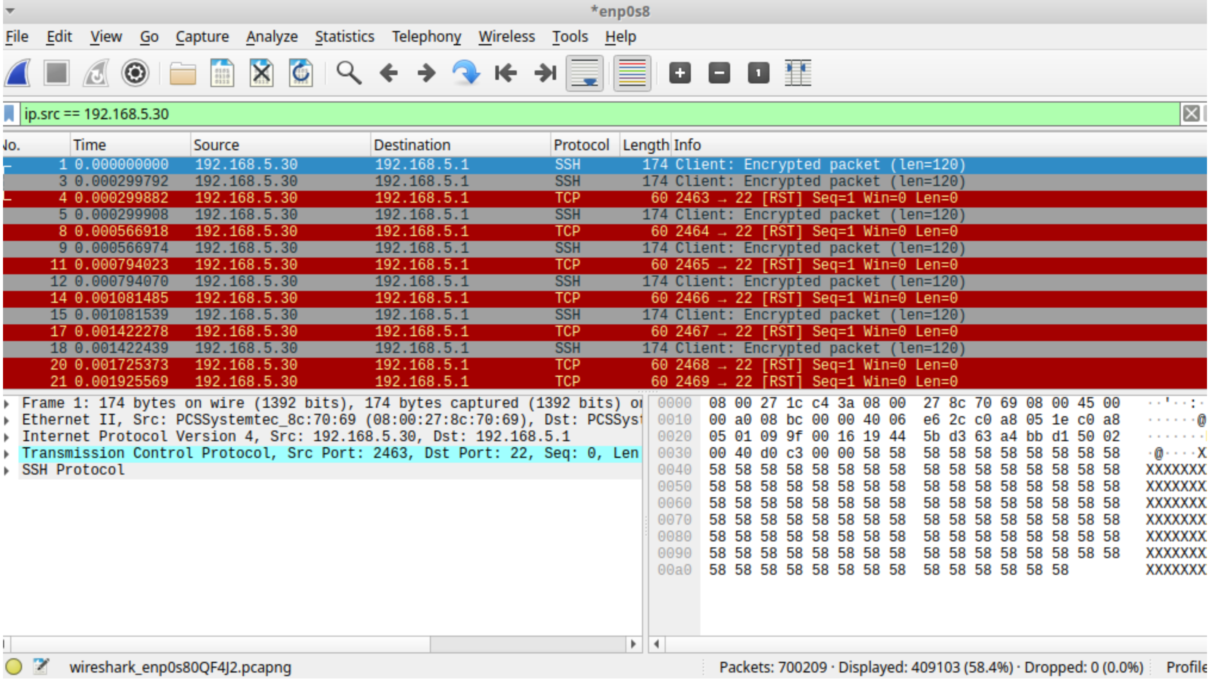
```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1
HPING 192.168.5.1 (enp0s3 192.168.5.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.5.1 hping statistic ---
4831512 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

There is an exception of admitted IP range on Nftable, therefore the entire traffic was allowed as it came from within such range:



```
iifname  lo  counter packets 0 bytes 0 accept
ct state established,related counter packets 228470 bytes 25826360 accept
```

Wireshark capture on VM-Firewall 192.168.5.1 registering incoming traffic:



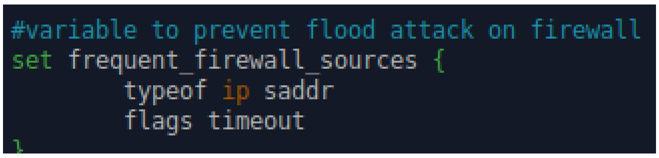
*enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.src == 192.168.5.30

No.   Time         Source         Destination    Protocol Length Info
    1 0.000000000  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
    3 0.000299792  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
    4 0.000299882  192.168.5.30   192.168.5.1    TCP       60 2463 → 22 [RST] Seq=1 Win=0 Len=0
    5 0.000299908  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
    8 0.000566918  192.168.5.30   192.168.5.1    TCP       60 2464 → 22 [RST] Seq=1 Win=0 Len=0
    9 0.000566974  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
   11 0.000794023  192.168.5.30   192.168.5.1    TCP       60 2465 → 22 [RST] Seq=1 Win=0 Len=0
   12 0.000794070  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
   14 0.001081485  192.168.5.30   192.168.5.1    TCP       60 2466 → 22 [RST] Seq=1 Win=0 Len=0
   15 0.001081539  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
   17 0.001422278  192.168.5.30   192.168.5.1    TCP       60 2467 → 22 [RST] Seq=1 Win=0 Len=0
   18 0.001422439  192.168.5.30   192.168.5.1    SSH      174 Client: Encrypted packet (len=120)
   20 0.001725373  192.168.5.30   192.168.5.1    TCP       60 2468 → 22 [RST] Seq=1 Win=0 Len=0
   21 0.001925569  192.168.5.30   192.168.5.1    TCP       60 2469 → 22 [RST] Seq=1 Win=0 Len=0

Frame 1: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) o
Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27:8c:70:69), Dst: PCSSys
Internet Protocol Version 4, Src: 192.168.5.30, Dst: 192.168.5.1
Transmission Control Protocol, Src Port: 2463, Dst Port: 22, Seq: 0, Len
SSH Protocol

0000  08 00 27 1c c4 3a 08 00  27 8c 70 69 08 00 45 00   ··'··:·
0010  00 a0 08 bc 00 00 40 06  e6 2c c0 a8 05 1e c0 a8   ······@
0020  05 01 09 9f 00 16 19 44  5b d3 63 a4 bb d1 50 02   ·······
0030  00 40 d0 c3 00 00 58 58  58 58 58 58 58 58 58 58   ·@····X
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXX
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58         XXXXXXX

wireshark_enp0s80QF4J2.pcapng    Packets: 700209 · Displayed: 409103 (58.4%) · Dropped: 0 (0.0%)    Profile
```

