PORT SCAN DETECTION

Index

f. OS Finger Printing
g. Aggressive Scanning
h. Stealth Scan
i. ACK Scan
j. TCP Connect Scan
k. Null Scan
l. Xmas Scan
m. Fin Scan
n. Verbosity
o. Version Detection
p. Save Results
q. Timing Templates
r. Best Practice when using Nmap
s. Nmap Scripting Engine (NSE)
t. Default Scripts
u. Banner Script
v. HTTP Methods Script
w. HTTP Enumeration Script
x. FTP Anon Login
y. FTP Brute Login
z. Vulners Script
aa. Fragmented Packets to evade Firewall/IDS
bb. Spoofing to evade IDS/Firewall
cc. Decoy to evade Firewall/IDS

6. Current Port Scan Detection Flaws
    a. Time
    b. Anomalous Packet
    c. Multiple IP's
    d. Distributed Port Scanning

Port Scan

- Port scanning is defined as a search for open ports through which intruders gain access to computers.
- This technique consists of sending a message to a port and listening for an answer. The received response indicates the port status and can be helpful in determining a host's operating system and other information relevant to launching a future attack.

Purpose of Port Scanning

- There are two general purposes that an attacker might have in conducting a port scan: a primary one, and a secondary one.
- The primary purpose is that of gathering information about the reachability and status of certain combination of IP Address and port.
- The secondary purpose is to flood intrusion detection systems with alerts, with the intension of distracting the network defenders or preventing them from doing their jobs.

Tools

- There are various tools that aim to determine a system's weakness and determine the best method for an attack.

- The best known and documented tool is Nmap. Nmap uses a variety of active probing techniques and changes the packet probes options to determine host's operating system.
- This technique consists of sending a message to a port and listening for an answer. The received response indicates the port status and can be helpful in determining a host's operating system and other information relevant to launching a future attack.

- Other port scanners include queso, checkos, and SS. However, these tools do not provide all the capabilities of nmap and thus are not as popular.

Port Scan Detection Tools

1. NSM
   - The Network Security Monitor (NSM) was the first NIDS, and also the first NIDS to detect scanning.
   - It had rules to detect any source IP address connecting to more than 15 other source IP addresses (presumably within some time window)

2. GrIDS
   - GrIDS detected portscanning by building graphs of activity in which the nodes represented hosts, and the edges represented some network traffic between hosts. Thus a

scan probe could be represented as an edge between the scanning host and the server being scanned.

- GrIDS assembled these edges into graphs based on the fact that the edges shared at least one node, and on other user definable rules. Thus scans in which all the probes had the same source IP could be detected.
- In practice, the rules were usually conditioned on time so that only scans that occurred fairly rapidly were detected.

3. Snort portscan preprocessor

- Snort is an open source lightweight network intrusion detection system based on libpcap.
- The portscan detection functionality in Snort is made possible by a preprocessor plugin.
- The Snort portscan detector attempts to look for X TCP or UDP packets sent to any number of host/port combinations from a single source host in Y seconds, where X and Y are user defined values.
- The Snort portscan detector attempts to look for X TCP or UDP packets sent to any number of host/port combinations from a single source host in Y seconds, where X and Y are user defined values.

- Upon arrival, a packet's structure is checked for soundness. The packet is then tested to see if it is part of a scan currently in progress. This is achieved by comparing the packet type and source address to those of scans currently being investigated. If it is not part of a current scan, it becomes the starting node of a new scan.

- Upon arrival, a packet's structure is checked for soundness. The packet is then tested to see if it is part of a scan currently in progress. This is achieved by comparing the packet type and source address to those of scans currently being investigated. If it is not part of a current scan, it becomes the starting node of a new scan.
- Upon arrival, a packet's structure is checked for soundness. The packet is then tested to see if it is part of a scan currently in progress. This is achieved by comparing the packet type and source address to those of scans currently being investigated. If it is not part of a current scan, it becomes the starting node of a new scan.

- The current version of the Snort portscan detector has a couple notable shortcomings that can easily be used to evade portscan detection. First, it is unable to detect scans originating from multiple hosts.
- Also, it is easy to avoid detection by increasing the time between sending scan probes.

4. Emerald

- The EMERALD system [10] from SRI International has also been used to detect portscanning, and uses a different algorithm than the usual one. EMERALD can regard each source IP address communicating with the monitored network as a subject.
- It constructs statistical profiles for subjects, and matches a short term weighted profile of subject behaviour to a long term weighted profile. When the short term profile goes far

enough into the tails of the distribution for the long term profile, EMERALD views it as suspicious.

- One of the aspects of subject behavior can be the volume of particular kinds of network traffic generated. This can be used to detect portscanning as a sudden increase in the volume of syn packets, for example, from a particular source IP.
- This approach has some limitations. It is not capable of detecting slow stealthy scans, since those will not create the kind of sharp volume increase that EMERALD looks for.

Port Scan Detection

- Several port scan detection mechanisms have been developed and are commonly included as part of intrusion detection systems.
- However, many of the detectors are easy to evade since they use simple rules that classify a port scan as more than X distinct probes within Y seconds from a single source.
- Time limit (Y) is set to be short because if it were too long, it would become difficult to manage the amount of data that the detection mechanism would need to store and process.

- Spice, a tool developed at Silicon Defense, tries to avoid this drawback.
- Spice does not use simple rules to decide if a port scan is happening. Instead, it assigns a score to each packet that is being monitored. This score is based on how likely the packet is to be part of a port scan.
- If a packet has a high score, it is stored for a longer period of time so that it can be analyzed more closely. On the other hand, if a packet is not suspicious, its data is safely discarded.
- This method of prioritizing suspicious packets over non-suspicious packets allows Spice to detect stealthy port scans more effectively.

Classification Methodology

- We classify port scans into three basic types based on the pattern of target destinations and ports the scan explores.
  1. Vertical Scans:
     - The vertical scan is a port scan that targets several destination ports on a single host.

- This scan is among the easiest to detect because only local (single host) detection mechanisms are required.

2. Horizontal Scans:

- A horizontal scan is a port scan that targets the same port on several hosts.
- Most often the attacker is aware of a particular vulnerability and wishes to find susceptible machines.
- One would expect to see many horizontal scans for a particular port immediately following the publicizing of a vulnerability on that port.
- The distribution of ports of interest changes over time as the popularity of different exploits grows in attacker community.

3. Block Scans:

- The last method mentioned is called block scanning, and it's a combination of both vertical and horizontal scanning. The attackers use this method to sweep through the entire range of address-port combinations, looking for vulnerable machines that they can exploit in the future.
- This technique can be very effective in creating a hit-list of targets for future attacks.

