

How do the configuration parameters of Nmap-based firewall bypassing port scans affect the vulnerability of host-based firewalls?

Rajiv Govind

Table of Contents

1. Introduction.....	3
1.1 Introduction to port scanning and nmap	4
2. Types of bypassing port scans	6
2.1 IP Fragmentation	6
2.2 Source Port Manipulation	6
2.3 Decoy Scans	7
2.4 Time-to-live attack	8
3. Data Collection and Processing.....	9
3.1 Raw Data Collection	9
3.2 Methodology and Procedure	9
3.2.1 Setting up the Virtual Lab Environment	9
3.2.2 Creating the network of virtual machines.....	Error! Bookmark not defined.
3.2.3 Configuring the firewall on the target host.....	11
3.2.4 Starting the services on the system required for testing	12
3.2.5 The Decoy Scan.....	12
3.2.6 IP Fragmentation.....	14
3.2.7 Source Port Manipulation	15
3.2.8 Time-to-live attack	16
4. Analysis	17
4.1 Raw Data Tables:.....	17
4.2 Processed Data	19
4.3 Data Presentation	20
4.4 Limitations of the Investigation	24
5. Conclusion and Extension of the investigation	25
5.1 Conclusion	25
5.2 Extension of the Investigation	26
Appendix.....	26
Bibliography.....	29

1. Introduction

In today's networked world, the dependence on firewalls has greatly increased. The types of firewalls available for public use today include host-based firewalls, network firewalls and application-level firewalls. Network firewalls ensure no unauthorized access to a private network, whereas application-level firewalls govern traffic to and from an application or a service. A host-based firewall is installed on each individual server or device, and it acts as an intermediary for incoming and outgoing traffic. Custom firewall rules can be used for a host-based firewall, and this type of firewall provides protection for a device irrespective of its location and which network it is connected to. Host-based firewalls have been chosen for this investigation because they are the default type of firewall present in most operating systems popularly used today.¹

However, as with anything, there are flaws with the existing technology as well, which can be bypassed as firewall configuration is an important step to ensure security. Most computer users and network administrators do not correctly configure their firewalls, leading to a variety of vulnerabilities that can be exploited by hackers.² Once the firewall is bypassed to reveal open ports, these open ports can then be used by hackers to directly gain access to a machine. To prevent this from occurring, ethical hackers have discovered a variety of methods to test firewalls, using vulnerability analysis tools. One such tool is nmap, a vulnerability analysis tool used by ethical hackers to conduct port scans that discover open ports on systems and reveal services that are running. Within nmap, there are a variety of methods and parameters present to customize the type of port scan, which can allow these scans to bypass firewalls if the parameters are configured correctly. Based on the parameters used, these are different "nmap-based firewall evasion port scans". To choose

¹ Richardson, Stephen. "Host Based Firewalls - Firewall Security - Cisco Certified Expert." *Cisco Certified Expert*, 7 Jan. 2022, <https://www.ccexpert.us/firewall-security/hostbased-firewalls.html>.

² "The Dangers of Firewall Misconfigurations | Guardicore." *Guardicore*, 16 Nov. 2020, <https://www.guardicore.com/blog/the-dangers-of-firewall-misconfigurations-and-how-to-avoid-them/>.

these methods for this research, nmap's list of the most potent firewall evasion methods was referred to, and 4 practical methods were chosen based on available technology.³

For a bypassing port scan to be deemed "successful", it would have to display all running services and ports on the machine that it scans, while the firewall is active. This success of showing the services and ports determines how vulnerable the firewall is, which is why it is referred to as the "vulnerability rate" throughout this investigation. After examining these bypassing port scan, it was evident that due to the flexibility in changing parameters that nmap provides, the vulnerability rate of the firewall bypassing scan would vary significantly with a change in its parameters. This means that even minor adjustments in the parameters for these firewall evasion scans could greatly increase or decrease the vulnerability rate of the firewall. Knowledge of the parameters that lead to the highest vulnerability rate, or "optimal parameters" could benefit computer users as they can implement specific measures to prevent exploitation of this vulnerability. This led me to formulate my research question as "How do the configuration parameters of nmap-based firewall bypassing port scans affect the vulnerability of host-based firewalls?"

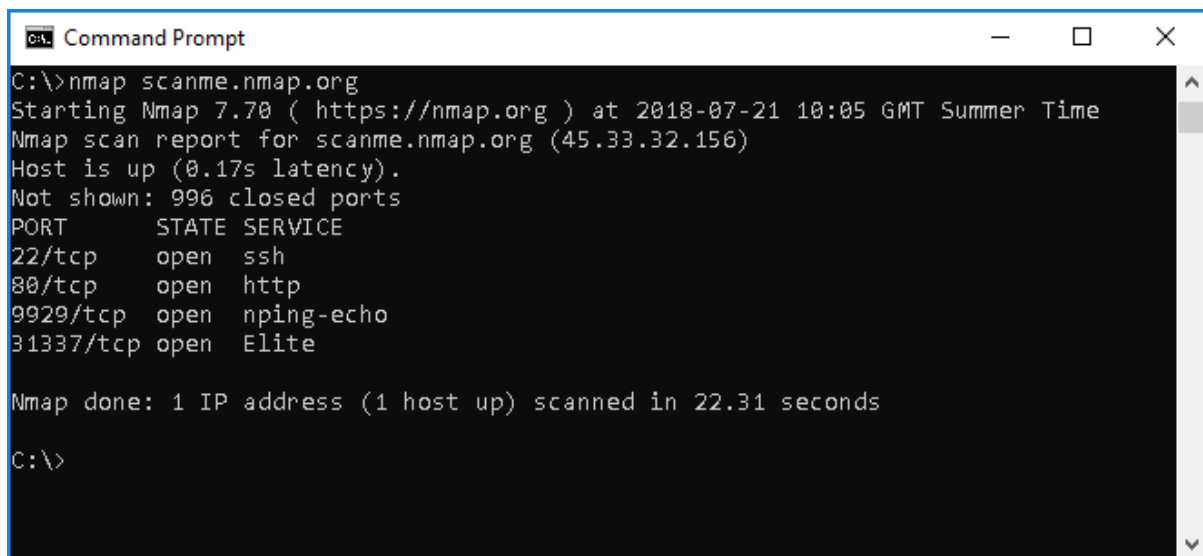
1.1 Introduction to port scanning and nmap

The term port scanning refers to a computer sending packets over a network to a target machine to determine the state of ports and services on the target machine. Ports can take 3 different forms: open, closed or filtered. "Open" refers to the state in which there is service listening on the port and the machine accepting connections from other devices. "Closed" refers to the state in which no services are listening on the port. When a port is filtered, it indicates that there is a firewall that is blocking communication to the port. Filtered ports can be either open or closed, and the goal of bypassing a firewall is to reveal whether the filtered ports are either open or closed.

As mentioned in the introduction, the primary vulnerability analysis software being used to test the firewall evasion port scans is nmap. Nmap is a command-line based tool which allows users to enter commands based on the type of port scan they wish to conduct.

³ "Bypassing Firewall Rules | Nmap Network Scanning." *Nmap: The Network Mapper - Free Security Scanner*, <https://nmap.org/book/firewall-subversion>.

The below screenshot shows a sample port scan conducted using nmap.