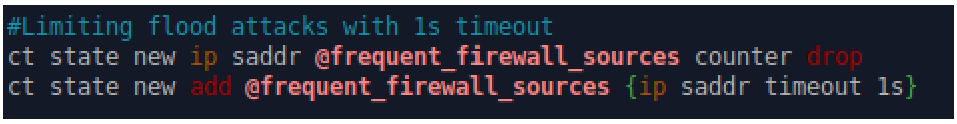
RESULT: FAIL 4,831,000 packets sent and 409,103 received resulting in total of 700,209 packets sent and received by VM2-Firewall

3.12. FIXED - DOS on VM2-Firewall on port 22 using IP of privileged range - IP 192.168.5.30 used

Test 3.12 was re-run with additional rule on input hook in VM2-Firewall:

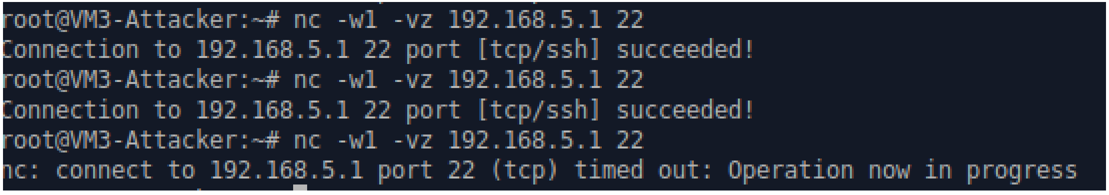


```
#variable to prevent flood attack on firewall
set frequent_firewall_sources {
        typeof ip saddr
        flags timeout
}
```



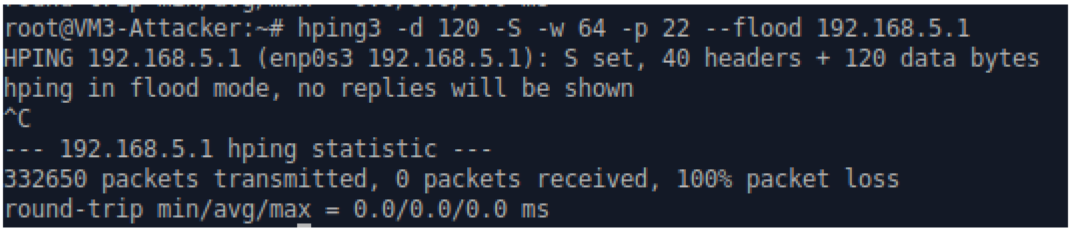
```
#Limiting flood attacks with 1s timeout
ct state new ip saddr @frequent_firewall_sources counter drop
ct state new add @frequent_firewall_sources {ip saddr timeout 1s}
```

Verified that connection is possible from VM3-Attacker but blocked on 3rd connection:



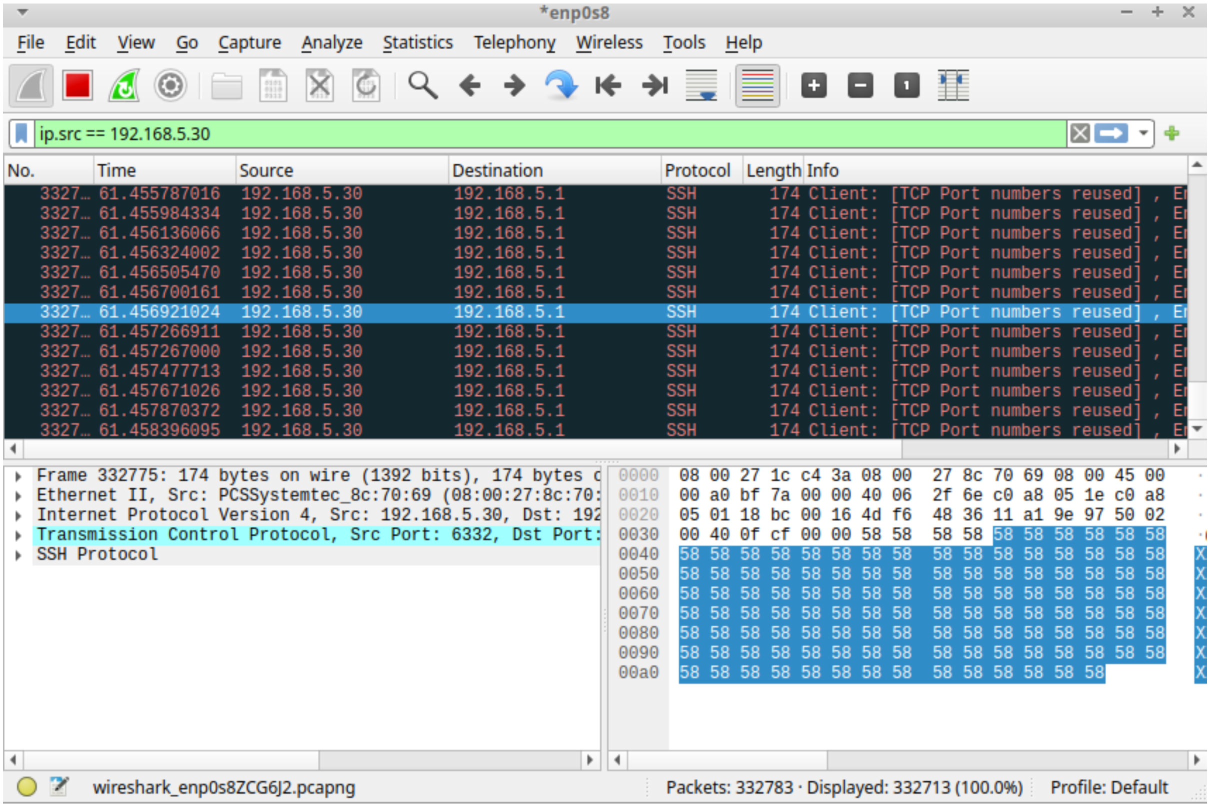
```
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
Connection to 192.168.5.1 22 port [tcp/ssh] succeeded!
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
Connection to 192.168.5.1 22 port [tcp/ssh] succeeded!
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
nc: connect to 192.168.5.1 port 22 (tcp) timed out: Operation now in progress
```