Normal Traffic vs Port Scans

- All the scan probe types which involve illegal flag combinations are extremely straightforward to detect. Rules which simply flag any packet with a non-standard combination of flags will detect all such scans with essentially no false positives.

- These are often referred to as “stealth scans”, they are not stealthy for our case.
- A network intrusion detection system is much more challenged by full connect scans, SYN scans, and UDP scans where the individual packets could be normal traffic.

Footprint Size

- It is helpful for us to be able to characterize how “big” a scan footprint is.
- For example, a scan of every port on every host of a full Class C network involves the attacker checking 16,646,144 distinct host/port combinations. This is going to be hard to hide. By contrast, an attacker who only wants to know whether a single port is turned on, on a single host on our network, will find it much easier to evade detection.

Methods to calculate footprint size

1. Method 1:
   - This is the simplest method in which we calculate size of a footprint to count the IP/port combinations the attacker needs to test. We call this the total size of the footprint.
   - Counting the IP/port combinations the attacker needs to test is often a good metric of how big her scan is and how much information she can obtain from the target network.
   - Sometimes the attacker is trying to obtain specific information from each IP/port combination, such as the operating system being used. In these cases, simply counting the number of IP/port combinations may not give an accurate measure of the amount of information the attacker is trying to obtain.

2. Method 2:

- If a scan probe finds an open port, it is much harder for us to determine that this was a scan than if it hits a closed port. This is because, other than some misconfiguration, normal traffic will not hit closed ports very often.
- Connections to closed ports are inherently a lot more suspicious than connections to open ports.
- So, another measure of the size of a scan footprint is the closed size of it – the number of distinct port/IP combinations the scan is targeting which are in fact closed at the time of scanning.

3. Method 3:

- One approach to port scan detection is not to look at the scan packets (which we will call forward scan detection), but rather to look for packets that could be responses to port scan probes from closed server ports; TCP resets in the case of TCP scans, and ICMP port unreachable packets in the case of UDP scans. We refer to this as backward scan detection.
- The advantage of backward scan detection is that the packets used are more unusual than the packets used in "forward scan detection".
- However, this method has a drawback because it may not detect port scans into empty IP addresses. An empty IP address refers to an IP address that does not have any active hosts or devices connected to it. An attacker might be attempting to scan a range of IP addresses to identify vulnerable systems, but some of the IP addresses being

scanned may not have any devices or hosts connected to them. When the backward scan detection method is used, these empty IP addresses will not trigger any responses, and therefore, the scanning activity will go undetected.

4. Method 4:

- The final measure is called an anomaly score and it is based on the probability of normal traffic going to certain ports and IP addresses. The more unlikely it is for normal traffic to target a particular port/IP combination, the higher the anomaly score will be for that combination.
- The total anomaly score of a network scan is calculated by adding up the anomaly scores of all the port/IP combinations being scanned.
- It's important to know that the anomaly score is specific to each network, so two networks with different traffic patterns will have different anomaly scores for the same scan. The probability distribution of traffic also changes over time.

Port Scan Avoidance

1. Change the scan order:

- Most scans in the wild at present move through IP addresses sequentially, going from lower to higher. However, if this assumption is used by defenders in detection, it is straightforward for attackers to change it.
- Randomizing the order in which IP addresses and ports are searched can easily be done. nmap is currently capable of randomizing the addresses it uses within blocks of 2048 hosts.

2. Slowing down:

- One way to avoid detection is to increase the time between consecutive probes.
- This technique works since most intrusion detection systems look for X events in a Y sized time window and can only keep a limited amount of state.

3. Affecting the source address:

- An attacker can also conceal her IP address by using IP decoys, or “zombie” computers under an attacker’s control. Different scans originates from several IPs.
- In the simplest case, the attacker does need to use a real source address, since she needs to see the packets that servers generate in response to the scan in order to know what ports are actually open.

4. Distributed Scanning:

- An attacker who can launch her scan from a number of different real IP addresses can investigate different parts of

the footprint from different places. This complicates the detection task.

- In the extreme case, large networks of agents similar to those used for distributed denial of service attacks could be used for port scanning.

Nmap

Definition

- Nmap stands for Network Mapper.
- With the help of Nmap, we can map a network in terms of what kind of devices are out there.

- It also helps in figuring out what ports are open and available on that device. What services are running and what type of applications could be supported by that device.
- It also helps in figuring out versions and types of servers, like HTTP or SQL servers.
- It also helps in scanning certain vulnerabilities on that device.

Open/Closed/Filtered Ports

Open Port:

- When we send a TCP SYN for a port and we get a SYN ACK back, then it means port is available and open.

Closed Port:

- When we send a TCP SYN to a server with a closed port, then we see a RST back and we call them closed ports

Filtered Port:

- When we send a TCP SYN to a server and we neither got a SYN ACK nor a RST, then it means that the port we are scanning is being filtered either by a firewall or some other device that doesn't let us know that its there.

Simple Nmap Scans

Individual Domain/IP Scans

- Domain: nmap scanme.nmap.org
- IP: nmap 192.168.1.9

- Both of these scans will scan top 1000 ports.

Subnet Scans

- One Subnet: nmap 192.168.1.0/24
- Multiple Subnets: nmap 192.168.1.0/24 10.211.55.0/24

IP Range Scans

- nmap 192.168.1.0-20
- Scans IP Addresses from 192.168.1.0 to 192.168.1.20

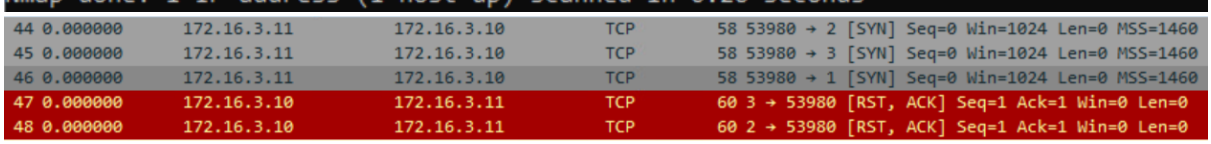
-F Switch

- nmap -F 192.168.1.0 scans top 100 ports. F means fast.