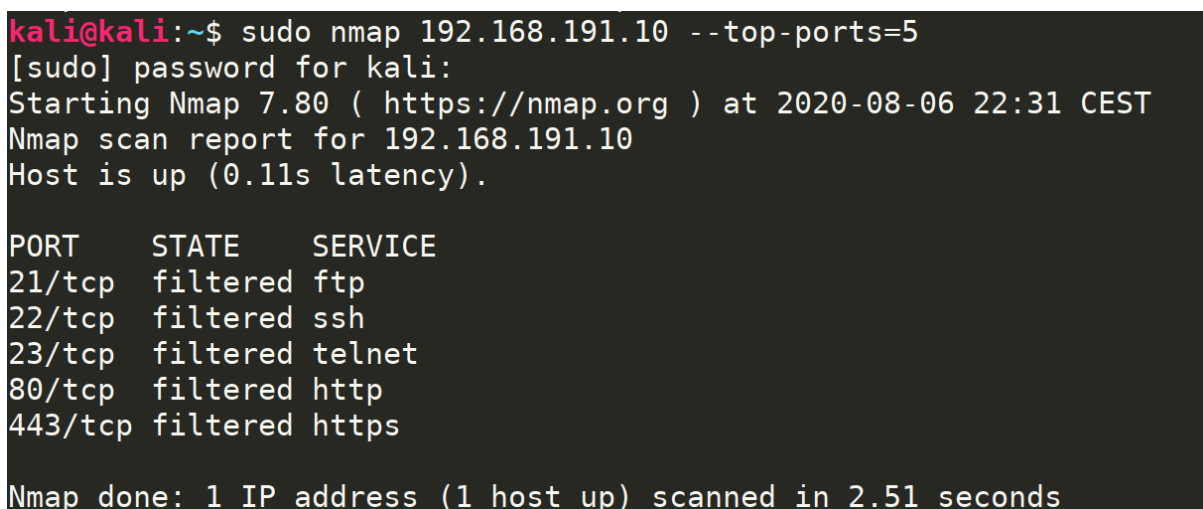
A screenshot of a Windows Command Prompt window titled "Command Prompt". The window has a black background with white text. The text shows the execution of the nmap command on the host scanme.nmap.org. The output indicates that the host is up and lists several open ports with their corresponding services. The scan was completed in 22.31 seconds.

```
C:\>nmap scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-21 10:05 GMT Summer Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.17s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 22.31 seconds
C:\>
```

Figure 1: Sample Port Scan

As seen in the above scan, nmap displays the state of the ports on a machine along with the services listening on these ports. However, this scan was conducted on a system with no firewall. When a firewall is implemented, the results of this scan change. The below figure demonstrates this.

A screenshot of a Kali Linux terminal window. The text shows the execution of the nmap command on the IP address 192.168.191.10, specifically targeting the top 5 ports. The output shows that all five ports (21, 22, 23, 80, 443) are in a 'filtered' state, indicating that the packets are being blocked by a firewall. The scan was completed in 2.51 seconds.

```
kali@kali:~$ sudo nmap 192.168.191.10 --top-ports=5
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-06 22:31 CEST
Nmap scan report for 192.168.191.10
Host is up (0.11s latency).

PORT      STATE SERVICE
21/tcp    filtered ftp
22/tcp    filtered ssh
23/tcp    filtered telnet
80/tcp    filtered http
443/tcp   filtered https

Nmap done: 1 IP address (1 host up) scanned in 2.51 seconds
```

Figure 2: Filtered ports on nmap

As seen, the ports scanned all appear to be filtered, which means that the packets are blocked by the host-based firewall. For Nmap firewall evasion port scans to be successful, they should be able to bypass this filter in order to identify which ports are open.

2. Types of bypassing port scans

2.1 IP Fragmentation

IP Fragmentation attacks use the fact that network links have a maximum size of messages that can be transmitted through them, known as the maximum transmission unit. If the service data unit of a packet along with the metadata, which contains data pertaining to the structure of network protocols and packets, exceeds the maximum transmission unit, the service data unit needs to be fragmented. This process is used as an attack vector in IP fragmentation attacks. Since the packets exceed the maximum transmission unit, they are impossible to reassemble, occupying server resources and resulting in a denial-of-service attack. The denial-of-service results in the firewall unable to filter additional packets that are sent to the machine, which results in it bypassed.⁴

2.2 Source Port Manipulation

In this attack, the attacker sends packets from a source port of his choice, to exploit misconfigurations in the host's operating system. The source port refers to the port from which these packets originate, which deceives the firewall into thinking the traffic is legitimate. This attack works well because using services such as File Transfer Protocol (FTP), leads users to oftentimes blindly allow the traffic that originates from port 21, the port used for FTP. In addition, since DNS replies originate from port 53, many network administrators allow all incoming traffic from this port.

In the past, Microsoft Windows 2000 and XP came with a vulnerability that allowed all TCP and UDP traffic from port 88, which is used for the Kerberos service, a computer-network authentication protocol.

Below is a screenshot from JJ Gray's demonstration of the exploitation of port 88.

⁴ Namburu, Anupama, and Soubhagya Sankar Barpanda. *Recent Advances in Computer Based Systems, Processes and Applications*. CRC Press, 2020.

