



Assignment 1 – Open-Source Firewalls

**Abel Melinte
B00137882**

**Adam Giembicki
B00134937**

*Department of Informatics,
School of Informatics and Engineering,
Technological University Dublin,
Dublin 15*

[Page Count: 20

Word Count: 2195]

**Digital Forensics and Cyber Security
Network Security
13/02/2023**

Contents

1.Introduction	3
2.Firewalls	4
2.1 Nftables	4
2.1.1 Nftables Rules	4
2.1.2 Nftables Chains	4
2.2 IPFire	5
2.2.1 Features	5
3.Attacking	5
3.1 SYN Flood	5
3.1.1 What is the procedure behind a SYN Flood Attack?	6
3.2 SSH Brute Force Attack	6
3.2.1 SSH Brute Force Attack	6
3.3 Nmap.....	6
4. Procedures	7
4.1 VM Setup.....	7
4.2 Network Setup	7
4.3 What attacks were used?.....	8
4.4 Attacking without a firewall configured.	9
4.4.1 SYN Flood	9
4.4.2 Ping.....	9
4.4.3 Nmap Scan	10
4.4.4 SSH Brute Force.....	11
4.5 Attacking with Nftables firewall configured.	13
4.5.1 Ping (Nftables).....	13
4.5.2 Nmap Scan	14
4.5.3 SSH Brute Force.....	14
4.6 Attacking with IPFire enabled.	15
4.6.1 SSH Brute Force.....	15
4.6.2 Ping.....	17
5. Comparison and Conclusion	19
6. References	20

1.Introduction

In this assignment we were given a task for us to complete, the task was to go ahead with researching and testing any two open source firewalls that we could find, we had to go and find one as we were given a firewall called “Nftables”.

The goal to for assignment is to make us understand how firewalls operate and also how we can configure them ourselves. The plan was to setup some virtual machines and install Kali Linux and Ubuntu on two separate virtual machines.

Victim

Ubuntu was set up as a victim in this operation.

Attacker

Kali Linux was set up to be the attacker in this operation.

The tests that were planned to go ahead with were. Nmap, Angry IP Scanner, Nemesis (Port Scanning Category), SSH, FTP, ARP Attack, DoS (hping3, LOIC, Ping of Death, SMURF).

Firewalls

Nftables & Pfsense

2.Firewalls

The plan for this assignment is to go ahead with two firewalls, these firewalls are Nftables and Pfsense.

2.1 Nftables

Nftables is an open-source firewall tool that can be downloaded and accessed by anyone at any time. It provides a new packet filtering framework, a new user-space utility, also known as (nft), and a compatibility layer for (ip, ip6) tables.

Nftables contains three main components, a kernel implementation, the libnl netlink communication and the nftables user-space front-end. (ArchLinux 2023)

NFtables is mostly known to be extremely like iptables, the known difference between them is that nftables uses a different command line when operating, it also comes with a compatible layer that allows the user to run iptables syntax command-lines over the fresh nftables kernel framework. (Pablo 2016)

2.1.1 Nftables Rules

The rules within nftables take actions upon the network packets, e.g., accepting or dropping them based on if they match the specific criteria. Each “Rule” is made up of zero or more expressions which is followed by one or more statements. When it comes to expressions, expressions test whether if a packet meets the specific payload field or even a packet metadata. (Pablo 2016)

Expressions are generally evaluated from left to right, for example, whenever the first expression matches, then the following expression assesses and so on. If it manages to reach the last expression, the packet then matches every expression that was included in the rule and then the rule’s statements are executed. (Pablo 2016)

After the statements are executed, the statements could either:

- start setting the net filter mark.
- counting the packet
- they log of the packet.
- accepting or denying the packet or jumping to a different chain

Overall, with expressions, multiple statements are usually linearly set to be from left to right, one single rule can produce multiple actions with involving multiple statements. (Pablo 2016)

2.1.2 Nftables Chains

The reason why chains are part of nftables is because the chains purpose is to hold the Rules. When it comes to nftables, chains don’t come with built-in-chains. If the user doesn’t add a chain within

the net filter framework, any packets that will pass through those specific chains will not be dealt with by nftables. (ArchLinux 2023)

The chains come in two types:

Base Chain

Is an entry point where packets from a networking stack and a hook value is assigned.

Regular Chain

A chain that can be used as a jump target for better organization.