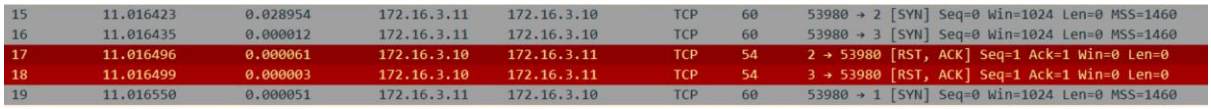
-p Switch

- nmap -p80,443 192.168.1.1 scans ports 80 and 443
- Configuration – If no. of ports scanned exceed more than 2, generate alert.
  Attacker:

```
C:\Users\Admin>nmap -p1,2,3 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 02:42 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.00s latency).

PORT  STATE    SERVICE
1/tcp filtered tcpmux
2/tcp closed   compressnet
3/tcp closed   compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
```



Target:



15	11.016423	0.028954	172.16.3.11	172.16.3.10	TCP	60	53980 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	11.016435	0.000012	172.16.3.11	172.16.3.10	TCP	60	53980 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	11.016496	0.000061	172.16.3.10	172.16.3.11	TCP	54	2 → 53980 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
18	11.016499	0.000003	172.16.3.10	172.16.3.11	TCP	54	3 → 53980 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
19	11.016550	0.000051	172.16.3.11	172.16.3.10	TCP	60	53980 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

- Alert is generated at target.

- nmap -p1-100 192.168.1.1 scans ports from 1 to 100
- nmap -p- 192.168.1.1 scans every port from 1 to 65535 but it will take a long time and it will cause lot of noise.

--top-ports Switch

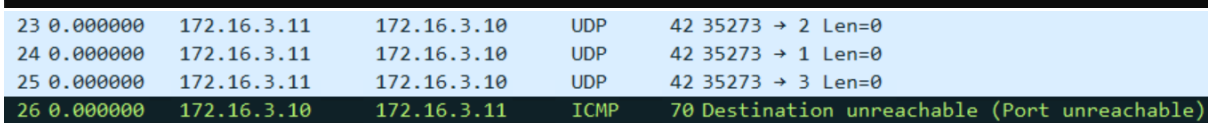
- nmap –top-ports 20 192.168.1.1
- Scans top 20 ports that are commonly open on devices.

UDP Scans

- If we do not specify any protocols during scan, it by default scans over TCP.
- We use -sU to scan over UDP.
- nmap -p80,443 -sU 192.168.1.1 will scan ports 80 and 443 over UDP.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

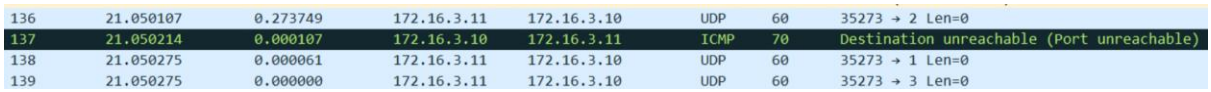
  Attacker:

```
C:\Users\Admin>nmap -p1,2,3 -sU -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 03:07 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.0010s latency).

PORT  STATE         SERVICE
1/udp open|filtered tcpmux
2/udp closed        compressnet
3/udp open|filtered compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```



23	0.000000	172.16.3.11	172.16.3.10	UDP	42	35273 → 2 Len=0
24	0.000000	172.16.3.11	172.16.3.10	UDP	42	35273 → 1 Len=0
25	0.000000	172.16.3.11	172.16.3.10	UDP	42	35273 → 3 Len=0
26	0.000000	172.16.3.10	172.16.3.11	ICMP	70	Destination unreachable (Port unreachable)

Target:



136	21.050107	0.273749	172.16.3.11	172.16.3.10	UDP	60	35273 → 2 Len=0
137	21.050214	0.000107	172.16.3.10	172.16.3.11	ICMP	70	Destination unreachable (Port unreachable)
138	21.050275	0.000061	172.16.3.11	172.16.3.10	UDP	60	35273 → 1 Len=0
139	21.050275	0.000000	172.16.3.11	172.16.3.10	UDP	60	35273 → 3 Len=0

- Alert is generated at target.

- If the port is closed, the server should respond with an ICMP message of Type 3 and Code 3 (Destination Unreachable because Port Unreachable).
- UDP scanning involves listening for the ICMP responses from closed ports and then assuming that any ports that do not respond are open.

Ping Scans

- nmap -sn 192.168.1.0/24
- -sn is used for ping scan.

- If we run -sn for ping scan in non-privilege mode, then we don't actually send ICMP.
- Ping is actually a TCP scan.

- To send ICMP, we use sudo nmap -sn 192.168.1.1

- sudo nmap -sn 192.168.1.0/24 will be used to check what devices are active on the subnet.
- Most of the packets that are sent are ARPs because we are first finding if the device with a particular address exists in the local environment or not. If information about an address is already present in ARP cache, then ARP packet will not be sent.

- If the station is available on the local subnet, we doing a 3-way handshake and after completion of that we will reset the connection which will complete ping.

- Ping scan scans for TCP port 80 and 443

OS Finger Printing

- sudo nmap -O 192.168.1.1 will try to get a finger print about the TCP stack running on the target device.
- But when the device is running a very tight firewall, OS enumeration is not performed properly and it will not be able to identify ports and OS running on the target.
- Therefore, -O is used to finger print OS running on target and matches with the predefined database of OS in nmap and guess what OS is running on the target.

- If we see a lot of ARP activity from the same device, then it means that they are trying to scan and find out devices that are active on the network.

Aggressive Scanning

- sudo nmap -A 192.168.1.1 will scan the target from top to bottom.
- OS fingerprinting, traceroute, all ports, versions, etc are scanned with -A.
- It will cause a lot of noise.

Stealth Scan

- sudo nmap -sS scanme.nmap.org is called as stealth scan.

- In stealth scan, we send SYN, receive SYN-ACK and reset the connection. Since, we did not fully complete the 3-way handshake there will not be any log on the server and hence it is stealthy.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

  Attacker:

```
C:\Users\Admin>nmap -p135 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 03:32 Pacific Daylight Time
Nmap scan report for 172.16.3.10
Host is up (0.0010s latency).

PORT    STATE SERVICE
135/tcp open  msrpc
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
7	2.783431	172.16.3.11	172.16.3.10	TCP	58	38361 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	2.783849	172.16.3.10	172.16.3.11	TCP	60	135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
10	3.797000	172.16.3.10	172.16.3.11	TCP	60	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
14	5.798609	172.16.3.10	172.16.3.11	TCP	60	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
24	9.811562	172.16.3.10	172.16.3.11	TCP	60	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
56	17.826427	172.16.3.10	172.16.3.11	TCP	60	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
62	23.827762	172.16.3.10	172.16.3.11	TCP	60	135 → 38361 [RST] Seq=1 Win=0 Len=0

  Target:

No.	Time	Source	Destination	Protocol	Length	Info
13	8.393096	172.16.3.11	172.16.3.10	TCP	60	38361 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	8.393320	172.16.3.10	172.16.3.11	TCP	58	135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
16	9.406389	172.16.3.10	172.16.3.11	TCP	58	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=
21	11.408043	172.16.3.10	172.16.3.11	TCP	58	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=
30	15.420987	172.16.3.10	172.16.3.11	TCP	58	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=
85	23.435854	172.16.3.10	172.16.3.11	TCP	58	[TCP Retransmission] 135 → 38361 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=
91	29.437151	172.16.3.10	172.16.3.11	TCP	54	135 → 38361 [RST] Seq=1 Win=0 Len=0

- No Alert is generated as only one port is scanned.