Figure 3: JJ Gray's demonstration of Kerberos exploitation

```
# nmap -sS -v -v -Pn 172.25.0.14
Starting Nmap ( http://nmap.org )
Nmap scan report for 172.25.0.14
Not shown: 1658 filtered ports
PORT      STATE SERVICE
88/tcp    closed kerberos-sec

Nmap done: 1 IP address (1 host up) scanned in 7.02 seconds

# nmap -sS -v -v -Pn -g 88 172.25.0.14
Starting Nmap ( http://nmap.org )
Nmap scan report for 172.25.0.14
Not shown: 1653 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
1027/tcp  open  IIS
1433/tcp  open  ms-sql-s

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

As seen, when an nmap scan is conducted with the source port given as port 88, as shown by the parameter “-g 88”, the firewall is bypassed and all the open ports on the system are shown, along with the services running on each port. This is because the scan is re-routed through an approved service on the system, and the firewall allows the traffic under the assumption that it is legitimate.⁵

2.3 Decoy Scans

Decoy scans make it appear to the machine being scanned (machine with an active host-based firewall) that other hosts are scanning the network as well, which would result in the machine not knowing which IP was scanning it searching for services and which addresses were decoys. Ordinarily, computers in networks constantly receive packets from multiple machines. This means that when the target host appears to be getting scanned by multiple machines at once through the decoy scan, the Nmap port scan may go undetected.

⁵ “Bypassing Firewall Rules | Nmap Network Scanning.” *Nmap: The Network Mapper - Free Security Scanner*, <https://nmap.org/book/firewall-subversion.html>.

2.4 Time-to-live attack

Time-to-live or TTL refers to the amount of time a packet is supposed to exist within a network before it is discarded by a router.

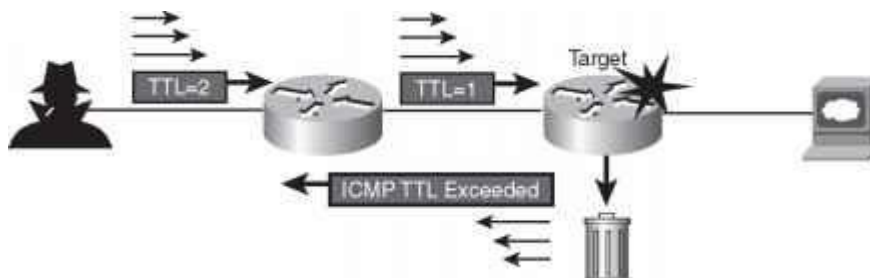
This attack works on the principle that when the time-to-live (TTL) of a packet reaches 0 on the routing platform, a TTL Exceeded message is required to be sent back to the sender.

Constantly having to send these TTL Exceeded messages back to the sender would result in a denial-of-service attack on the firewall, leaving it unable to filter additional packets. These additional packets can then be used to scan the system for open ports and services, bypassing the firewall.

Using this principle, the TTL value for packets to be sent to the host must be calculated. This can be done by sending a packet with a TTL value of 1 minus the number of hops to the system. When this is done, the packet ends up having a TTL value of exactly 1 once it reaches the network switch, meaning that it would have a TTL value of 0 when it reaches the host, resulting in the generation of a TTL Exceeded message that is then sent back to the sender. If enough packets are sent, the host machine ends up using a significant number of resources to process these TTL expired packets, making it unstable.

The below diagram represents the time-to-live attack:

Figure 4: Time-to-live attacks



3. Data Collection and Processing

3.1 Raw Data Collection

To collect raw data for the investigation, virtual machines running host-based firewalls with different operating systems were used. A variety of operating systems were selected for this investigation, some operating systems were catered to casual computer users (such as Windows 7 Home Basic, Windows 10 Home, Windows 10 Professional, and Ubuntu Linux), and the others were catered to system administrators (such as Windows Server 2012 and Windows Server 2016). Since this variety of operating systems had been used, this research would benefit both casual users to high-level network administrators, as the most potent threats posed by each bypassing port scan have been identified in the investigation. The appendix also contains custom configurations for users to put in place to completely counter the potent firewall evasion methods shown in this investigation, which mitigates any ethical issues associated with this research.

The system used to conduct the attacks was running Kali Linux (version 2021.1). The primary tool used in testing was Nmap and 5 trials were taken for each variation of the parameter, for each operating system. 5 trials are sufficient because it results in a total of 30 attempts for each parameter, which provides a significant amount of data for each parameter. For each trial for the operating system, a different virtual machine file running the same operating system was used, to ensure no bias between results from each trial. Throughout the research, the term “successes” refers to whenever the firewall is successfully bypassed. This occurs when all services and open ports that are on the system are displayed through the Nmap port scan on the target machine when the firewall is active.

3.2 Methodology and Procedure

3.2.1 Setting up the Virtual Lab Environment

Before conducting the **firewall bypassing** port scans on each system, the lab environment had to be set up. In order to facilitate this, VMWare 16.2 was used with the NAT network adapter for each virtual machine, creating an isolated network with the desired systems.

This also ensures the Maximum Transmission Unit for the network is constant, and the number of hops between systems do not fluctuate. Below is a screenshot of how this was done.

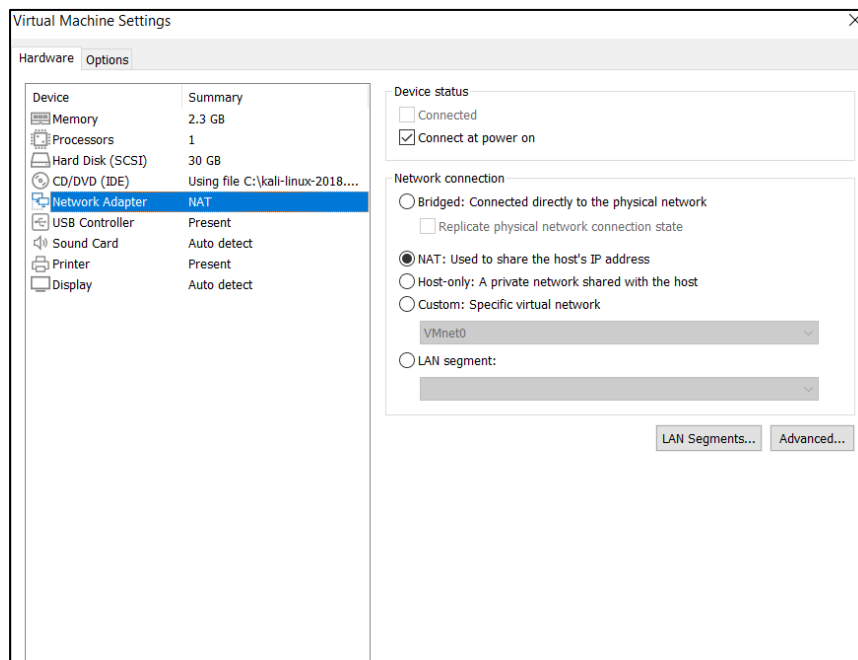


Figure 5: Network adapter settings for virtual machines

In the lab setup there were 2 main parties. There was the attack machine, which was standard for all trials, and was a virtual machine running Kali Linux 2020.1. Kali Linux is a popular operating system used for system exploitation and vulnerability analysis. On the other end, target virtual machines were set up, with different operating systems for each system, with the default host-based firewalls present.

Below is a screenshot of the Kali Linux user interface.



Figure 6: Kali Linux

3.2.3 Configuring the firewall on the target host

Once the network was set up, the firewall had to be configured on the target host. Below is a screenshot that shows how the firewall was set up on the Windows Defender Firewall

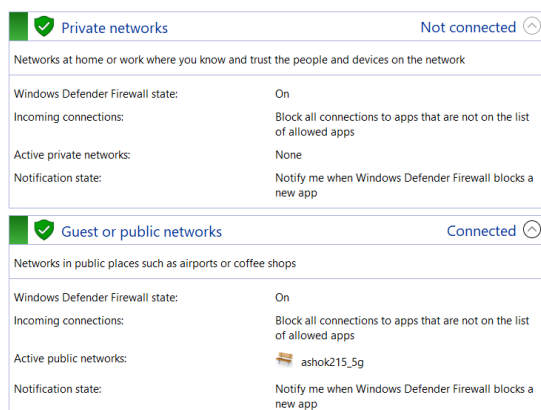


Figure 9: Configuring the firewall

As seen, the firewall does not accept any connections that are not from the list of allowed apps and services, which means that when scanning the system, **all the ports would appear to be filtered**. A port state of “filtered” means that the port can either be open or closed, but it is guarded by a firewall. The display of a filtered state means that the scan was not successful in bypassing the firewall.

3.2.4 Starting the services on the system required for testing

For each system, the XAMPP Control Panel was used to enable particular services that were used for testing.

Below is a screenshot of what the control panel looks like:

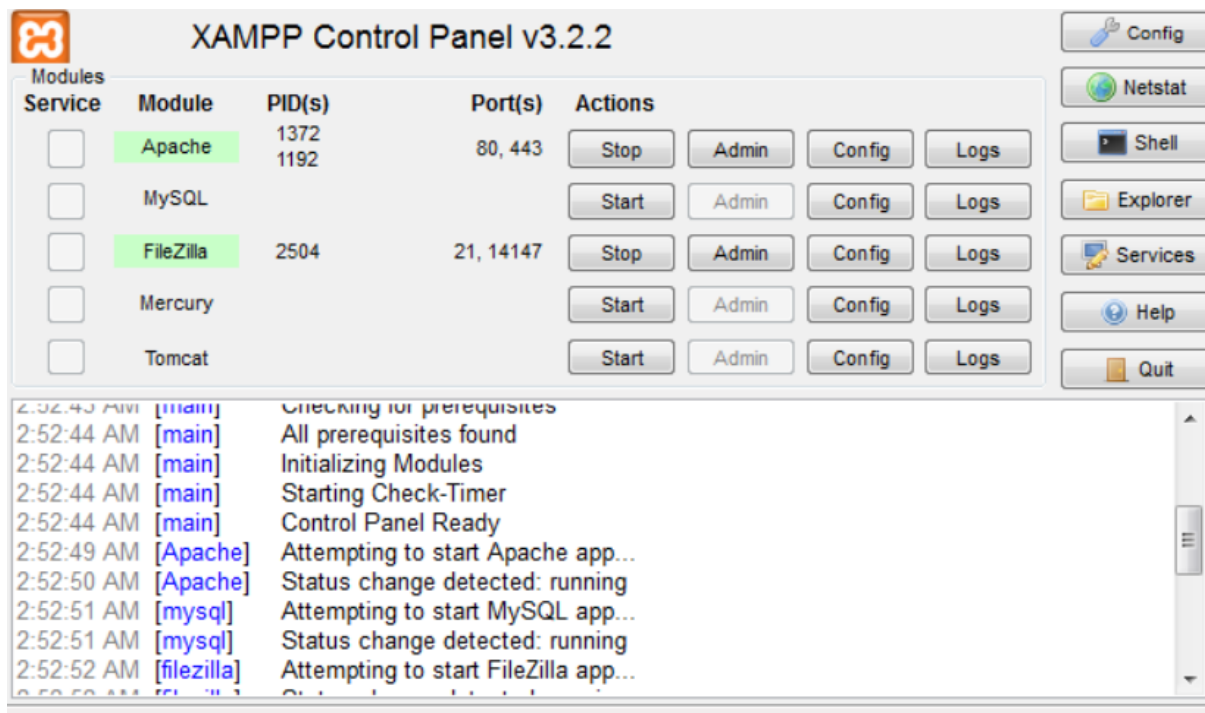


Figure 10: Configuring XAMPP

For the purpose of this experiment, two common services were enabled on every machine that was tested: Apache (which uses http and https), and FileZilla, a common FTP service that is used by many around the world. All the other default services were kept running on every system, to ensure that the results were without bias.

3.2.5 The Decoy Scan

Using nmap, the parameter for the Decoy Scan that was varied was the number of decoys, or total number of machines that would be scanning the target machine.

In order to create a decoy scan, the nmap command had to be used with the -D parameter along with "RND" which specifies the number of decoys that are going to be deployed on the target host. Finally, the IP address of the machine had to be specified, 192.168.177.129.

Below is a scan that was used as a control scan on the machine, to ensure that the firewall was operating correctly. As seen, all 1000 scanned ports on the machine appear to be “filtered”, showing that the firewall is blocking traffic correctly.