2.2 IPFire

IPFire is a open source firewall that can be installed on any network, from your home to even a data centre. IPFire is secure, rapid, and adaptable. It can also function as a VPN gateway, analyse data packets as it contains Intrusion Prevention System also known as (IPS), it also comes with multiple add-ons that adds extra functionality to itself. (Bernhard Bitsch 2023)

2.2.1 Features

- The IPfire firewall is extremely to use, yet powerful. A creation of groups of some networks, hosts and service gives allowance to a single rule for large parts of a network to be assigned in one go.
- The Quality of Service (QoS) maintains your internet to remain fast. Allocated the correct amount of bandwidth for censorious apps like VoIP calls is quickly done and that will sort you out with never having a bad service call or a slow-loading website.
- The Intrusion Prevention System provides detailed packet inspection, it checks the packets for any well-known malware, and it also detects any suspicious behaviour to make your network more protected and secure against any random attackers that try attack your personal system.
- When it comes to IPFires, the web proxy is one of its most powerful features. Whenever a client wants to access a website, they will be checked for the access, content can be reached to speed up any browsing and it can also cache an entire update for the active operating system. It also has a URL filter which can prevent students from universities/schools from accessing any adult/malicious websites.
- If there is a plan to keep your network secure, IPFire contains an internal DNS proxy which uses DNSSEC to filter any ongoing attacks. It caches the DNS responses so it can improve the performance and can use DNS-over-TLS also known as (DoT) to speak securely upon upstream name servers. (Bernhard Bitsch 2023)

3.Attacking

3.1 SYN Flood

A SYN Flood Attack is similar to a denial-of-service (DDoS) attack, the aim for a SYN Flood attack is to make a targeted server unavailable to the public. When sending multiple requests packets over and over, the attacker has power over the available ports on the targeted server system, this causes the

system to struggle on replying to oncoming traffic or even not responding to any incoming traffic. (Cloudflare, Inc. 2020)

3.1.1 What is the procedure behind a SYN Flood Attack?

Overall, SYN Flood Attacks functions by exploiting the handshake of a TCP connection. TCP connection. In normal conditions, TCP reveals three processes for a connection to be made. (Cloudflare, Inc. 2020)

Processes:

- Client forwards a SYN packet to a targeted server in order to make a connection.
- The targeted server then responds back to the Client with a SYN/ACK packet, this happens so the server knows that it's communicating with a client.
- In the final step, the client starts by returning the ACK packet accept the presence of the packet from the targeted server. After it's finally finished with the packet receiving and sending, the TCP connection is finally open, and it can receive data and also send the data. (Cloudflare, Inc. 2020)

3.2 SSH Brute Force Attack

Brute Force attacks are a means of username and password or even hashes tokens that are stored and ready to gain unauthorized bypassed access to an account, a file or even protected data. A brute force attack can also be known as a trial-&-error-based attack method that functions by just attempting the guessing the login credentials, file paths or URLs. (ExtraHop Networks, Inc. 2020)

Attacker usually send out tools that contain malware so they can use the victim's machine as a victim within the brute force attack. There are some tools that are common to be used for brute forcing, these are "Hydra", "Chaos", "CrackMapExec", and "PoshC2". (ExtraHop Networks, Inc. 2020)

If an attacker completes his attack and its successful, the attacker then has access to all the data within the system such as financial information, hijacking the victim's system or malware spread. (ExtraHop Networks, Inc. 2020)

3.2.1 SSH Brute Force Attack

Secure Shell or SSH is known as a network protocol that allows encrypted communication across insecure networks. SSH is mainly used for when a user wants to remote login, command line executions, file transfer and more. The SSH Brute Force attacks are mainly achieved when an attacker tries a basic username and password across multiple servers until it finally matches with what they need. (ExtraHop Networks, Inc. 2020)

3.3 Nmap

Nmap shortened for "Network Scanner" is an open source tool which allows the user to do network exploration and security auditing. Nmap was created to perform rapidly scans of large networks, although it works smooth even against single hosts. Nmap works with raw IP packets in novel ways to know what hosts are currently available on the specific scanned network, what operating system they are running, what services the hosts are offering, what sort of packet firewalls/filters are in current use, and much more characteristics. (Nmap 2023)

Whenever someone uses Nmap as a method of scanning ports, it displays to them a list of scanned targets, with in depth detail on the depending option the user decides to choose. There are four states, these are "open", "filtered", "closed", "unfiltered". (Nmap 2023)

- Open labels as the application on the machine is listening for any packets/connections on that specific port.
- Filtered means that a firewall, filter or other network obstacle is blocking a specific port, allows Nmap to display to the user whether the port is open or closed.
- Close ports have no application listening on them, they could possibly open at any moment.
- Unfiltered is ports that are responsive to the Nmap's probes, but the issue is that Nmap doesn't detect whether the port is open or closed.

While using Nmap to port scan, not only it can display in depth information about the network scans but it can also provide further information on the targets, this includes reverse DNS names, device types, MAC addresses and operating system guesses. (Nmap 2023)

4. Procedures

To perform these attacks, here is a step-by-step guide on how everything was done.