I reset the timeouts. Flood attack from VM3-Attacker sending 332,650 packets:



```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1
HPING 192.168.5.1 (enp0s3 192.168.5.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.5.1 hping statistic ---
332650 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Wireshark capture on VM-Firewall 192.168.5.1 **registering incoming traffic:**



ip.src == 192.168.5.30

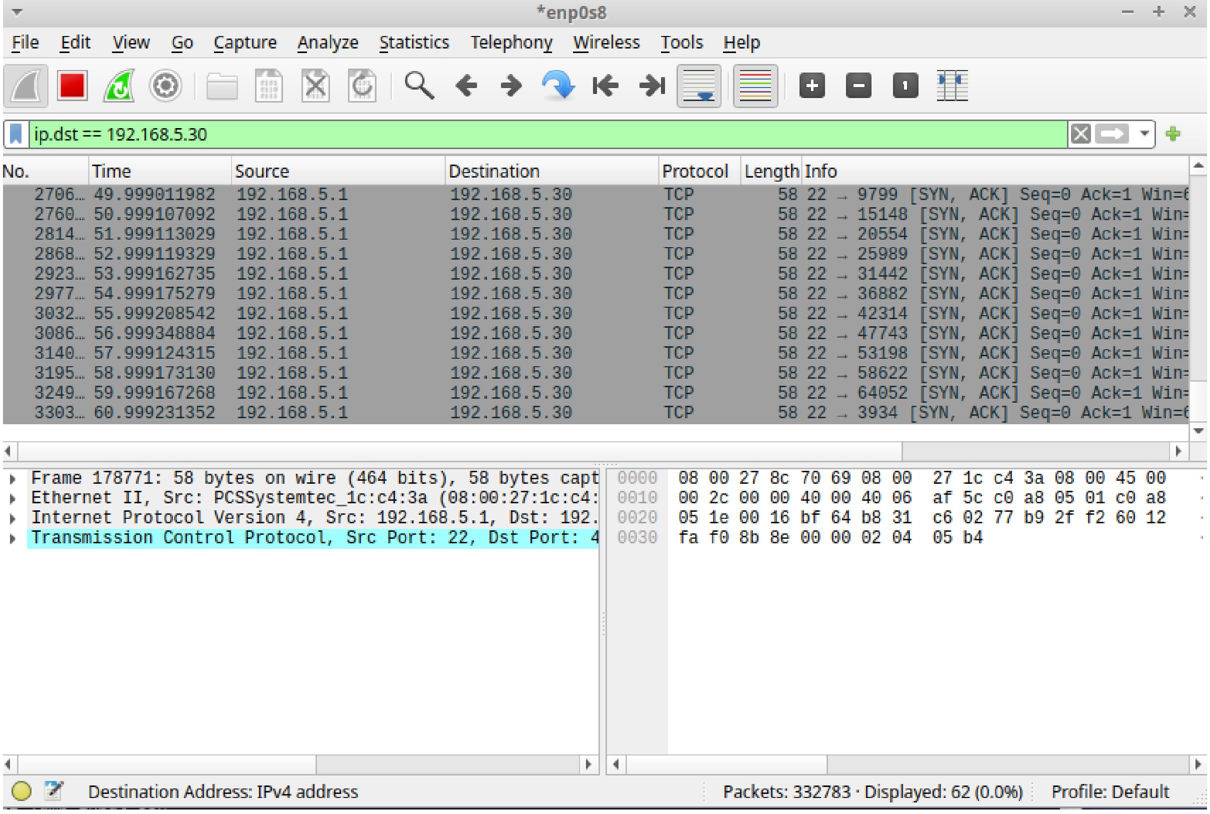
```
No.      Time          Source         Destination    Protocol  Length Info
  3327… 61.455787016  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.455984334  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.456136066  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.456324002  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.456505470  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.456700161  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.456921024  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.457266911  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.457267000  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.457477713  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.457671026  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.457870372  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
  3327… 61.458396095  192.168.5.30   192.168.5.1    SSH          174 Client: [TCP Port numbers reused] , En
```

```
▸ Frame 332775: 174 bytes on wire (1392 bits), 174 bytes c
▸ Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27:8c:70:
▸ Internet Protocol Version 4, Src: 192.168.5.30, Dst: 192
▸ Transmission Control Protocol, Src Port: 6332, Dst Port:
▸ SSH Protocol
```

```
0000  08 00 27 1c c4 3a 08 00  27 8c 70 69 08 00 45 00
0010  00 a0 bf 7a 00 00 40 06  2f 6e c0 a8 05 1e c0 a8
0020  05 01 18 bc 00 16 4d f6  48 36 11 a1 9e 97 50 02
0030  00 40 0f cf 00 00 58 58  58 58 58 58 58 58 58 58
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58
```

wireshark_enp0s8ZCG6J2.pcapng   Packets: 332783 · Displayed: 332713 (100.0%)   Profile: Default

Wireshark capture on VM-Firewall 192.168.5.1 registering outgoing traffic:


*enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

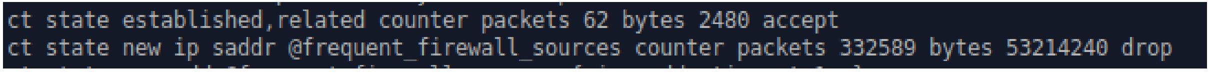
ip.dst == 192.168.5.30