```
root@kali:~# nmap 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-07-09 03:14 IST
Nmap scan report for 192.168.177.129
Host is up (0.00037s latency).
All 1000 scanned ports on 192.168.177.129 are filtered
MAC Address: 00:0C:29:BF:EA:4C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 21.61 seconds
```

Figure 11: Running a default nmap scan on the system

A successful decoy scan evades the firewall and shows the ports on the system that have services running. As seen in the figure below, when setting the decoy parameter to 3 decoys, there is an ftp server running on port 21, along with http and https running on 80 and 443. These were the services that were intentionally started through the XAMPP Control Panel, for the purpose of this test. This shows that the firewall was successfully bypassed as instead of these services appearing to be filtered, they are now shown to be actually open.

```
root@kali:~# nmap -D RND:3 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 20:02 IST
Nmap scan report for 192.168.177.129
Host is up (0.00066s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:BF:EA:4C (VMware)

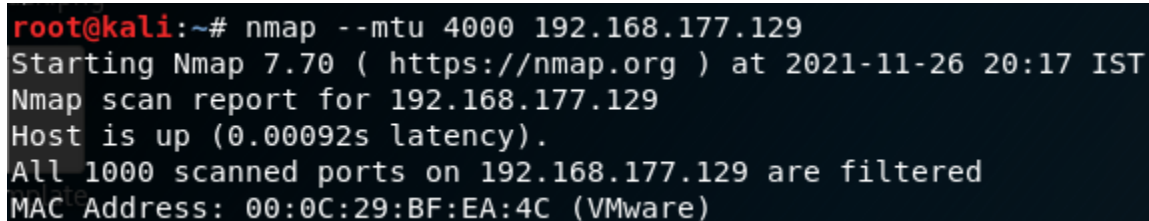
Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
root@kali:~#
```

Figure 13: Successful NMAP Decoy Scan with 3 decoys

3.2.6 IP Fragmentation

In order to conduct an IP fragmentation scan, the “—mtu” parameter was used to tweak the size of the fragment being sent to the target machine. This is the parameter that will be tested in the investigation. In nmap, the “—mtu” pattern only accepts the fragment size in the number of bits. However, for the purpose of the simplicity of this investigation, the fragment sizes will be described in terms of bytes.

For this network, the maximum transmission unit (MTU) was found out to be 1500 bytes, since the virtual machines used a NAT configuration (3.2.1). In theory, a fragmentation attack would be successful if the fragment size is greater than this value. The first scan shows the results of the scan when the fragment size was set to 500 bytes (4000 bits). As seen, the scan does not bypass the firewall, as it is within the MTU of the network.



```
root@kali:~# nmap --mtu 4000 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 20:17 IST
Nmap scan report for 192.168.177.129
Host is up (0.00092s latency).
All 1000 scanned ports on 192.168.177.129 are filtered
MAC Address: 00:0C:29:BF:EA:4C (VMware)
```

Figure 14: Running an IP fragmentation attack with a fragment size of 500 bytes

However, when the fragment size is increased past the MTU to 2500 bytes (20000 bits), the firewall is bypassed, showing all the services running on the target system.

```
nmap done: 1 IP address (1 host up) scanned in 22.20 seconds
root@kali:~# nmap --mtu 20000 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 20:18 IST
Nmap scan report for 192.168.177.129
Host is up (0.00097s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49157/tcp  open  unknown
MAC Address: 00:0C:29:BF:EA:4C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 2.13 seconds
```

Figure 15: Successful IP fragmentation scan with a fragment size of 2500 bytes

As seen in the scan, the same services and open ports exist on this system, which means that the firewall has been successfully bypassed.

3.2.7 Source Port Manipulation

In order to use the source port manipulation technique, one has to know what services are running on the target machine. For this test, each machine had the service running based on which parameter was being tested. For example, if source port manipulation using FTP was being assessed, the FTP service would be activated on all machines. This allows the manipulation of port 21 to be tested, using the “source-port” parameter for the nmap scan command. This “source-port” parameter is the parameter that is varied for this port scan.

When conducting the source port manipulation using port 21, the open ports as well as the services running on these ports are displayed. This shows that the firewall is bypassed as the ports are no longer filtered. The below screenshot shows an example of a “successful” source port manipulation attack.


```

$ sudo nmap -sS --source-port 21 192.168.1.93
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-09 09:09 UTC
Nmap scan report for 192.168.1.93
Host is up (0.0010s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
443/tcp   open  https
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:BF:EA:4C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.19 seconds

```

Figure 17: Running a source port manipulation attack on the new target machine

3.2.8 Time-to-live (TTL) attack

In order to conduct the TTL attack, the system had to first be pinged, in order to determine the number of hops, as well as the TTL for the packets. Since the value for the TTL was 64ms, the experiment was conducted by varying TTL values between 20 and 200ms. Theoretically, for the TTL attack to be successful, the TTL would have to result in a “TTL value exceeded” on the receiving system’s end. For this to happen, the TTL value would have to be higher than the TTL value from the ping scan, but not too much higher as this would result in the packets being completely dropped.

The below scan shows the results of a TTL scan with a 20ms TTL. As seen, the firewall is not bypassed as the ports appear filtered.