4.1 VM Setup

The Virtual Machine that has been used throughout this procedure was. VMware Workstation Pro 16

Website used to download VMware - <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>

The operating systems that were used was:

First Environment -

- Kali Linux 2022.3 - <https://www.kali.org/get-kali/#kali-virtual-machines>
- Ubuntu 22.04 - <https://ubuntu.com/download/desktop>

Second Environment –

- Kali Linux 2022.3 - <https://www.kali.org/get-kali/#kali-virtual-machines>
- Ubuntu 22.04 - <https://ubuntu.com/download/desktop>
- IPFire 2.27 - <https://www.ipfire.org/download/ipfire-2.27-core172>

Firewalls used –

- IPFire 2.27 - <https://www.ipfire.org/download/ipfire-2.27-core172>
- Nftables 1.0.6 - <https://www.nftables.org/>

4.2 Network Setup

For the network setup, it went forward with using “Host Only Network”, this created a private VPN on the useable host machine.

Assigned IP Addresses for environment 1:

Kali Linux – 192.168.85.128

Ubuntu 1 – 192.168.85.129

Assignment 1 – Open-Source Firewalls

Again, for the second environment Host Only Network was used with a private LAN segment.

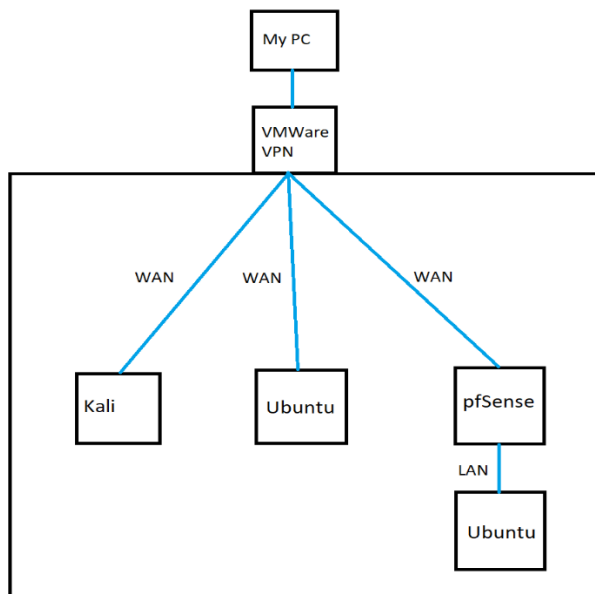
Assigned IP Addresses for environment 2:

Kali Linux – 192.168.85.129

IPFire Firewall – 192.168.85.133

Private LAN segment – 192.168.1.1/24

Second Ubuntu WM – 192.168.1.100



4.3 What attacks were used?

For Scanning & Pining, two methods were used, these are:

- Nmap Scan (Stealth)
- Ping

For Denial-of-Service (DoS) attacks, one method was used:

- SYN Flood

Here are the tools and commands that were used to work out these attacks:

- Nmap
- Ping
- Hping3
- Hydra

Assignment 1 – Open-Source Firewalls

4.4 Attacking without a firewall configured.

4.4.1 SYN Flood

Throughout the attacks, a tool that was used to be able to perform network troubleshooting and analysis of the packets. This tool is strong as it shows the user whenever there is a SYN Flood occurring, DoS and much more.

Performing SYN Flood attack while using the “hping3” tool.

```
(nsa@kalilinux-vm)-[~]  
$ sudo hping3 -c 10000 -d 120 -p 80 -S --flood --rand-source 192.168.85.129  
HPING 192.168.85.129 (eth0 192.168.85.129): S set, 40 headers + 120 data bytes
```

-c = counting response packets (10000)

-d = packet body size (120)

-p = port used (80)

-S = Syn

--flood = as fast as possible

--rand-source = random source mode

Display of a SYN flood attack in Wireshark

7212	13.254501	231.234.42.101	192.168.85.129	TCP	174 9743 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7213	13.254504	142.62.153.61	192.168.85.129	TCP	174 9744 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7214	13.254536	147.149.2.173	192.168.85.129	TCP	174 9745 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7215	13.254539	92.232.233.128	192.168.85.129	TCP	174 9746 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7216	13.254571	38.146.175.9	192.168.85.129	TCP	174 9747 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7217	13.254574	32.57.55.193	192.168.85.129	TCP	174 9748 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7218	13.254604	94.253.237.200	192.168.85.129	TCP	174 9749 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7219	13.254606	140.118.145.249	192.168.85.129	TCP	174 9750 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7220	13.254637	231.134.127.32	192.168.85.129	TCP	174 9751 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7221	13.254639	29.18.6.234	192.168.85.129	TCP	174 9752 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7222	13.254674	238.56.10.242	192.168.85.129	TCP	174 9753 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7223	13.254677	71.186.254.209	192.168.85.129	TCP	174 9754 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7224	13.254711	250.120.242.201	192.168.85.129	TCP	174 9755 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7225	13.254714	186.72.20.87	192.168.85.129	TCP	174 9756 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]
7226	13.254744	187.160.77.169	192.168.85.129	TCP	174 9757 → 80 [SYN] Seq=0 Win=512 Len=120 [TCP segment of a reassembled PDU]