```
No.      Time          Source         Destination     Protocol Length Info
   2706… 49.999011982  192.168.5.1    192.168.5.30    TCP         58 22 → 9799 [SYN, ACK] Seq=0 Ack=1 Win=6
   2760… 50.999107092  192.168.5.1    192.168.5.30    TCP         58 22 → 15148 [SYN, ACK] Seq=0 Ack=1 Win=
   2814… 51.999113029  192.168.5.1    192.168.5.30    TCP         58 22 → 20554 [SYN, ACK] Seq=0 Ack=1 Win=
   2868… 52.999119329  192.168.5.1    192.168.5.30    TCP         58 22 → 25989 [SYN, ACK] Seq=0 Ack=1 Win=
   2923… 53.999162735  192.168.5.1    192.168.5.30    TCP         58 22 → 31442 [SYN, ACK] Seq=0 Ack=1 Win=
   2977… 54.999175279  192.168.5.1    192.168.5.30    TCP         58 22 → 36882 [SYN, ACK] Seq=0 Ack=1 Win=
   3032… 55.999208542  192.168.5.1    192.168.5.30    TCP         58 22 → 42314 [SYN, ACK] Seq=0 Ack=1 Win=
   3086… 56.999348884  192.168.5.1    192.168.5.30    TCP         58 22 → 47743 [SYN, ACK] Seq=0 Ack=1 Win=
   3140… 57.999124315  192.168.5.1    192.168.5.30    TCP         58 22 → 53198 [SYN, ACK] Seq=0 Ack=1 Win=
   3195… 58.999173130  192.168.5.1    192.168.5.30    TCP         58 22 → 58622 [SYN, ACK] Seq=0 Ack=1 Win=
   3249… 59.999167268  192.168.5.1    192.168.5.30    TCP         58 22 → 64052 [SYN, ACK] Seq=0 Ack=1 Win=
   3303… 60.999231352  192.168.5.1    192.168.5.30    TCP         58 22 → 3934 [SYN, ACK] Seq=0 Ack=1 Win=6
```

```
▸ Frame 178771: 58 bytes on wire (464 bits), 58 bytes capt    0000  08 00 27 8c 70 69 08 00  27 1c c4 3a 08 00 45 00
▸ Ethernet II, Src: PCSSystemtec_1c:c4:3a (08:00:27:1c:c4:    0010  00 2c 00 00 40 00 40 06  af 5c c0 a8 05 01 c0 a8
▸ Internet Protocol Version 4, Src: 192.168.5.1, Dst: 192.    0020  05 1e 00 16 bf 64 b8 31  c6 02 77 b9 2f f2 60 12
▸ Transmission Control Protocol, Src Port: 22, Dst Port: 4    0030  fa f0 8b 8e 00 00 02 04  05 b4
```

Destination Address: IPv4 address   Packets: 332783 · Displayed: 62 (0.0%)   Profile: Default

Received on VM2-Firewall and blocked by rule:

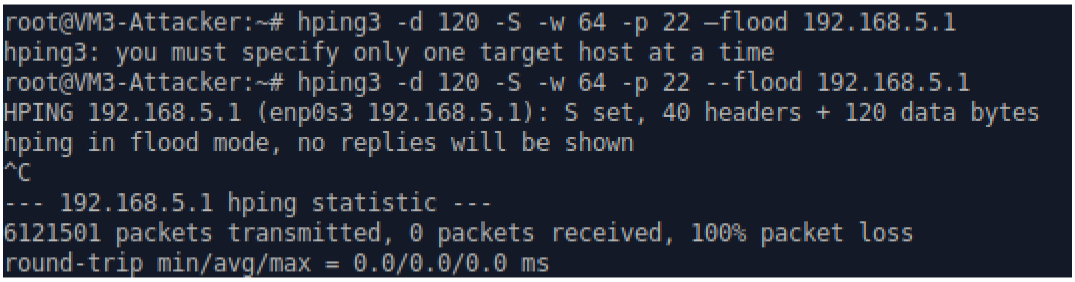