```

root@kali:~# nmap --ttl 20 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 20:26 IST
Nmap scan report for 192.168.177.129
Host is up (0.0013s latency).
All 1000 scanned ports on 192.168.177.129 are filtered
MAC Address: 00:0C:29:BF:EA:4C (VMware)

```

Figure 18: Running a TTL scan with a TTL of 20ms

However, when the TTL is increased to 120ms, the firewall is successfully bypassed, and the services running are displayed.


```

Nmap done: 1 IP address (1 host up) scanned in 22.51 seconds
root@kali:~# nmap --ttl 120 192.168.177.129
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 20:27 IST
Nmap scan report for 192.168.177.129
Host is up (0.00066s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:BF:EA:4C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 1.89 seconds

```

Figure 19: Running a TTL scan with a TTL of 120ms

4. Analysis

4.1 Raw Data Tables:

Table 1: Raw Data for Time-to-live Attack

TTL Value (ms)	Number of Successes out of 5 attempts					
	Windows 10 Home	Windows 10 Pro	Server 2016	Windows 7 Home Basic	Ubuntu Linux	Server 2012
20	0	0	0	0	0	0
40	0	0	0	0	0	0
60	0	0	0	0	2	2
80	2	1	2	3	3	2
100	1	1	2	4	3	3
120	3	2	3	4	4	3
140	3	3	2	5	3	4
160	2	1	2	3	3	3
180	0	0	0	2	1	1
200	0	0	0	1	0	1

Source Port	Number of Successes out of 5 attempts
-------------	---------------------------------------

	Windows 10 Home	Windows 10 Pro	Server 2016	Windows 7 Home Basic	Ubuntu Linux	Server 2012
21 (ftp)	2	2	3	3	3	4
80 (http)	3	2	2	4	4	3
88 (kerberos)	1	0	0	3	3	2
443 (https)	2	2	2	3	2	2

Table 2: Raw Data for Source Port Manipulation Attack

Table 3: Raw Data for Decoy Scan Attack

Number of Decoys	Number of Successes out of 5 attempts					
	Windows 10 Home	Windows 10 Pro	Server 2016	Windows 7 Home Basic	Ubuntu Linux	Server 2012
1	0	0	0	2	2	1
2	1	1	1	2	2	1
3	2	1	1	3	3	2
4	0	0	0	1	0	0
5	0	0	0	0	0	0

Table 4: Raw Data for IP Fragmentation Attack

IP fragment size (bytes)	Number of Successes out of 5 attempts					
	Windows 10 Home	Windows 10 Pro	Server 2016	Windows 7 Home Basic	Ubuntu Linux	Server 2012
500	0	0	0	0	0	0
1000	0	0	0	0	0	0
1500	2	2	1	2	1	2
2000	3	2	2	3	3	3
2500	3	3	2	4	4	3
3000	1	1	0	3	3	0

The number of successes for each attack were displayed, out of a total 5 attempts for each operating system, and each parameter value. This meant that there were a total of 30 attempts for each parameter value. The attacks were deemed to be successful in firewall evasion if they displayed the services that were turned on as “open” after the scan (ftp,http,https).

4.2 Processed Data

For the processed data, the number of successes was totalled for each parameter value, and the “vulnerability rate” was calculated by expressing the total number of successes as a percentage of 30, which represents the total number of scans for each parameter value. The vulnerability rate is essentially a metric that is a percentage value of how often a firewall is bypassed, with that particular parameter configuration.

Table 5: Processed Data for Source Port Manipulation Attack

Source Port Number	Vulnerability Rate /%
21 (ftp)	56.7
80 (http)	60.0
88 (kerberos)	30.0
443 (https)	43.3

Table 6: Processed Data for Time-to-live Attack

TTL Value /ms	Vulnerability Rate /%
20	0.0
40	0.0
60	13.3
80	43.3
100	46.7
120	63.3
140	66.7
160	46.7
180	13.3
200	6.7

Table 7: Processed Data for Decoy Scan Attack

Number of Decoys	Vulnerability Rate/%
1	16.7
2	26.7
3	40.0
4	3.3
5	0.0

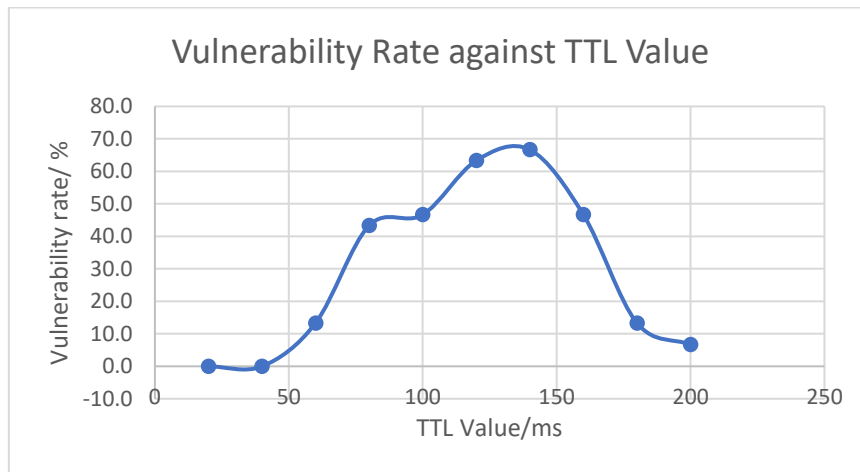
Table 8: Processed Data for IP Fragmentation Attack

IP fragment size /bytes	Vulnerability Rate/%
-------------------------	----------------------

500	0.0
1000	0.0
1500	33.3
2000	53.3
2500	63.3
3000	26.7

4.3 Data Presentation

Graph 1: Vulnerability rate for TTL Attack Configuration Parameters

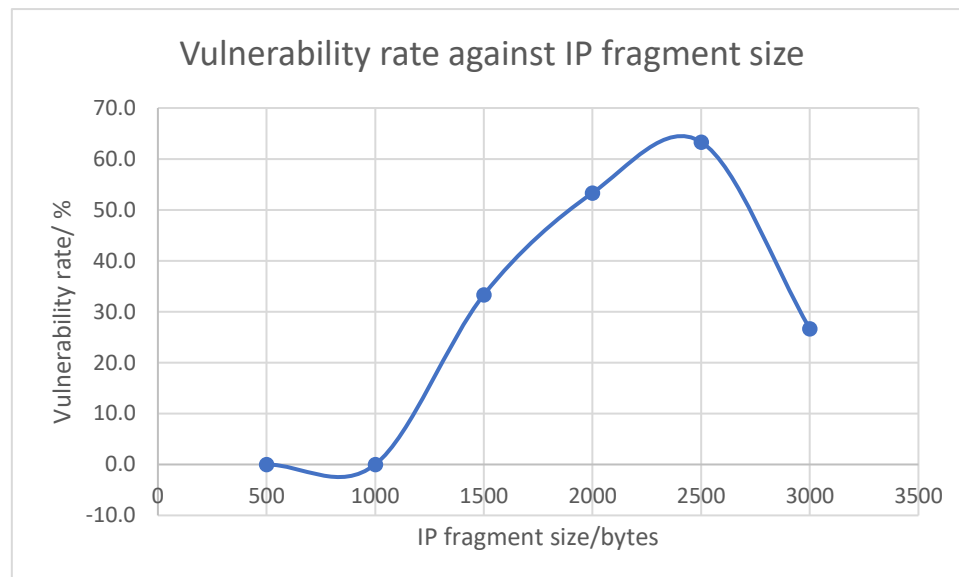


From the graph, the TTL value with the highest vulnerability rate is 140ms, with a vulnerability rate of 66.7% across all 6 operating systems. This is because this TTL value is significantly higher than the standard TTL value for the network, which was determined to be 64ms. This means that in order for the packets to have a TTL value of exactly 0 on the receiving end, the TTL value should be between 120ms and 140ms, as these are the values that produced the highest vulnerability rate of 63.3% and 66.7%. The reason for this vulnerability rate was because due to the TTL value that was significantly higher than the standard TTL for the network, all packets resulted in a perfect Denial of Service attack on the receiving end, allowing information about the operating system's services to be accessed without the firewall's intervention. Moreover, as seen in the graph, there was a small vulnerability rate for values even below the optimum TTL value. This is because with certain open-source operating systems such as Ubuntu Linux and Server 2012, the default host-based firewall improperly deals with packets that exceed the accepted TTL value, even if it does not happen every time. For values below the accepted TTL value, the vulnerability rate stays constant at 0%, because the firewall cannot be bypassed with standard port scans. Moreover, once the TTL value is increased past the maximum, the vulnerability rate starts to