4.4.2 Ping

Here is a simple ping command that allows us to see if the Kali machine can send request over to the Ubuntu machine.

```
(nsa@kalilinux-vm)-[~]  
$ ping 192.168.85.129  
PING 192.168.85.129 (192.168.85.129) 56(84) bytes of data.  
64 bytes from 192.168.85.129: icmp_seq=1 ttl=64 time=0.657 ms  
64 bytes from 192.168.85.129: icmp_seq=2 ttl=64 time=0.303 ms  
64 bytes from 192.168.85.129: icmp_seq=3 ttl=64 time=0.342 ms
```

Assignment 1 – Open-Source Firewalls

Wireshark displaying the incoming pings from the Kali to Ubuntu.

75	30.264132	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request	id=0x7058, seq=1/256, ttl=64 (reply in 76)
76	30.264228	192.168.85.129	192.168.85.128	ICMP	98 Echo (ping) reply	id=0x7058, seq=1/256, ttl=64 (request in 75)
77	31.000485	192.168.85.1	224.2.2.2	UDP	72 61713 → 8995	Len=30
78	31.274729	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request	id=0x7058, seq=2/512, ttl=64 (reply in 79)
79	31.274894	192.168.85.129	192.168.85.128	ICMP	98 Echo (ping) reply	id=0x7058, seq=2/512, ttl=64 (request in 78)
80	32.001059	192.168.85.1	224.2.2.2	UDP	72 61713 → 8995	Len=30
81	32.298980	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request	id=0x7058, seq=3/768, ttl=64 (reply in 82)
82	32.299160	192.168.85.129	192.168.85.128	ICMP	98 Echo (ping) reply	id=0x7058, seq=3/768, ttl=64 (request in 81)

4.4.3 Nmap Scan

How an Nmap scan is performed with Kali

```
(nsa@kalilinux-vm)-[~]
$ sudo nmap -sS -sV 192.168.85.129
[sudo] password for nsa:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-25 17:51 GMT
Nmap scan report for 192.168.85.129
Host is up (0.000089s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
MAC Address: 00:0C:29:84:65:4B (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 19.54 seconds
```

-sS = TCP SYN port scan (Default)

-sV = Determines what version of the service is running on the port

Image shows the attack of the Nmap scan in Wireshark.