```
ct state established,related counter packets 62 bytes 2480 accept
ct state new ip saddr @frequent_firewall_sources counter packets 332589 bytes 53214240 drop
```

RESULT: PASS - 332650 packets transmitted by VM3-Attaccker and only 62 packets processed by VM2-Firewall

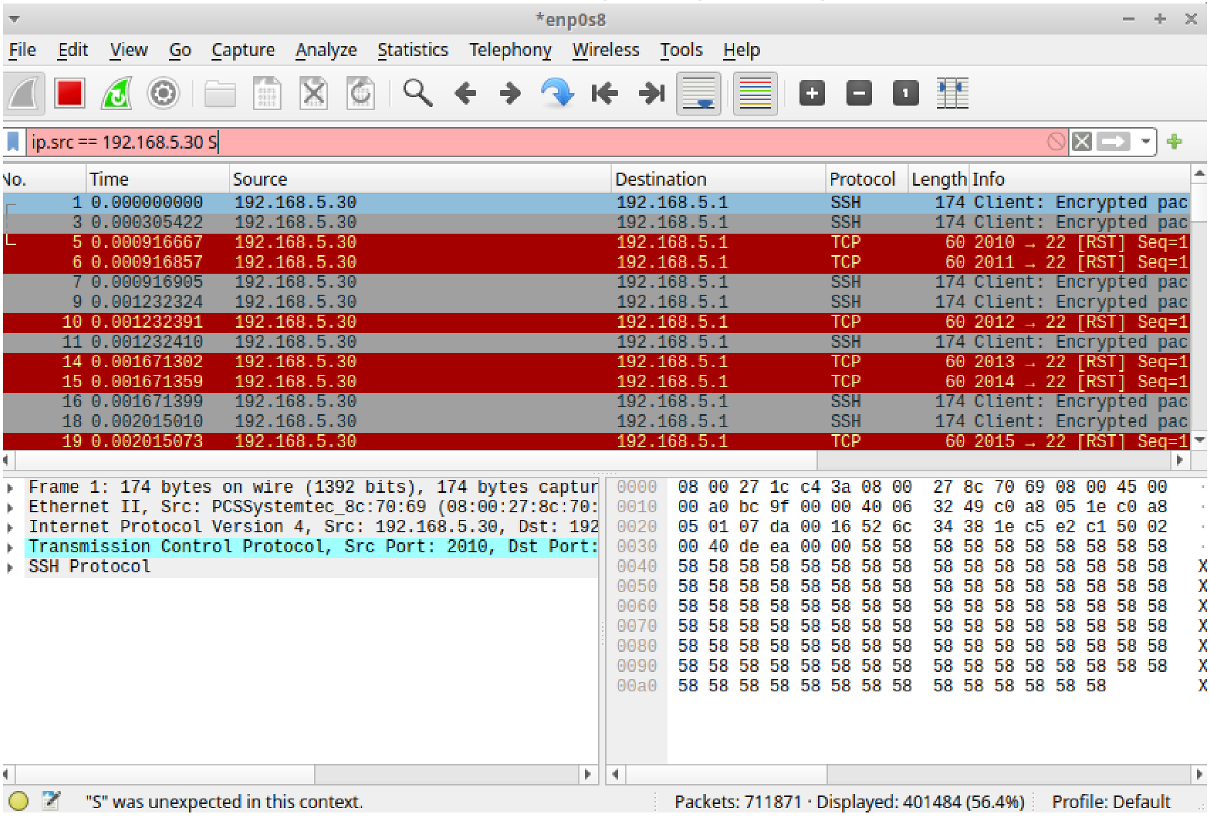
3.13. DOS on VM2-Firewall on port 22 using IP of privileged range - IP 192.168.5.30 used with Nftable off

Flood attack from VM3-Attacker sending 6,121,501 packets:



```
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 22 –flood 192.168.5.1
hping3: you must specify only one target host at a time
root@VM3-Attacker:~# hping3 -d 120 -S -w 64 -p 22 --flood 192.168.5.1
HPING 192.168.5.1 (enp0s3 192.168.5.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.5.1 hping statistic ---
6121501 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Wireshark capture on VM-Firewall 192.168.5.1 **registering incoming traffic:**



```
ip.src == 192.168.5.30 S
```

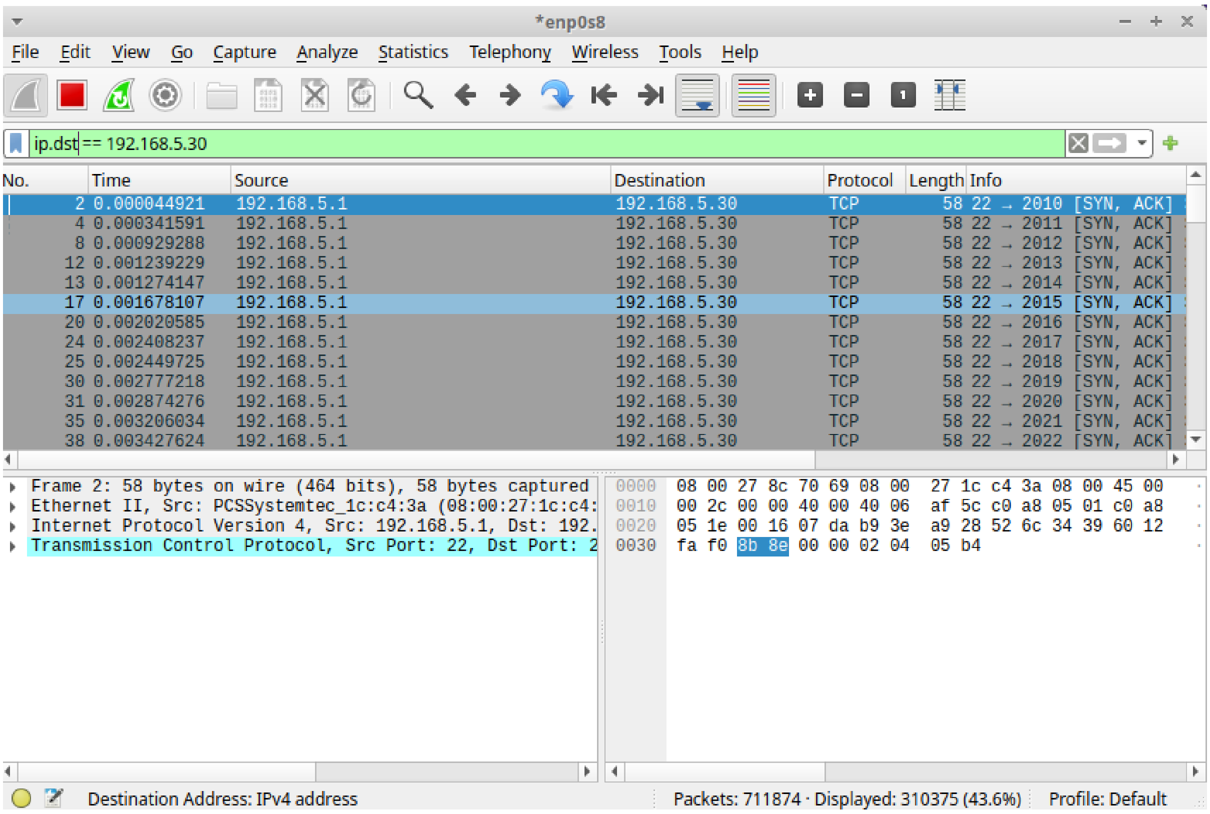
```
No.  Time         Source          Destination   Protocol Length Info
   1 0.000000000  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
   3 0.000305422  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
   5 0.000916667  192.168.5.30    192.168.5.1   TCP       60 2010 → 22 [RST] Seq=1
   6 0.000916857  192.168.5.30    192.168.5.1   TCP       60 2011 → 22 [RST] Seq=1
   7 0.000916905  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
   9 0.001232324  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
  10 0.001232391  192.168.5.30    192.168.5.1   TCP       60 2012 → 22 [RST] Seq=1
  11 0.001232410  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
  14 0.001671302  192.168.5.30    192.168.5.1   TCP       60 2013 → 22 [RST] Seq=1
  15 0.001671359  192.168.5.30    192.168.5.1   TCP       60 2014 → 22 [RST] Seq=1
  16 0.001671399  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
  18 0.002015010  192.168.5.30    192.168.5.1   SSH      174 Client: Encrypted pac
  19 0.002015073  192.168.5.30    192.168.5.1   TCP       60 2015 → 22 [RST] Seq=1
```

```
Frame 1: 174 bytes on wire (1392 bits), 174 bytes captur
Ethernet II, Src: PCSSystemtec_8c:70:69 (08:00:27:8c:70:
Internet Protocol Version 4, Src: 192.168.5.30, Dst: 192
Transmission Control Protocol, Src Port: 2010, Dst Port:
SSH Protocol
```

```
0000  08 00 27 1c c4 3a 08 00  27 8c 70 69 08 00 45 00
0010  00 a0 bc 9f 00 00 40 06  32 49 c0 a8 05 1e c0 a8
0020  05 01 07 da 00 16 52 6c  34 38 1e c5 e2 c1 50 02
0030  00 40 de ea 00 00 58 58  58 58 58 58 58 58 58 58
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   X
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58         X
```