fall. This is because for TTL values that are too large, the packet will end up being dropped by the firewall completely, leading to no scan and no information about the host system.

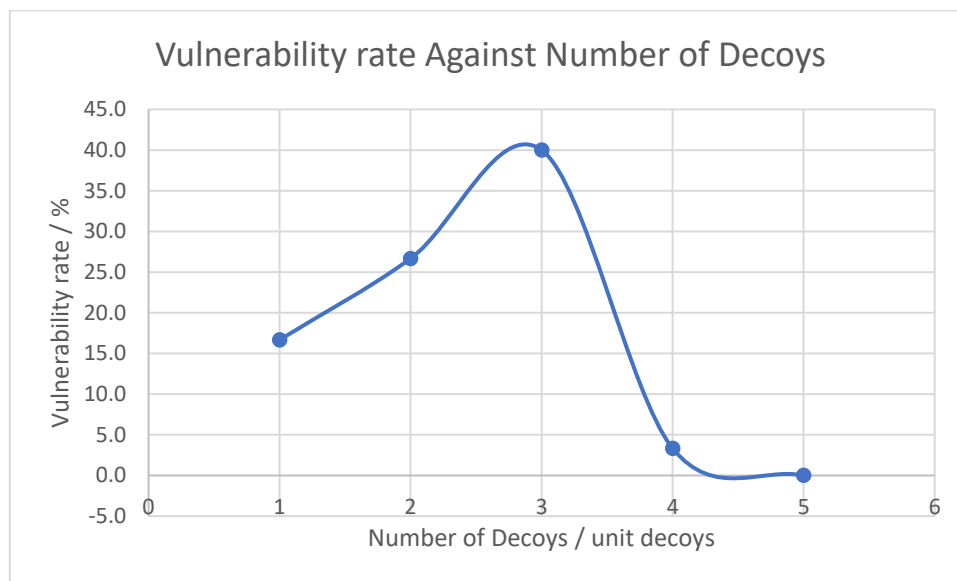
Graph 2: Vulnerability rate for IP Fragmentation Attack Configuration

Parameters



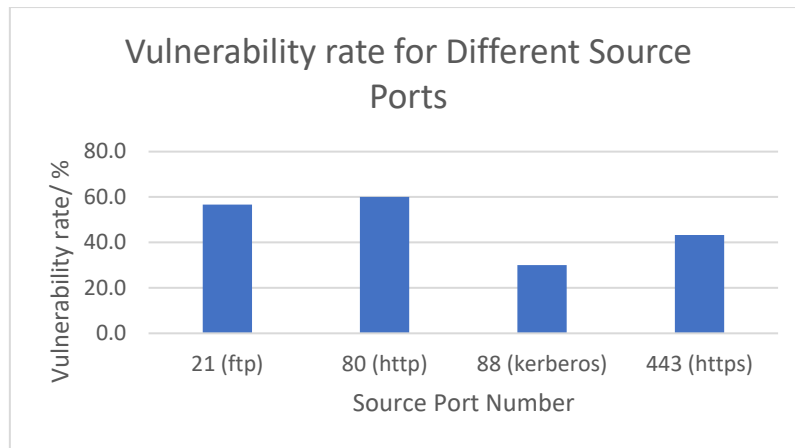
From the graph, the vulnerability rate is at 0 for 500 and 1000 bytes, as these are significantly below the Maximum Transmission Unit for the network, which means that there is no denial-of-service attack on the firewall, and the firewall is not bypassed. However, as the Maximum Transmission Unit is increased closer to 1500, which was the standard value for the network used in this investigation, the vulnerability rate begins to increase. Eventually, the vulnerability rate peaks at a value of 63.3% at 2500 bytes. This peak occurs due to the fact that at this point, most operating systems are unable to resolve the fragment because it is higher than the Maximum Transmission Unit, and lower than the point at which the packets are completely dropped. However, for fragment values greater than this peak, most operating systems dropped the packets completely, leading to a low average vulnerability rate. On the contrary, there were still certain machines that fell victim to this attack, but at a significantly lower rate. As seen with the previous attacks, it was **the older operating systems** and Linux that still fell victim to the IP fragmentation attack even after the fragment size was increased past the optimum.

Graph 3: Vulnerability rate for Decoy Scan Attack Configuration Parameters



For the decoy scan, there was also an optimal number of decoys, which lied at 3 decoys, for which the vulnerability rate on average was 40%. In addition, it is worthy to note that the decoy scan was, on average, far less successful than all other methods of firewall evasion. This is because in order for the decoy scan to be successful, the default configuration for the host-based firewall should allow simultaneous scans from multiple machines, a configuration that was clearly not uniform across all operating systems. As the number of decoys increased, for all machines, irrespective of operating system, the vulnerability rate of this bypassing method reduced. This is because all operating systems could not deal with the increased amount of traffic and completely dropped the packets from the attack system.

Graph 4: Vulnerability rate for Source Port Manipulation Attack Configuration Parameters



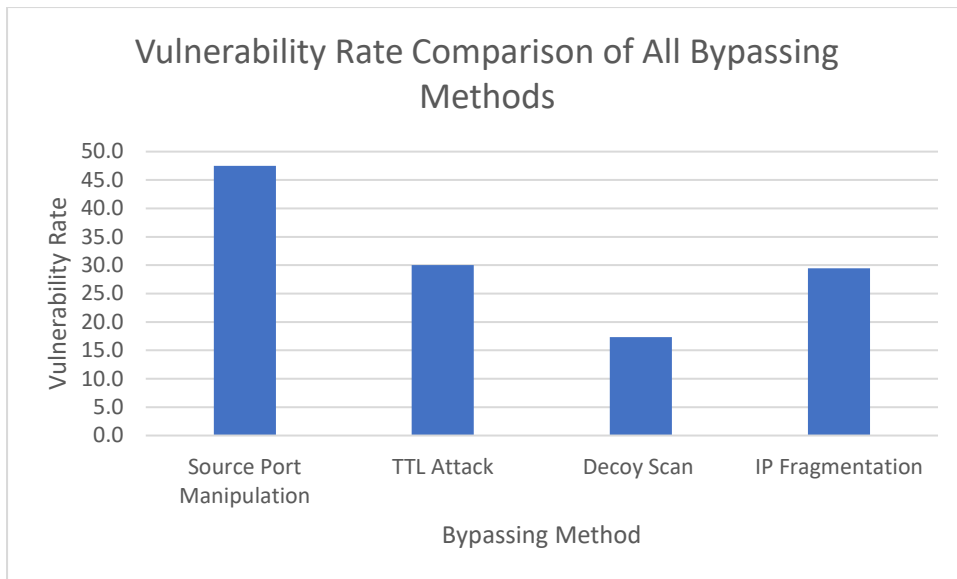
As seen in the above graph, based on the port that was being used as a source port to bypass the firewall, the vulnerability rate of bypassing the firewall changed. From the graph, HTTP was the easiest port to exploit and use as a source port, with a 60% vulnerability rate, but the vulnerability rate of using port 21 (ftp) as a source port was only slightly lower. This shows that exploitation of both services should be mitigated, which does not involve blocking traffic to the port, but instead creating additional security measures to authenticate connections through these ports. These mitigations are further described in the appendix. The reason these ports were easier to exploit than the other ports such as Kerberos is because most latest operating system versions have been able to patch the Kerberos vulnerability, but it still very much exists.

In addition, on closer examination of the data, it can be observed that across operating systems, certain nmap-based firewall bypassing port scans had a higher vulnerability rate than others, on average. The below table shows the total vulnerability rate for each bypassing port scan across all operating systems. This was done by totalling the number of successful bypasses for each parameter and each operating system, and then dividing this by the total number of trials.

Table 9: Vulnerability rate against type of attack