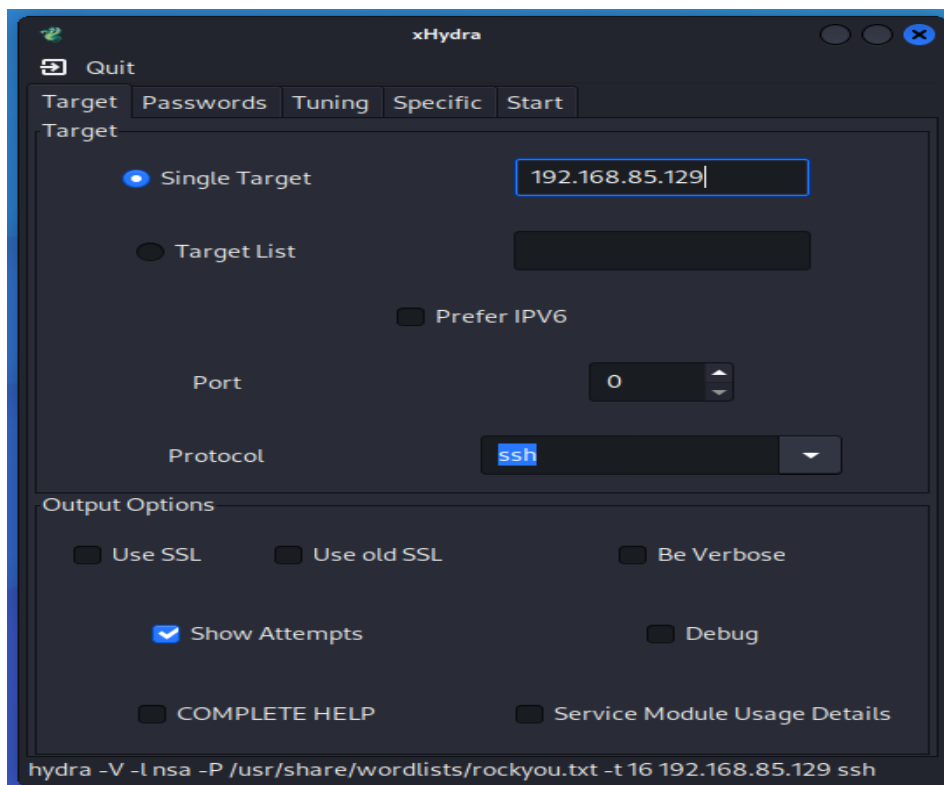
75	17.496779	192.168.85.129	192.168.85.128	TCP	60 3306 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
76	17.496799	192.168.85.129	192.168.85.128	TCP	60 587 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
77	17.496809	192.168.85.128	192.168.85.129	TCP	60 64114 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
78	17.496813	192.168.85.129	192.168.85.128	TCP	60 1723 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
79	17.496829	192.168.85.128	192.168.85.129	TCP	60 64114 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
80	17.496846	192.168.85.129	192.168.85.128	TCP	60 113 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
81	17.496846	192.168.85.128	192.168.85.129	TCP	60 64114 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
82	17.496863	192.168.85.128	192.168.85.129	TCP	60 64114 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
83	17.496864	192.168.85.129	192.168.85.128	TCP	60 1025 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
84	17.496879	192.168.85.129	192.168.85.128	TCP	60 993 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
85	17.496882	192.168.85.128	192.168.85.129	TCP	60 64114 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
86	17.496903	192.168.85.128	192.168.85.129	TCP	60 64114 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
87	17.496905	192.168.85.129	192.168.85.128	TCP	60 80 → 64114 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
88	17.496923	192.168.85.128	192.168.85.129	TCP	60 64114 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
89	17.496936	192.168.85.128	192.168.85.129	TCP	60 64114 → 80 [RST] Seq=1 Win=0 Len=0
90	17.496947	192.168.85.129	192.168.85.128	TCP	60 23 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
91	17.496954	192.168.85.128	192.168.85.129	TCP	60 64114 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
92	17.496969	192.168.85.129	192.168.85.128	TCP	60 143 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
93	17.496976	192.168.85.128	192.168.85.129	TCP	60 64114 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
94	17.497015	192.168.85.128	192.168.85.129	TCP	60 64114 → 1104 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
95	17.497058	192.168.85.128	192.168.85.129	TCP	60 64114 → 2811 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
96	17.497063	192.168.85.129	192.168.85.128	TCP	60 256 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
97	17.497135	192.168.85.129	192.168.85.128	TCP	60 21 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
98	17.497151	192.168.85.129	192.168.85.128	TCP	60 554 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
99	17.497194	192.168.85.129	192.168.85.128	TCP	60 22 → 64114 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
100	17.497217	192.168.85.129	192.168.85.128	TCP	60 445 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
101	17.497234	192.168.85.129	192.168.85.128	TCP	60 995 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
102	17.497248	192.168.85.129	192.168.85.128	TCP	60 1104 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
103	17.497262	192.168.85.129	192.168.85.128	TCP	60 2811 → 64114 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
104	17.497324	192.168.85.128	192.168.85.129	TCP	60 64114 → 22 [RST] Seq=1 Win=0 Len=0

Assignment 1 – Open-Source Firewalls

4.4.4 SSH Brute Force

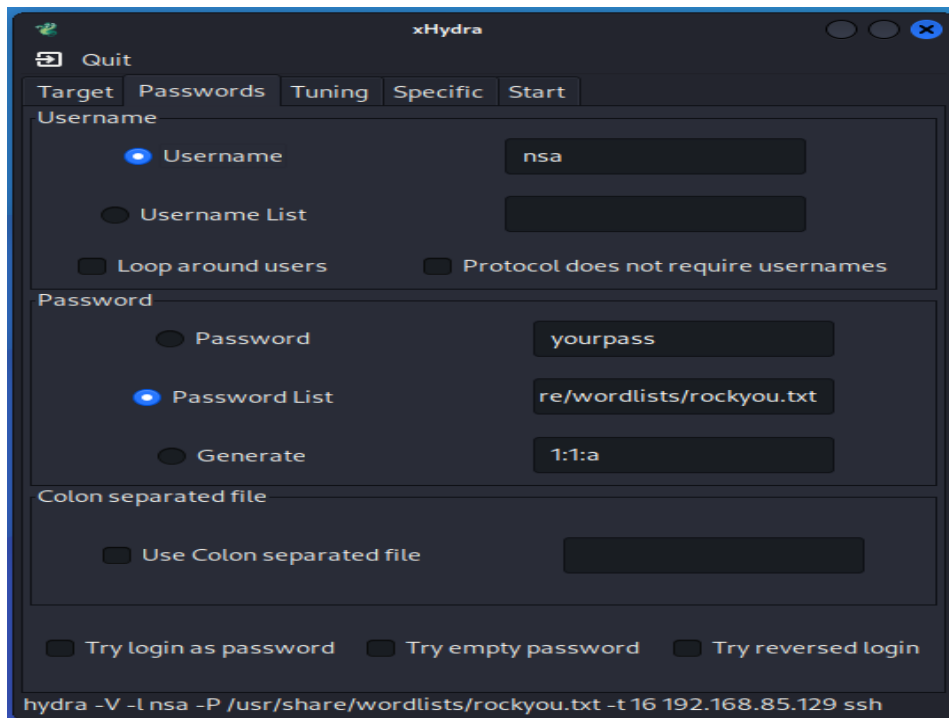
To perform a SSH Brute attack, a tool is need which is “xHydra”