"S" was unexpected in this context. Packets: 711871 · Displayed: 401484 (56.4%) Profile: Default

Wireshark capture on VM-Firewall 192.168.5.1 **registering outgoing traffic:**



```
ip.dst == 192.168.5.30
```

```
No.  Time         Source          Destination   Protocol Length Info
   2 0.000044921  192.168.5.1     192.168.5.30  TCP       58 22 → 2010 [SYN, ACK]
   4 0.000341591  192.168.5.1     192.168.5.30  TCP       58 22 → 2011 [SYN, ACK]
   8 0.000929288  192.168.5.1     192.168.5.30  TCP       58 22 → 2012 [SYN, ACK]
  12 0.001239229  192.168.5.1     192.168.5.30  TCP       58 22 → 2013 [SYN, ACK]
  13 0.001274147  192.168.5.1     192.168.5.30  TCP       58 22 → 2014 [SYN, ACK]
  17 0.001678107  192.168.5.1     192.168.5.30  TCP       58 22 → 2015 [SYN, ACK]
  20 0.002020585  192.168.5.1     192.168.5.30  TCP       58 22 → 2016 [SYN, ACK]
  24 0.002408237  192.168.5.1     192.168.5.30  TCP       58 22 → 2017 [SYN, ACK]
  25 0.002449725  192.168.5.1     192.168.5.30  TCP       58 22 → 2018 [SYN, ACK]
  30 0.002777218  192.168.5.1     192.168.5.30  TCP       58 22 → 2019 [SYN, ACK]
  31 0.002874276  192.168.5.1     192.168.5.30  TCP       58 22 → 2020 [SYN, ACK]
  35 0.003206034  192.168.5.1     192.168.5.30  TCP       58 22 → 2021 [SYN, ACK]
  38 0.003427624  192.168.5.1     192.168.5.30  TCP       58 22 → 2022 [SYN, ACK]
```

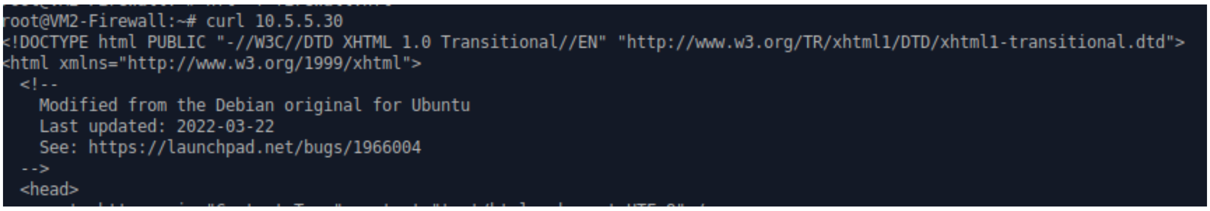
```
Frame 2: 58 bytes on wire (464 bits), 58 bytes captured
Ethernet II, Src: PCSSystemtec_1c:c4:3a (08:00:27:1c:c4:
Internet Protocol Version 4, Src: 192.168.5.1, Dst: 192.
Transmission Control Protocol, Src Port: 22, Dst Port: 2
```

```
0000  08 00 27 8c 70 69 08 00  27 1c c4 3a 08 00 45 00
0010  00 2c 00 00 40 00 40 06  af 5c c0 a8 05 01 c0 a8
0020  05 1e 00 16 07 da b9 3e  a9 28 52 6c 34 39 60 12
0030  fa f0 8b 8e 00 00 02 04  05 b4
```

Destination Address: IPv4 address Packets: 711874 · Displayed: 310375 (43.6%) Profile: Default