Type of Attack	Total Vulnerability Rate Across Operating Systems/%
Source Port Manipulation	47.5
TTL Attack	30.0
Decoy Scan	17.3
IP Fragmentation	29.4

The below graph represents this diagrammatically:



Graph 5: Vulnerability rate comparison across all bypassing methods

As seen, the source port manipulation attack has the highest vulnerability rate, making it the most potent. When examining the data for the source port manipulation attack, it can be seen that port 80 (used for http) had the highest vulnerability rate for host-based firewalls, as seen in the processed data table for source port manipulation attacks below.

Source Port Number	Vulnerability Rate /%
21 (ftp)	56.7
80 (http)	60.0
88 (kerberos)	30.0
443 (https)	43.3

4.4 Limitations of the Investigation

Despite there having been several measures taken to ensure the maximal accuracy of the investigation, there are certain limitations, primarily because it is conducted on a completely virtual environment. Since all the machines are virtual, and the investigation environment attempts to, but does not completely resemble an authentic computer network, the results may slightly vary from a normal computer setting. However, these results are still applicable in an authentic computer environment as the parameter ranges that have pose the highest vulnerability threat would still continue to do so, but the exact values may slightly vary. In addition, since the standard operating system files were used,

and the operating system files were changed for each trial, this limitation has been mitigated for the most part.

5. Conclusion and Extension of the investigation

5.1 Conclusion

Through the analyses that has been presented, it is possible for one to have a conclusive answer to the research question “How do the configuration parameters of nmap-based firewall bypassing methods affect the vulnerability of host-based firewalls?”.

Firstly, from the sample of bypassing methods used, which were provided by NMAP, the vulnerability rate of bypassing the firewall varied in a parabolic manner, as the parameters were changed. This means that for each bypassing method, there is an optimal parameter, that leads to the highest possible vulnerability of a host-based firewall. Therefore, in order to answer the question, it can be concluded that there exists an optimum value for each parameter for a firewall bypassing method where the vulnerability rate of a host-based firewall is the highest. However, once the parameter is increased past this optimum value, the vulnerability rate of the host-based firewall decreases. The vulnerability decreases past the optimum as in most cases, the packets are completely dropped by the host-based firewall as the parameter values become too abnormal.

Secondly, it is possible to infer that if there exist optimum values for scans that can make host-based firewalls vulnerable, there have to exist countermeasures that network administrators and computer users can implement to prevent the exploitation of their systems by using such parameters. These countermeasures are further explored in the appendix.

Thirdly, it is easy to infer that the services running on one's computer can lead to vulnerabilities that can then be exploited by hackers. This was seen in the source port manipulation attack, which leveraged the running of http and ftp servers on a machine, allowing hackers to create backdoors into a system through these services. The appendix discusses potential mitigations to this problem.

Finally, when analysing the data, it is evident that source port manipulation is the most potent port scan to evade firewalls, particularly when configured with the parameter of port 80. This means that highest priority should be placed on securing port 80 and mitigating scans that leverage port 80, from a user perspective. This also means that servers and systems that often use port 80, such as web servers, should be mindful of additional security measures such as implementing an active intrusion prevention system as opposed to a standard host-based firewall.

5.2 Extension of the Investigation

In order to extend this investigation, the process of system exploitation can be visited. Since the investigation deals with bypassing a firewall to find out running services, an extension could deal with exploiting the firewall and gaining access to the system. An investigation can be conducted to check how the type of firewall being used affects the vulnerability rate of a system being exploited. The expected results for this investigation would be to show that systems with a passive intrusion detection system, or application-level firewall, would be easier to exploit than those with active intrusion detection systems and a network-based firewall.