First image it shows that it’s assigning the Ubuntu VM, also shows the SSH protocol being used.

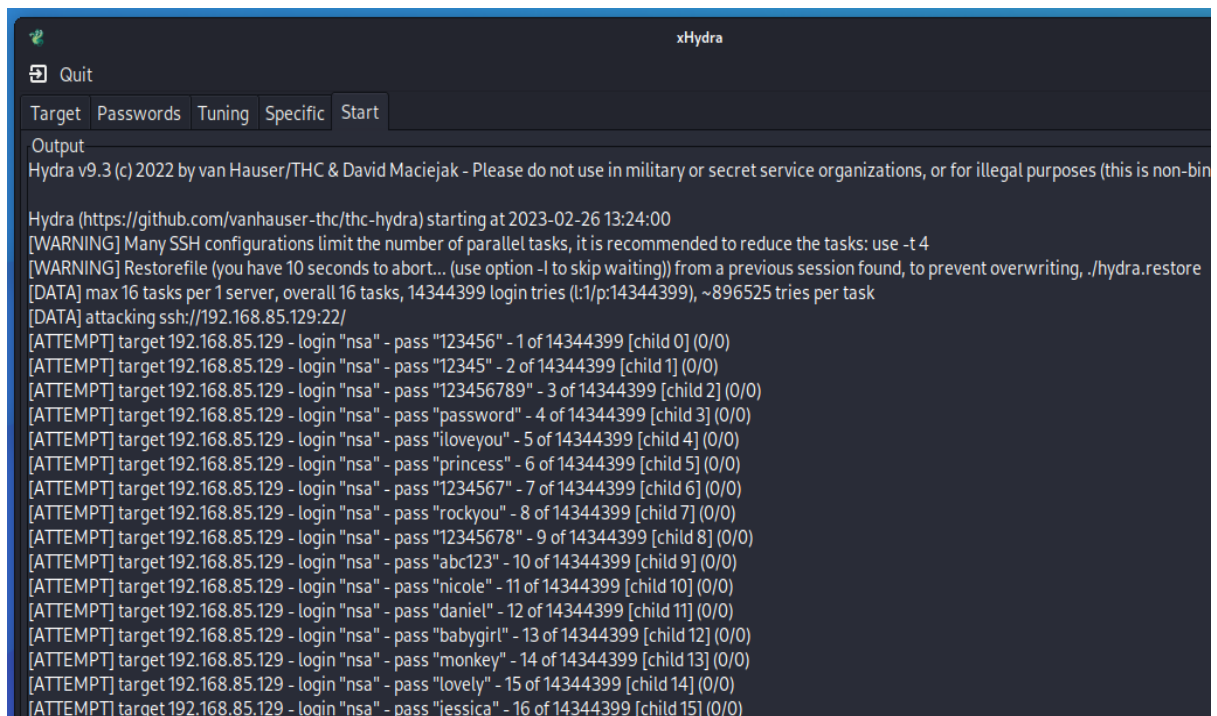


Second image shows that it will be using a pass wordlist to brute force the password

Assignment 1 – Open-Source Firewalls



Here it shows the multiple attempts being done to crack the password.



And here is an image of the Nmap scan displaying in Wireshark.