- Stealth scan is the default scan nmap will perform if we do not specify the type of scan we want to do.

- If we look at SYN sent by nmap, window size is 1024 and options MSS is 1460 bytes.
- This means we are saying to the receiver that we can handle only 1024 bytes at a time but MSS is 1460 which means receiver can send a segment of 1460 bytes which is encapsulated with TCP, IP and Ethernet. But receiver buffer is 1024.

- If we apply 1024 window size as display filter, then that will mean it is nmap.
- Therefore, stealth scan is easy to find because its window size is 1024 always.

ACK Scan

- sudo nmap -sA scanme.nmap.org is called as ACK scan.
- When we perform an ACK scan, we send an acknowledgement packet and we will get a RST which means that endpoint port is listening or is available.
- Its also possible that some other device might have sent the RST because we are sending out-of-context ACK without establishing any connection.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

  Attacker:

```
C:\Users\Admin>nmap -sA -p1,2,3 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 04:07 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.00089s latency).

PORT  STATE      SERVICE
1/tcp unfiltered tcpmux
2/tcp filtered   compressnet
3/tcp unfiltered compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.62 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
24	10.394303	172.16.3.11	172.16.3.10	TCP	54	59673 → 1 [ACK] Seq=1 Ack=1 Win=1024 Len=0
25	10.394440	172.16.3.11	172.16.3.10	TCP	54	59673 → 3 [ACK] Seq=1 Ack=1 Win=1024 Len=0
26	10.394502	172.16.3.11	172.16.3.10	TCP	54	59673 → 2 [ACK] Seq=1 Ack=1 Win=1024 Len=0
27	10.394608	172.16.3.10	172.16.3.11	TCP	60	1 → 59673 [RST] Seq=1 Win=0 Len=0
28	10.395194	172.16.3.10	172.16.3.11	TCP	60	3 → 59673 [RST] Seq=1 Win=0 Len=0

  Target:

No.	Time	Source	Destination	Protocol	Length	Info
7	2.858926	172.16.3.11	172.16.3.10	TCP	60	59673 → 1 [ACK] Seq=1 Ack=1 Win=1024 Len=0
8	2.858953	172.16.3.11	172.16.3.10	TCP	60	59673 → 3 [ACK] Seq=1 Ack=1 Win=1024 Len=0
9	2.858982	172.16.3.10	172.16.3.11	TCP	54	1 → 59673 [RST] Seq=1 Win=0 Len=0
10	2.858990	172.16.3.10	172.16.3.11	TCP	54	3 → 59673 [RST] Seq=1 Win=0 Len=0
11	2.859043	172.16.3.11	172.16.3.10	TCP	60	59673 → 2 [ACK] Seq=1 Ack=1 Win=1024 Len=0

- Alert is generated at target.

TCP Connect Scan

- nmap -sT scanme.nmap.org is called as TCP connect scan.
- In this scan a full TCP connection is established. We send SYN, receives SYN-ACK and send ACK and then send RST.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

  Attacker:

```
C:\Users\Admin>nmap -sT -p135 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 04:16 Pacific Daylight Time
Nmap scan report for 172.16.3.10
Host is up (0.0010s latency).

PORT    STATE SERVICE
135/tcp open  msrpc
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
11	7.488181	172.16.3.11	172.16.3.10	TCP	66	56533 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
12	7.488560	172.16.3.10	172.16.3.11	TCP	66	135 → 56533 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_
13	7.488667	172.16.3.11	172.16.3.10	TCP	54	56533 → 135 [ACK] Seq=1 Ack=1 Win=262656 Len=0
14	7.488797	172.16.3.11	172.16.3.10	TCP	54	56533 → 135 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

  Target:

No.	Time	Source	Destination	Protocol	Length	Info
16	12.968333	172.16.3.11	172.16.3.10	TCP	66	56533 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
17	12.968555	172.16.3.10	172.16.3.11	TCP	66	135 → 56533 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SAC
18	12.968778	172.16.3.11	172.16.3.10	TCP	60	56533 → 135 [ACK] Seq=1 Ack=1 Win=262656 Len=0
19	12.968945	172.16.3.11	172.16.3.10	TCP	60	56533 → 135 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

- Alert is not generated since we are only scanning one port.

- Window size is not common, like 1024 in stealth scan. It is looking like real TCP connection establishment which is more normal but we are risking the other side logging this that the connection was attempted.

Null Scan

- nmap -sN scanme.nmap.org is called as Null scan.
- If we look at TCP packets, there will be no flags set. This scan sends an illegal packet which involves no flag set.
- We will get a RST from this scan.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

Attacker:

```
C:\Users\Admin>nmap -sN -p1,2,3 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 04:24 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.00088s latency).

PORT  STATE         SERVICE
1/tcp closed        tcpmux
2/tcp closed        compressnet
3/tcp open|filtered compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
8	4.115492	172.16.3.11	172.16.3.10	TCP	54	44182 → 2 [<None>] Seq=1 Win=1024 Len=0
9	4.115588	172.16.3.11	172.16.3.10	TCP	54	44182 → 1 [<None>] Seq=1 Win=1024 Len=0
10	4.115656	172.16.3.11	172.16.3.10	TCP	54	44182 → 3 [<None>] Seq=1 Win=1024 Len=0
11	4.115772	172.16.3.10	172.16.3.11	TCP	60	2 → 44182 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
12	4.115829	172.16.3.10	172.16.3.11	TCP	60	1 → 44182 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Target:

No.	Time	Source	Destination	Protocol	Length	Info
25	10.397597	172.16.3.11	172.16.3.10	TCP	60	44182 → 2 [<None>] Seq=1 Win=1024 Len=0
26	10.397686	172.16.3.10	172.16.3.11	TCP	54	2 → 44182 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	10.397696	172.16.3.11	172.16.3.10	TCP	60	44182 → 3 [<None>] Seq=1 Win=1024 Len=0
28	10.397746	172.16.3.11	172.16.3.10	TCP	60	44182 → 1 [<None>] Seq=1 Win=1024 Len=0
29	10.397778	172.16.3.10	172.16.3.11	TCP	54	1 → 44182 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

- Alert is generated at target.

Xmas Scan

- nmap -sX scanme.nmap.org is called as Xmas scan.
- In this scan Urgent, Push and Fin bits are set. It is called as Xmas scan because flags are up like Xmas tree. It will also result in RST.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

Attacker:

```
C:\Users\Admin>nmap -sX -p1,2,3 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 04:36 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.00023s latency).