Appendix

Ethical warning: Conducting unauthorized port scans using Nmap on public networks is punishable by law. This research is for cybersecurity information purposes only.

In order to counter the exploitation of optimum parameters for attacks such as the IP fragmentation and TTL attack, users can configure their firewalls in the manner below:

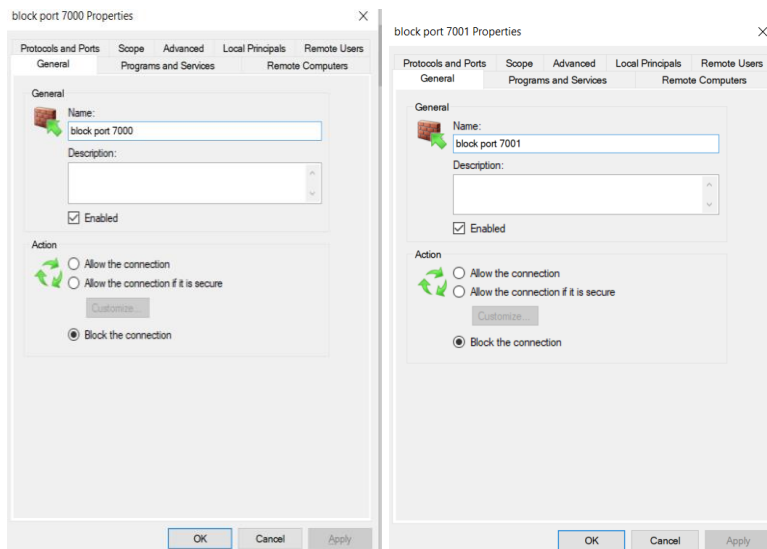


Figure 20: Blocking ports to prevent NMAP port scans on a machine

Blocking ports 7000 and 7001 blocks port scans on a machine, which means that for the TTL scan and IP fragmentation attack, the packets would be dropped irrespective of the parameters assigned for each of these values. In addition, users can block the port 7002 and 7016, which prevents their computer from being discovered on the network by other computers, which could be attempting to attack the system. For instance, blocking port 7016 blocks ICMPv6 host discovery for a system.

Blocking port 7004 (ICMPv4) through a host-based firewall prevents the exploitation of the decoy scan, as this does not allow attackers to make use of a distributed scan on a system. This means that even with the optimal configuration parameter for a decoy scan (3 decoys), the packets would still be completely dropped, resulting in no firewall evasion.

However, the above countermeasures do not secure systems against the source port manipulation attack. The below solutions help defend against such attacks.

If using FTP, users can enable the connection to the port as secure only, which would require authentication when users try to connect to the port. This prevents the exploitation of the source port attack, unless users have access to the login credentials for that ftp service. The below host-based firewall shows the mentioned solution.

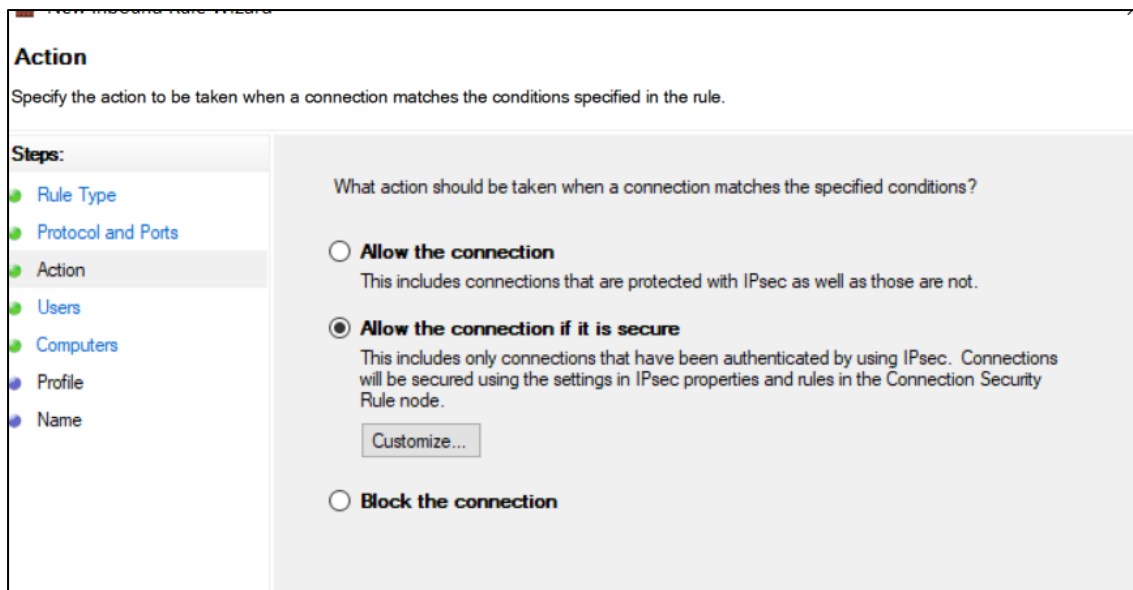


Figure 21: Writing rules to allow only secure FTP connections

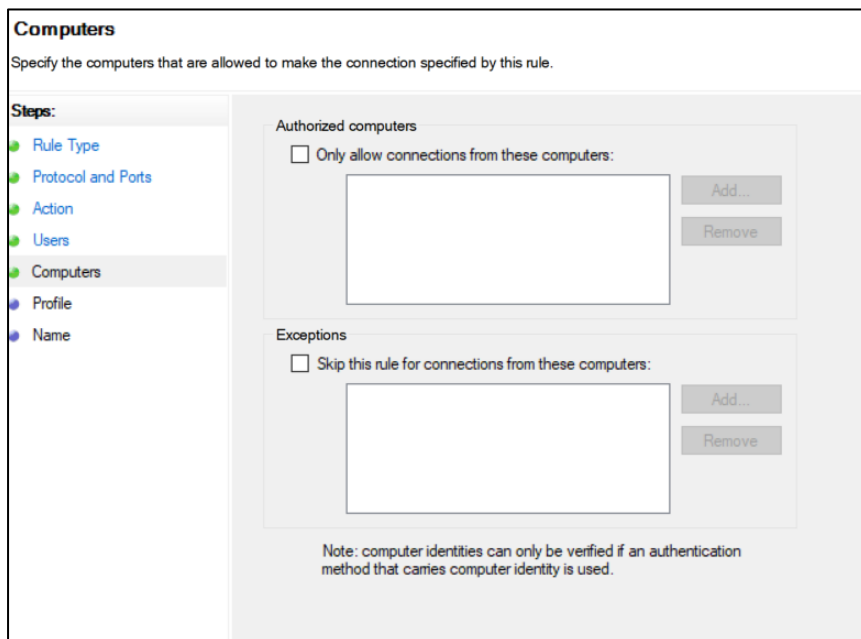


Figure 22: Allowing connections from specific users in the firewall

Users can also specify which computers can connect to the FTP port, which means that other computers trying to exploit this FTP service would not be able to, preventing malicious hackers from exploiting this service. The same configuration can be created for the other services, such as HTTP, HTTPs, and Kerberos. However, it should be kept in mind that if using a public http server, allowing connections only from specific users would not be viable as it prevents access to other systems.

In order to counter decoy scans, router path tracing and response-dropping can be used, which completely mitigate the attack.

Bibliography

“Angry IP Scanner - TTL.” *Angry IP Scanner - the Original IP Scanner for Windows, Mac and Linux*, <https://angryip.org/faq/ttl>. Accessed 11 July 2021.

“Bypassing Firewall Rules | Nmap Network Scanning.” *Nmap: The Network Mapper - Free Security Scanner*, <https://nmap.org/book/firewall-subversion.html#:~:text=Source%20Port%20Manipulation,users%20whose%20applications%20stopped%20working>. Accessed 11 July 2021.

“Firewall/IDS Evasion and Spoofing | Nmap Network Scanning.” *Nmap: The Network Mapper - Free Security Scanner*, <https://nmap.org/book/man-bypass-firewalls-ids.html>. Accessed 11 July 2021.

“IP Fragmentation in Detail - Packet Pushers.” *Packet Pushers - Where Too Much Technology Would Be Barely Enough*, <https://packetpushers.net/ip-fragmentation-in-detail/>. Accessed 11 July 2021.

“Nmap – Techniques for Avoiding Firewalls – Penetration Testing Lab.” *Penetration Testing Lab*, <https://www.facebook.com/WordPresscom>, 2 Apr. 2012, <https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/>.

“TCP 3-Way Handshake Process - GeeksforGeeks.” *GeeksforGeeks*, 5 Oct. 2017, <https://www.geeksforgeeks.org/tcp-3-way-handshake-process/>.

“TCP/IP Protocols.” *IBM - United States*, <https://www.ibm.com/docs/en/aix/7.2?topic=protocol-tcpip-protocols>. Accessed 11 July 2021.

User, Super. “How Host-Based Firewalls Work: Architecture, Rules, and Alerts.” *Apriorit*, ApriorIT, <https://www.apriorit.com/dev-blog/543-how-host-based-firewall-works>. Accessed 11 July 2021.

“Using TTL Value of Response Packets on Nmap Port Scans.” *Dev@nmap.Org*, <https://dev.nmap.narkive.com/SqucGHlz/using-ttl-value-of-response-packets-on-nmap-port-scans>. Accessed 11 July 2021.

“What Is Transmission Control Protocol (TCP)?” *Fortinet*, <https://www.fortinet.com/resources/cyberglossary/tcp-ip>. Accessed 11 July 2021.