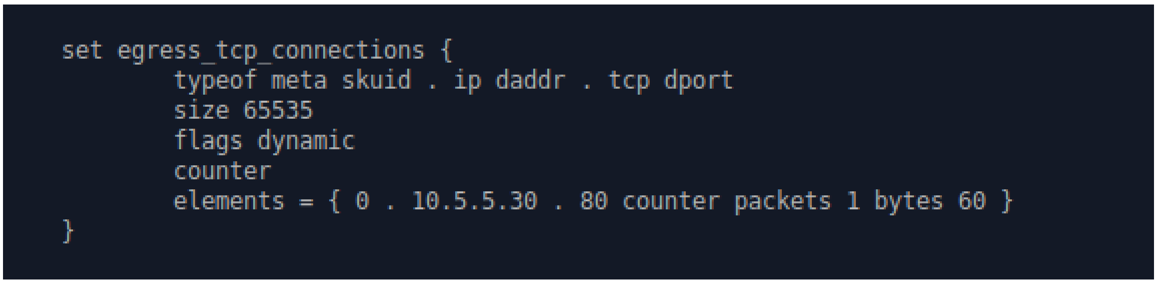
RESULT: FAIL - 6,121,501 packets transmitted by VM3-Attaccker and 401,484 packets accepted by VM2-Firewall

3.14. Triggering egress hook by accessing web server

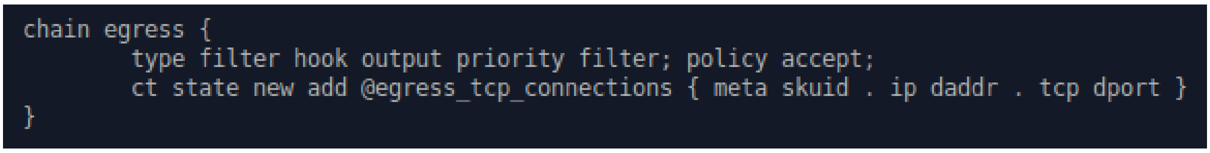


```
root@VM2-Firewall:~# curl 10.5.5.30
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
```

Triggered egress hook:



```
    set egress_tcp_connections {
            typeof meta skuid . ip daddr . tcp dport
            size 65535
            flags dynamic
            counter
            elements = { 0 . 10.5.5.30 . 80 counter packets 1 bytes 60 }
    }
```



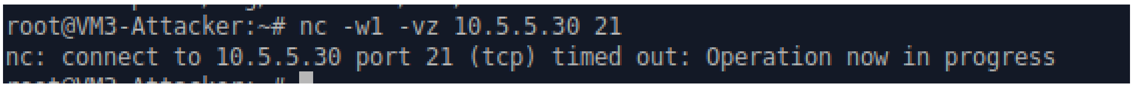
```
chain egress {
        type filter hook output priority filter; policy accept;
        ct state new add @egress_tcp_connections { meta skuid . ip daddr . tcp dport }
}
```

RESULT: PASS – Egress hook triggered

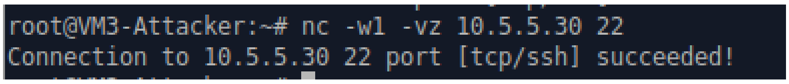
3.15. Connecting to FTP (hidden open port) & SSH

Both FTP and SSH ports are open as discovered by nmap in test 3.2 & 3.4 when NFtable was off.

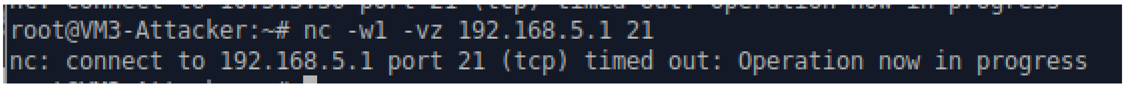
Tests:



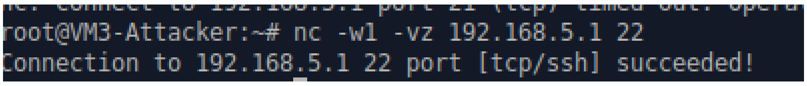
```
root@VM3-Attacker:~# nc -w1 -vz 10.5.5.30 21
nc: connect to 10.5.5.30 port 21 (tcp) timed out: Operation now in progress
```



```
root@VM3-Attacker:~# nc -w1 -vz 10.5.5.30 22
Connection to 10.5.5.30 22 port [tcp/ssh] succeeded!
```



```
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 21
nc: connect to 192.168.5.1 port 21 (tcp) timed out: Operation now in progress
```



```
root@VM3-Attacker:~# nc -w1 -vz 192.168.5.1 22
Connection to 192.168.5.1 22 port [tcp/ssh] succeeded!
```

RESULT: PASS NFTable blocked open FTP port and allowed SSH traffic

4. Conclusions

The assignment involved developing a testbed scenario involving 3VMs with Linux OS simulating an attacker, and a firewall in between the communication and server.

In the first stage, I had to configure all VMs to enable FTP, SSH and Web services and communication through the VM2-Fierwall.

Subsequently, I learned about and configured NFtables to establish a robust firewall, implemented security measures, and finally conducted various tests. I developed key components such as port filtering, SSH & Web access control, a prerouting hook for Destination Network Address Translation (DNAT), and an egress rule.

In my project, I designed the ruleset to prevent flooding attacks, managed timeouts for SSH connections, defined privileged SSH access for management, and developed a forwarding hook for specific incoming traffic to a designated server.

In the testing stage, I covered scenarios such as port scans, SSH flood attacks depending on source IP, Web flood attacks, IPv6 accessibility, connecting via Netcat and egress hook triggering.

Continuous testing and refinement are crucial to adapting to evolving threats and ensuring the effectiveness of the firewall configuration. As I mentioned before in the report, further development might consist of developing Honeypots, Logging and Analysis, Geo-IP Filtering, Load Balancing.

Certainly, the most time-consuming was developing firewall logic and troubleshooting/debugging including in the testing stage.