PORT  STATE         SERVICE
1/tcp closed        tcpmux
2/tcp open|filtered compressnet
3/tcp closed        compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
43	25.228270	172.16.3.11	172.16.3.10	TCP	54	64677 → 1 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
44	25.228373	172.16.3.11	172.16.3.10	TCP	54	64677 → 3 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
45	25.228447	172.16.3.11	172.16.3.10	TCP	54	64677 → 2 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
46	25.228531	172.16.3.10	172.16.3.11	TCP	60	1 → 64677 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
47	25.228568	172.16.3.10	172.16.3.11	TCP	60	3 → 64677 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0

Target:

No.	Time	Source	Destination	Protocol	Length	Info
239	38.939309	172.16.3.11	172.16.3.10	TCP	60	64677 → 1 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
240	38.939371	172.16.3.11	172.16.3.10	TCP	60	64677 → 3 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
241	38.939377	172.16.3.10	172.16.3.11	TCP	54	1 → 64677 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
242	38.939430	172.16.3.10	172.16.3.11	TCP	54	3 → 64677 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
243	38.939442	172.16.3.11	172.16.3.10	TCP	60	64677 → 2 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0

- Alert is generated at target.

Fin Scan

- nmap -sF scanme.nmap.org is called as FIN scan.
- In this also we receive RST packet.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