Assignment 1 – Open-Source Firewalls

9	5.584153	192.168.85.128	192.168.85.129	TCP	66 57234 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2962952593 TSecr=2314777614
10	5.584209	192.168.85.128	192.168.85.129	SSHv2	89 Client: Protocol (SSH-2.0-libssh_0.10.4)
11	5.584279	192.168.85.129	192.168.85.128	TCP	66 22 → 57234 [ACK] Seq=1 Ack=24 Win=65152 Len=0 TSval=2314777614 TSecr=2962952593
12	5.591295	192.168.85.129	192.168.85.128	SSHv2	107 Server: Protocol (SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.1)
13	5.591435	192.168.85.128	192.168.85.129	TCP	66 57234 → 22 [ACK] Seq=24 Ack=42 Win=64256 Len=0 TSval=2962952600 TSecr=2314777621
14	5.592854	192.168.85.128	192.168.85.129	SSHv2	882 Client: Key Exchange Init
15	5.593756	192.168.85.129	192.168.85.128	SSHv2	1146 Server: Key Exchange Init
16	5.594060	192.168.85.128	192.168.85.129	SSHv2	114 Client: Elliptic Curve Diffie-Hellman Key Exchange Init
17	5.598493	192.168.85.129	192.168.85.128	SSHv2	590 Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
18	5.598786	192.168.85.128	192.168.85.129	SSHv2	82 Client: New Keys
19	5.640665	192.168.85.129	192.168.85.128	TCP	66 22 → 57234 [ACK] Seq=1646 Ack=904 Win=64384 Len=0 TSval=2314777671 TSecr=2962952607
20	5.640847	192.168.85.128	192.168.85.129	SSHv2	110 Client:
21	5.640998	192.168.85.129	192.168.85.128	TCP	66 22 → 57234 [ACK] Seq=1646 Ack=948 Win=64384 Len=0 TSval=2314777671 TSecr=2962952649
22	5.641100	192.168.85.129	192.168.85.128	SSHv2	110 Server:
23	5.641336	192.168.85.128	192.168.85.129	SSHv2	126 Client:
24	5.648534	192.168.85.129	192.168.85.128	SSHv2	118 Server:
25	5.648815	192.168.85.128	192.168.85.129	SSHv2	118 Client:

4.5 Attacking with Nftables firewall configured.

Here is how the pining attack returns when the Nftables is configured

4.5.1 Ping (Nftables)

```
(nsa@kalilinux-vm)-[~]
$ ping -c 10 192.168.85.129
PING 192.168.85.129 (192.168.85.129) 56(84) bytes of data.

— 192.168.85.129 ping statistics —
10 packets transmitted, 0 received, 100% packet loss, time 9220ms
```

Nftables conf file

```
#Prevent ICMP and Unset Flag Packets
ip protocol icmp log prefix "ICMP Dropped: " drop
ip protocol tcp tcp flags & (fin|syn|rst|ack) == 0 log prefix "ICMP Flag Dropped:" drop
```

Wireshark result:

Assignment 1 – Open-Source Firewalls

20	5.290136	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request id=0x4a31, seq=2/512, ttl=64 (no response found!)
21	5.996960	192.168.85.1	224.2.2.2	UDP	72 64055 + 8995 Len=30
22	6.314409	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request id=0x4a31, seq=3/768, ttl=64 (no response found!)
23	6.997512	192.168.85.1	224.2.2.2	UDP	72 64055 + 8995 Len=30
24	7.337975	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request id=0x4a31, seq=4/1024, ttl=64 (no response found!)
25	7.998491	192.168.85.1	224.2.2.2	UDP	72 64055 + 8995 Len=30
26	8.362239	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request id=0x4a31, seq=5/1280, ttl=64 (no response found!)
27	8.997398	192.168.85.1	224.2.2.2	UDP	72 64055 + 8995 Len=30
28	9.386362	192.168.85.128	192.168.85.129	ICMP	98 Echo (ping) request id=0x4a31, seq=6/1536, ttl=64 (no response found!)

4.5.2 Nmap Scan

Here is the result after trying to perform an Nmap scan while Nftables is configured

```
(nsa@kalilinux-vm)-[~]
$ sudo nmap -sS -sV 192.168.85.133
[sudo] password for nsa:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-26 13:41 GMT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.72 seconds
zsh: segmentation fault  sudo nmap -sS -sV 192.168.85.133

(nsa@kalilinux-vm)-[~]
$ sudo nmap -sS -sV 192.168.85.133 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-26 13:42 GMT
Nmap done: 1 IP address (0 hosts up) scanned in 1.65 seconds
zsh: segmentation fault  sudo nmap -sS -sV 192.168.85.133 -Pn
```

Nftables conf file:

```
#Prevent ICMP and Unset Flag Packets
ip protocol icmp log prefix "ICMP Dropped: " drop
ip protocol tcp tcp flags & (fin|syn|rst|ack) == 0 log prefix "ICMP Flag Dropped:" drop
```

4.5.3 SSH Brute Force

Hydra when Nftables is configured

```
xHydra
Quit
Target Passwords Tuning Specific Start
Output
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, c
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-26 13:46:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -l to skip waiting)) from a previous session found, to p
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.85.129:22/
[ERROR] could not connect to ssh://192.168.85.129:22 - Connection refused
```

Result in Nftables:

```
#Prevent SSH Bruting
tcp dport 22 ct state new limit rate 3/minute burst 3 packets counter log prefix "SSH Dropped: " reject with tcp reset
```

Assignment 1 – Open-Source Firewalls

Wireshark result:

43	12.182333	192.168.85.128	192.168.85.129	TCP	74 47648 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2964560184 TSecr=0 WS=128
44	12.182550	192.168.85.129	192.168.85.128	TCP	60 22 → 47648 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

4.6 Attacking with IPFire enabled.

Shows that SSH is remained opened even when IPFire is enabled

4.6.1 SSH Brute Force

Brute Force before IPFire patches

Assignment 1 – Open-Source Firewalls

```
(nsa@kalilinux-vm)-[~]
$ ssh nsa@192.168.85.133
nsa@192.168.85.133's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-60-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.





nsa@ubuntu-vm:~$
```

IPFire allows you to open SSH but it doesn't allow you to perform a SSH Brute Force Attack.

```
(nsa@kalilinux-vm)-[~]
$ nmap 192.168.85.133 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-26 22:46 GMT
Nmap scan report for 192.168.85.133
Host is up (0.00076s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 19.57 seconds
```

Brute Force after IPFire patches

Firewall Rules						
#	Protocol:	Source	Log	Destination	Action	
1	TCP	RED	<input checked="" type="checkbox"/>	Firewall : 22 ->192.168.1.100: 22	<input checked="" type="checkbox"/>	   

Additional settings

Remark:

Rule position:

☒ Activate rule

☒ Log rule

☐ Use time constraints

☒ Limit concurrent connections per IP address

Max. concurrent connections:

☒ Rate-limit new connections

Number of connections: /

4.6.2 Ping

This rule drops all ICMP types.

Firewall Rules

Source

☐ Source address (MAC/IP address or network):

☒ Standard networks:

☐ Location:

☐ Firewall:

NAT

☐ Use Network Address Translation (NAT)

Destination

☐ Destination address (IP address or network):

☒ Standard networks:

☐ Location:

☐ Firewall:

Protocol

ICMP type:

☐ ACCEPT ☒ DROP ☐ REJECT

Additional settings

Remark:

Rule position:

☒ Activate rule

☒ Log rule

☐ Use time constraints

☐ Limit concurrent connections per IP address

☐ Rate-limit new connections

Assignment 1 – Open-Source Firewalls

Firewall Rules ?

Source

☐ Source address (MAC/IP address or network):

☐ Firewall

All ▼

☒ Standard networks:

Any ▼

☐ Location

A1 - Anonymous Proxy ▼

NAT

☐ Use Network Address Translation (NAT)

Destination

☐ Destination address (IP address or network):

☐ Firewall

All ▼

☒ Standard networks:

RED ▼

☐ Location

A1 - Anonymous Proxy ▼

Protocol

ICMP ▼

ICMP type:

All ICMP types ▼

☐ ACCEPT

☒ DROP

☐ REJECT

Additional settings

Remark:

Rule position:

1 ▼

☒ Activate rule

☐ Log rule

☐ Use time constraints

☐ Limit concurrent connections per IP address

☐ Rate-limit new connections

Update

Back

These two images shows that the rules block ICMP requests

B00137882 B00134937

2

5. Comparison and Conclusion

Hence, even though Nftables and IPFire are both concerned with network security, their functions are distinct. Nftables is a framework for packet filtering, whereas IPFire is a full-featured firewall solution. Both tools or one of them might be helpful for network security depending on the precise requirements and goals.

Overall Nftables is more for local based network or home network whereas IPFire will be considered a commercial network solution on a larger scale.

6. References

National Institute of Standards and Technology. (2021). Nmap Reference Guide. Available at: <https://nmap.org/book/man.html#man-description> (Accessed: February 5, 2023)

Bernhard Bitsch (2023) What is IPFire? IPFire Wiki. Available at: <https://wiki.ipfire.org/what-is-ipfire> (Accessed: February 3, 2023)

ExtraHop Networks, Inc. (2020). Brute-Force Attacks: What You Need to Know. Available at: <https://www.extrahop.com/resources/attacks/brute-force/#:~:text=SSH%20is%20used%20for%20remote,until%20they%20find%20a%20match.> (Accessed: February 2, 2023)

Pablo (2016) Simple rule management. nftables wiki. Available at: https://wiki.nftables.org/wiki-nftables/index.php/Simple_rule_management (Accessed: February 5, 2023)

Cloudflare, Inc. (2020). What is a SYN Flood DDoS Attack? Available at: <https://www.cloudflare.com/en-gb/learning/ddos/syn-flood-ddos-attack/> (Accessed: February 1, 2023)

Pablo (2016). What is nftables? nftables wiki. Available at: https://wiki.nftables.org/wiki-nftables/index.php/What_is_nftables%3F (Accessed: February 5, 2023)

ArchWiki contributors. (2022). nftables. ArchWiki. Available at: <https://wiki.archlinux.org/title/nftables> (Accessed: January 28, 2023)