5. Source editable NFTable:

```
  GNU nano 6.2                                  firewall.nft
flush ruleset
table ip Server{

#Variables:
    #Decoupling list of IPs allowed to connect via SSH
    set allowed_ssh_ips{
        typeof ip saddr . tcp dport
        flags interval,constant  #to use CIDR range and prevent changes from cl
        auto-merge #merge any overlaping range
        elements = {192.168.5.30/30 . 22}
    }

    #Counter variables
    counter counter_ct_web{
    }
    counter counter_ct_ssh{
    }
    #3x timeout variables
    set timeout1{
        typeof ip saddr
        flags timeout
    }
    set timeout2{
        typeof ip saddr
        flags timeout
    }
    set timeout3{
        typeof ip saddr
        flags timeout
    }

    #Allowed  server ports
    set allowed_server_ports{
        typeof tcp dport
        elements = {22, 80, 443}
    }

    #Variable to prevent flood attack on server
    set frequent_server_sources {
        typeof ip saddr
        flags timeout
    }

    #variable to prevent flood attack on firewall
    set frequent_firewall_sources {
        typeof ip saddr
        flags timeout
    }

#CHAINS:
    chain PreroutingServer{
        type nat hook prerouting priority 0;
        #Rule to forward only specified IP - implemented for testing purposes
        iifname "enp0s8" ip saddr 192.168.5.30 tcp dport {80, 443 } counter dnat to 10.5.5.30:80-
443
        #continue to statefull packet examination
    }


    chain Serverforward {
        type filter hook forward priority 0;
        # Permit established and related SSH connections
        ct state established, related accept

        #Limiting new flood attack with 1s timeout
        ct state new ip saddr @frequent_server_sources counter drop
        ct state new add @frequent_server_sources {ip saddr timeout 1s}
        #Forawrding web services
        ct state new tcp dport {80, 443 }counter name counter_ct_web accept
        #Accepting icmp
        ip protocol icmp accept
        #Droping malicious/invalid packets
        ct state invalid counter drop

        #JUMP to chain dealing with SSH port 22
        tcp dport 22 jump input_ssh

        #Droping any other traffic
        counter  drop
    }

  chain input_firewall {
        type filter hook input priority filter; policy accept;
         iifname lo counter accept
        #if a contract receives a SYN packet it considers connection as new , SYN-ACK considers
as established
        ct state established, related counter accept

        #Limiting flood attacks with 1s timeout
        ct state new ip saddr @frequent_firewall_sources counter drop
        ct state new add @frequent_firewall_sources {ip saddr timeout 1s}

        #Accepting icmp
        ip protocol icmp accept
        #Droping malicious/invalid packets
        ct state invalid counter drop

        #JUMP to chain dealing with SSH port 22
        tcp dport 22 jump input_ssh

        #Droping any other traffic
        counter drop
    }
    chain input_ssh{
        # Permit established and related SSH connections
        ct state established, related accept
        #Dedicated rule to managment access via allowed range of IPs -it will accept unlimited connections but not more often
        #then every 1s as per limiting flood attack rule in input_firewall chain
        ct state new ip saddr . tcp dport @allowed_ssh_ips counter accept #to prevent blocking management access
        #Time-outs for SSH
        ct state new ip saddr @timeout2 tcp dport 22 add @timeout3 {ip saddr timeout 3d}
        ct state new ip saddr @timeout1 tcp dport 22 add @timeout2 {ip saddr timeout 3m}
        ct state new tcp dport 22 add @timeout1 {ip saddr timeout 1m}
        ct state new ip saddr @timeout3 tcp dport 22 counter drop
        #Only SYN packet will match this rule
        ct state new tcp dport 22 counter name counter_ct_ssh accept

    }

#OUTPUT
    #variable to track outgoing connections:
    set egress_tcp_connections{
        typeof meta skuid . ip daddr . tcp dport;
        counter;
    }
    chain egress {
        type filter hook output priority filter; policy accept;
        ct state new ip protocol tcp add @egress_tcp_connections {meta skuid . ip daddr . tcp dport}
    }

}
```

6. References:

1. Suehring S., (2015) Linux firewalls: enhancing security with nftables and beyond 4th edition, Addison-Wesley
2. Shirvar A., (2020) ‘A comprehensive guide to Nftables (A leading packet filtering framework for Linux)’ in linkedin.com [online]. Available at: https://www.linkedin.com/pulse/comprehensive-guide-nftables-leading-packet-filtering-arash-shirvar [accessed 26 February 2024]
3. SW Team, (2024) ‘Nftables: a deep look at linux security’ in swhosting.com [online]. Available at: https://www.swhosting.com/en/blog/nftables-a-deep-look-at-linux-security [accessed 26 February 2024]
4. Easillyy, (n.d.) ‘a comprehensive guide to nftables’ in easillyy.com [online]. Available at: http://easillyy.com/a-comprehensive-guide-to-nftables/ [accessed 26 February 2024]
5. Wiki, (2021) ‘What is nftable?’ in wiki.nftables.org [online]. Available at: https://wiki.nftables.org/wiki-nftables/index.php/What_is_nftables%3F [accessed 27 February 2024]
6. LinuxCloudHacks (2023) Mastering Nftables Sets: A Comprehensive Guide Available at: https://www.youtube.com/watch?v=YLVKuA4kiMA [accessed 28 February 2024]
7. LinuxCloudHacks (2023) Protecting Incoming Traffic with Nftables Available at: https://www.youtube.com/watch?v=K8JPwbcNy_0 [accessed 28 February 2024]
8. LinuxCloudHacks (2023) Unleashing the Power of Nftables Chains and Verdict Maps Available at: https://www.youtube.com/watch?v=Qy52m5hXiSg [accessed 28 February 2024]
9. Wiki (2018) ‘Rate limiting matchings’ in wiki.nftables.org [online]. Available at: https://wiki.nftables.org/wiki-nftables/index.php/Rate_limiting_matchings [accessed 29 February 2024]
10. ZacsTech (2022) How to install Wireshark on Ubuntu 22.04 | 20.04 LTS Available at: https://www.youtube.com/watch?v=lPxuiOLCtOY [accessed 29 February 2024]