  Attacker:

```
C:\Users\Admin>nmap -sF -p1,2,3 -T5 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 04:43 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.00076s latency).

PORT  STATE         SERVICE
1/tcp closed        tcpmux
2/tcp closed        compressnet
3/tcp open|filtered compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
50	15.961693	172.16.3.11	172.16.3.10	TCP	54	49485 → 1 [FIN] Seq=1 Win=1024 Len=0
51	15.961779	172.16.3.11	172.16.3.10	TCP	54	49485 → 2 [FIN] Seq=1 Win=1024 Len=0
52	15.961845	172.16.3.11	172.16.3.10	TCP	54	49485 → 3 [FIN] Seq=1 Win=1024 Len=0
53	15.961890	172.16.3.10	172.16.3.11	TCP	60	1 → 49485 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
54	15.961946	172.16.3.10	172.16.3.11	TCP	60	2 → 49485 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0

Target:

No.	Time	Source	Destination	Protocol	Length	Info
10	4.844168	172.16.3.11	172.16.3.10	TCP	60	49485 → 1 [FIN] Seq=1 Win=1024 Len=0
11	4.844220	172.16.3.10	172.16.3.11	TCP	54	1 → 49485 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
12	4.844280	172.16.3.11	172.16.3.10	TCP	60	49485 → 2 [FIN] Seq=1 Win=1024 Len=0
13	4.844294	172.16.3.10	172.16.3.11	TCP	54	2 → 49485 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
14	4.844294	172.16.3.11	172.16.3.10	TCP	60	49485 → 3 [FIN] Seq=1 Win=1024 Len=0

- The purpose of TCP null, Xmas and FIN scan is to take advantage of loophole in TCP RFC as well as to bypass firewalls.

- If we perform FIN scan, then it does not mean anything and some devices in old days does not respond with RST packets to this scan which would mean that the port is open.
- Therefore, we were able to know that the port is open without establishing connection but if we send a RST, then it means port is closed.

- Nowadays, most of devices will give RST for all these scans.

Verbosity

- nmap -O 192.168.1.1 -v will do OS footprinting.
- -v will make the scan more verbose which means it will give us more information.
- The more v's we put the more information we get at the time of scanning.

Version Detection

- nmap -sV scanme.nmap.org will determine the version of services running on the target.
- It first identified which ports are open or filtered or closed. Then, it feeds that information to version identification to identify version.
- Using request and responses, like GET request and OK response, nmap determine the version of service running on target.

Save Results

- To save the results of nmap, we use -oN switch.
- If we run nmap -sV scanme.nma.org -oN scanme.txt will save the result of version detection in scanme.txt.

Timing Templates

- One way to make our scans faster is by using timing template switch.
- Timing templates allows us how aggressively we want our scan to perform. Do we want scan to be quick but it will cause a lot of noise and might miss a thing or two or do we want scan to be slow that it will scan for all things and might evade IDS/IPS on the way.

- There are 6 timing templates: T0, T1, T2, T3, T4, T5.
- If we don't set a timing template, T3 is default.
- nmap scanme.nmap.org -T4 is the way we can use timing template.

Best Practice when using Nmap

- Best practice in order to perform an optimal scan:
  1. Use a ping scan (-sn) to avoid scanning lots of different TCP ports
  2. Avoid -A and -O if possible to avoid noise
  3. Use -p to specify ports to avoid scanning all 65535 or top 1000 ports

Nmap Scripting Engine (NSE)

- Nmap Scripting Engine (NSE) allows us to automate scans.
- We can create scripts that will run different type of scans, feed different parameters and do whole lot of work without us needing to be on command line typing one scan at a time. This saves us a lot of time.

- Scripts can do things like, network discovery, version detection, vulnerability detection, backdoor detection and backdoor exploitation. Therefore, nmap is not just a port scanner, it can do vulnerability detection, exploitation, etc.
- These scripts are written in lua programming language.

- Lot of these scripts are noisy as they run lot of things at the same time. They are easier to find then these simple scans from command line.

- nmap installation comes with prepackaged with a built-in script database.
- To get to script database, we go to nmap/scripts.

- We can read these scripts using cat or more in cmd.
- These are most useful and most common scripts shared by nmap community.
- nmap scripts has various categories, like default, vulnerability, authentication, etc.

- We can update the script database with nmap --script-updatedb.

Default Scripts

- If we go to nmap.org, then we will find a list of scripts that are tagged as default. If we run -sC or -A, then these scripts will be automatically be run.
- Default scripts include, OS discovery, version detection, sniffer-detect, sql injection, etc.

- Default scripts can be run as nmap -sC scanme.nmap.org or nmap -script=default scanme.nmap.org.
- We can also specify the script name that we want to run, like nmap --script default or nmap --script "http-auth" scanme.nmap.org.
- We can also run several scripts against a target, like nmap --script "ftp-*" scanme.nmap.org, this will run all the scripts nmap has against that ftp server.

Banner Script

- nmap ---script "banner" scanme.nmap.org will grab initial banner for any application that is active on the target. It will go to all the ports that are open and run banner script against them and returns the result.

HTTP Methods Script

- nmap --script "http-methods" scanme.nmap.org allows us to know what are all the supported methods on the target. It let us know what type of request we can run on the server.

HTTP Enumeration Script

- nmap --script "http-enum" scanme.nmap.org will allow us to find any hidden folders or private directories that are hidden on the front page.
- We will get a response saying /images and /shared are interesting directories.
- We can go on to website and type these directories to see what is contained in them, like scanme.nmap.org/images and scanme.nmap.org/shared.

- Nmap gave us the directories what it found to be interesting but we can find all directories that gave response to nmap request.
- When we run the script, nmap generates various requests according to script and got responses for some. We can go on to Wireshark and see the responses that nmap got and see all the directories that responded and go to their URLs.

FTP Anon Login

- nmap --script "ftp-anon" 192.168.56.102 -p21 will check any anonymous login is accepted on the server. -p 21 specifies we are scanning only for port 21 which is ftp port.
- We can then go to Wireshark and follow TCP stream of FTP port anonymous login to see the ID and Password of anonymous login.

FTP Brute Login

- nmap --script "ftp-brute" 192.168.56.101 -p21 will brute force the login/password with FTP. This can be used when we found an FTP server and anonymous login does not work.

Vulners Script

- Vulner script will show us the common vulnerabilities and exposers for a given service.
- nmap on a given target will first look at open ports, goes through their versions and then look up those versions on a vulnerability database and give us the cve's would work against that application.

- According nmap.org, vulnerability database is 250GB.

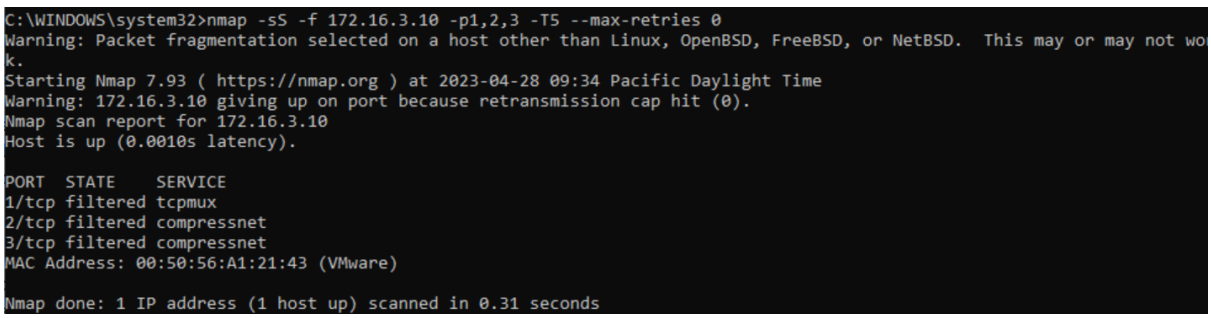
- We can run nmap -sV --script "vulners" 192.168.56.104.
- This will give us the list of services and vulnerabilities and cve's.

Fragmented Packets to evade Firewall/IDS

- -f is used for fragmentation, it causes the scan to use tiny fragmented IP packets.
- The idea is to split up the TCP header over several packets to make it harder for packet filters, intrusion detection systems and other devices to detect what we are doing.

- By default -f will take the packets generated by whatever scan we ran and break those packet into 8 bytes each and send them out.

- nmap -sS -f scanme.nmap.org -p80 will do stealth scan with fragmented packets.
- We can look at fragmented packet in Wireshark and see them how they are reassembled and how they are transferred.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.
  Attacker:



No.	Time	Source	Destination	Protocol	Length	Info
6	2.417675	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=b1ee) [Reassemblec
7	2.417774	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=b1ee) [Reassemblec
8	2.417823	172.16.3.11	172.16.3.10	TCP	42	62164 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	2.417877	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=9303) [Reassemblec
10	2.417919	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=9303) [Reassemblec
11	2.417963	172.16.3.11	172.16.3.10	TCP	42	62164 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	2.418009	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=63ef) [Reassemblec
13	2.418115	172.16.3.11	172.16.3.10	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=63ef) [Reassemblec
14	2.418161	172.16.3.11	172.16.3.10	TCP	42	62164 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Target:

No.	Time	Source	Destination	Protocol	Length	Info
23	6.220182	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=b1ee) [Reassembled
24	6.220182	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=b1ee) [Reassembled
25	6.220276	172.16.3.11	172.16.3.10	TCP	60	62164 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
26	6.220276	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=9303) [Reassembled
27	6.220310	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=9303) [Reassembled
28	6.220353	172.16.3.11	172.16.3.10	TCP	60	62164 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	6.220397	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=63ef) [Reassembled
30	6.220505	172.16.3.11	172.16.3.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=63ef) [Reassembled
31	6.220545	172.16.3.11	172.16.3.10	TCP	60	62164 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

- We got an alert at target.

- This scan is now patched in latest firewalls and IDS and they will look for it and we will get caught.

Spoofing to evade IDS/Firewall

- With nmap, we can spoof the IP Address from which we are originating from. We will tell our adapter to lie about our IP Address. The target will respond to the spoofed IP Address, therefore we should know what address we are spoofing.
- We can know about the interface from which we want to capture from using dumpcap -D or nmap --iflist.

- nmap --sS -p80 -S 192.168.1.200 -e eth4 scanme.nmap.org is the command we run for spoofed scan. -S will say to use the spoofed IP Address specified i.e. 192.168.1.200 and -e will tell which interface to use.
- If we capture the process using Wireshark, we will see the source as spoofed IP Address.

- Router will get the response but then it will ARP and check who as 192.168.1.200. Since, this is not our address, therefore we will not be receiving the response.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.

  Attacker:

```
C:\WINDOWS\system32>nmap -sS -p1,2,3 -T5 --max-retries 0 -S 172.16.1.200 -e eth0 172.16.3.10
WARNING: If -S is being used to fake your source address, you may also have to use -e <interface> and -Pn .  If you are
using it to specify your real source address, you can ignore this warning.
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 09:58 Pacific Daylight Time
NSOCK ERROR [0.1600s] mksock_bind_addr(): Bind to 172.16.1.200:0 failed (IOD #1): The requested address is not valid in
its context.  (10049)
NSOCK ERROR [0.1600s] mksock_bind_addr(): Bind to 172.16.1.200:0 failed (IOD #2): The requested address is not valid in
its context.  (10049)
NSOCK ERROR [0.1610s] mksock_bind_addr(): Bind to 172.16.1.200:0 failed (IOD #3): The requested address is not valid in
its context.  (10049)
NSOCK ERROR [0.1610s] mksock_bind_addr(): Bind to 172.16.1.200:0 failed (IOD #4): The requested address is not valid in
its context.  (10049)
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.0010s latency).

PORT  STATE    SERVICE
1/tcp filtered tcpmux
2/tcp filtered compressnet
3/tcp filtered compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

13	9.037107	172.16.1.200	172.16.3.10	TCP	58	62762 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	9.037208	172.16.1.200	172.16.3.10	TCP	58	62762 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
15	9.037281	172.16.1.200	172.16.3.10	TCP	58	62762 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

  Target:

No.	Time	Source	Destination	Protocol	Length	Info
22	3.417645	172.16.1.200	172.16.3.10	TCP	60	62762 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
23	3.417703	172.16.3.10	172.16.1.200	TCP	54	1 → 62762 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	3.417717	172.16.1.200	172.16.3.10	TCP	60	62762 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
25	3.417760	172.16.3.10	172.16.1.200	TCP	54	3 → 62762 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
26	3.417766	172.16.1.200	172.16.3.10	TCP	60	62762 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

- Alert is generated at target.

- Bettercap or Ettercap can be used for receiving the response because they are more tricky.
- Therefore, a lot of times when we are spoofing we will not get a response, this can be used for DDOS.

Decoy to evade Firewall/IDS

- Decoy or cloaked scan is when we hide our real address in a bunch of other request that come from other IPs. Therefore, we will get a response from the target and the target is going to think that it is being request from a bunch of IPs.

- When we cloak a scan, the target will be able to tell who is decoy and who we really are.

- nmap -sS -p80 -D 45.45.45.45,100.100.100.100,192.168.1.109 -e eth4 scanme.nmap.org is the command we ran.
- -D is used to specify the decoys and we have to type our real IP Address in between them to receive a response.
- If we look at Wireshark, we will be able to see the request and responses from all the specified IPs and it will have our real IP as well.

- We can let nmap to randomly choose IP Addresses along with our IP Address and send the nmap scan as nmap -sS -p80 -D RND:20 -e eth4 scanme.nmap.org will randomly choose 20 IP Addresses and send requests using them.
- In this case also we will get a response and all other IPs nmap chooses will get a response too.
- Configuration – If no. of ports scanned exceed more than 2, generate alert.
  Attacker:

```
C:\WINDOWS\system32>nmap -p1,2,3 --max-retries 0 -T5 -D RND:5 -e eth0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-28 10:03 Pacific Daylight Time
Warning: 172.16.3.10 giving up on port because retransmission cap hit (0).
Nmap scan report for 172.16.3.10
Host is up (0.0010s latency).

PORT  STATE    SERVICE
1/tcp closed   tcpmux
2/tcp filtered compressnet
3/tcp closed   compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
17	12.778465	97.18.69.199	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
18	12.778570	102.119.254.189	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
19	12.778633	101.224.222.225	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
20	12.778691	114.86.214.213	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
21	12.778744	172.16.3.11	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
22	12.778794	79.253.127.188	172.16.3.10	TCP	58	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
23	12.778842	97.18.69.199	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
24	12.778859	172.16.3.10	172.16.3.11	TCP	60	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
25	12.778924	102.119.254.189	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
26	12.778995	101.224.222.225	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27	12.779051	114.86.214.213	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
28	12.779105	172.16.3.11	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	12.779160	79.253.127.188	172.16.3.10	TCP	58	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
30	12.779221	172.16.3.10	172.16.3.11	TCP	60	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31	12.779271	97.18.69.199	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
32	12.779323	102.119.254.189	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
33	12.779370	101.224.222.225	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
34	12.779417	114.86.214.213	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
35	12.779467	172.16.3.11	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
36	12.779523	79.253.127.188	172.16.3.10	TCP	58	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Target:

No.	Time	Source	Destination	Protocol	Length	Info
36	5.969703	97.18.69.199	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
37	5.969736	102.119.254.189	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38	5.969767	172.16.3.10	97.18.69.199	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
39	5.969786	172.16.3.10	102.119.254.189	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
40	5.969824	101.224.222.225	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
41	5.969824	114.86.214.213	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
42	5.969851	172.16.3.10	101.224.222.225	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
43	5.969867	172.16.3.10	114.86.214.213	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
44	5.969890	172.16.3.11	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
45	5.969906	172.16.3.10	172.16.3.11	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
46	5.969965	79.253.127.188	172.16.3.10	TCP	60	43297 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
47	5.969992	172.16.3.10	79.253.127.188	TCP	54	3 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
48	5.970025	97.18.69.199	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49	5.970045	172.16.3.10	97.18.69.199	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
50	5.970101	102.119.254.189	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	5.970118	172.16.3.10	102.119.254.189	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
52	5.970151	101.224.222.225	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
53	5.970169	172.16.3.10	101.224.222.225	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
54	5.970211	114.86.214.213	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
55	5.970226	172.16.3.10	114.86.214.213	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
56	5.970260	172.16.3.11	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
57	5.970273	172.16.3.10	172.16.3.11	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
58	5.970303	79.253.127.188	172.16.3.10	TCP	60	43297 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
59	5.970319	172.16.3.10	79.253.127.188	TCP	54	1 → 43297 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60	5.970416	97.18.69.199	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
61	5.970464	102.119.254.189	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
62	5.970509	101.224.222.225	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
63	5.970553	114.86.214.213	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
64	5.970612	172.16.3.11	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
65	5.970663	79.253.127.188	172.16.3.10	TCP	60	43297 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Current Port Scan Detection Flaws

Time

- We are detecting port scan attacks by figuring out X ports scanned in Y seconds by the same IP.
- This approach is not efficient because attacking tools are now configured such that they can scan their targets as slow as they want and as fast as they want.
- For example, in nmap we can configure the attack to be slow to evade our port scan detection logic.

- We have set configuration as if more than 2 ports are scanned within 6 seconds by the same IP, then we will generate an alert.

  Attacker:

```
C:\WINDOWS\system32>nmap -p1,2,3 -T1 --max-retries 0 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-29 05:04 Pacific Daylight Time
Nmap scan report for 172.16.3.10
Host is up (0.00013s latency).

PORT  STATE  SERVICE
1/tcp closed tcpmux
2/tcp closed compressnet
3/tcp closed compressnet
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 60.20 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
109	33.790458	172.16.3.11	172.16.3.10	TCP	58	38384 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
110	33.790806	172.16.3.10	172.16.3.11	TCP	60	2 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
125	48.798484	172.16.3.11	172.16.3.10	TCP	58	38384 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
126	48.798796	172.16.3.10	172.16.3.11	TCP	60	3 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
140	63.812582	172.16.3.11	172.16.3.10	TCP	58	38384 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
141	63.812940	172.16.3.10	172.16.3.11	TCP	60	1 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

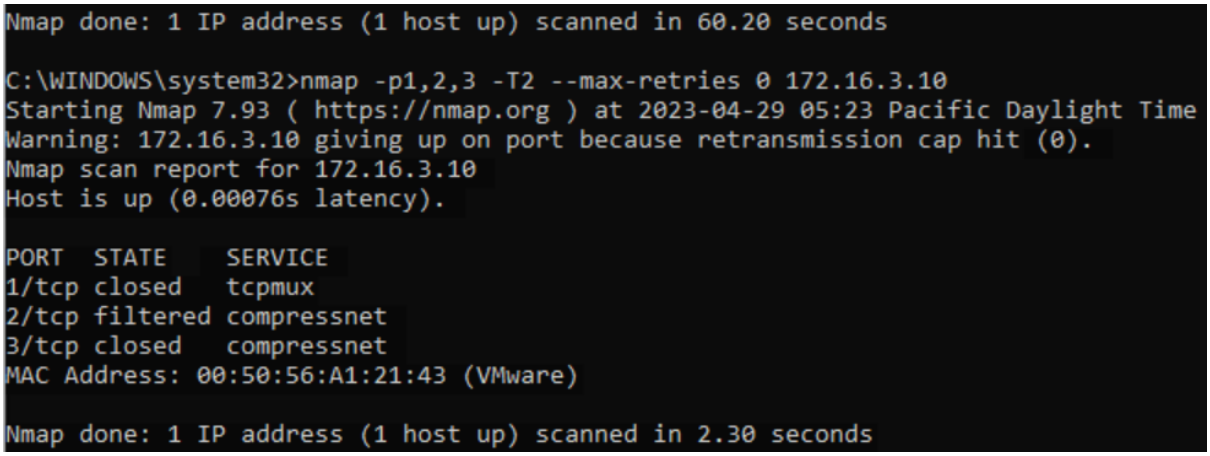
  Target:

No.	Time	Source	Destination	Protocol	Length	Info
39	38.611353	172.16.3.11	172.16.3.10	TCP	60	38384 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40	38.611418	172.16.3.10	172.16.3.11	TCP	54	2 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
55	53.619370	172.16.3.11	172.16.3.10	TCP	60	38384 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
56	53.619433	172.16.3.10	172.16.3.11	TCP	54	3 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
78	68.633460	172.16.3.11	172.16.3.10	TCP	60	38384 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
79	68.633531	172.16.3.10	172.16.3.11	TCP	54	1 → 38384 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

- We did not get an alert even though the attacker was scanning the target.
- This is because we used lower timing template available in Nmap “T1”. This allows the attacker to delay the generation of TCP packet for the attack.
- The delay between every TCP packet is about 15 seconds and our port scan detector logic was configured to detect attack within 6 seconds and thus attacker evaded the IPS.

- If we use the same configurations on our port scan detection alert system and scan the target with timing template T2, then we will get an alert.
- This is because T2 scans the target with initial delay of 400ms and for TCP 1s.

  Attacker:



No.	Time	Source	Destination	Protocol	Length	Info
26	16.431839	172.16.3.11	172.16.3.10	TCP	58	45364 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27	16.432201	172.16.3.10	172.16.3.11	TCP	60	1 → 45364 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
28	16.838047	172.16.3.11	172.16.3.10	TCP	58	45364 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	16.838391	172.16.3.10	172.16.3.11	TCP	60	3 → 45364 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
30	17.244337	172.16.3.11	172.16.3.10	TCP	58	45364 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

  Target:

No.	Time	Source	Destination	Protocol	Length	Info
13	9.430931	172.16.3.11	172.16.3.10	TCP	60	45364 → 1 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	9.431043	172.16.3.10	172.16.3.11	TCP	54	1 → 45364 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15	9.837150	172.16.3.11	172.16.3.10	TCP	60	45364 → 3 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
16	9.837218	172.16.3.10	172.16.3.11	TCP	54	3 → 45364 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
17	10.243438	172.16.3.11	172.16.3.10	TCP	60	45364 → 2 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Anomalous Packet

- A packet which does not follow TCP/IP standards. For example, A packet which does not have any flags set, then it has to be a malicious packet.
- These malicious packets do not trigger any alert from current version of port scan detector.

- Target is configured to generate an alert if more than 2 ports are scanned within 6 seconds by same IP.
  Attacker:

```
C:\Users\Admin>nmap -p12 -sN --max-retries 0 -T5 172.16.3.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-03 03:31 Pacific Daylight Time
Nmap scan report for 172.16.3.10
Host is up (0.00013s latency).

PORT   STATE  SERVICE
12/tcp closed unknown
MAC Address: 00:50:56:A1:21:43 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

No.	Time	Source	Destination	Protocol	Length	Info
73	29.906950	172.16.3.11	172.16.3.10	TCP	54	36849 → 12 [<None>] Seq=1 Win=1024 Len=0
74	29.907159	172.16.3.10	172.16.3.11	TCP	60	12 → 36849 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

  Target:

No.	Time	Delta	Source	Destination	Protocol	Length	Info
40	5.216663	0.049742	172.16.3.11	172.16.3.10	TCP	60	36849 → 12 [<None>] Seq=1 Win=1024 Len=0
41	5.216709	0.000046	172.16.3.10	172.16.3.11	TCP	54	12 → 36849 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

- No alert is generated. Even though we are scanning by sending 1 packet, it should generate an alert because NULL scan don't have any flag set which indicate it is a scan attempt.

Multiple IP's

- If a vulnerability gets publicized about a particular service running on a specific port, then attacker will only attack that port. To find out the information about vulnerable port, attacker targets every system on the network and scans only for vulnerable port.

- Attacker is targeting multiple IP’s and scanning only a single port. This type of port scan attack will go undetected by current version.

Distributed Port Scanning

- An attacker can use multiple computers in her access to do port scanning. These computers will be able to communicate with each other and every computer will perform port scan attack in coordinated manner. One computer will only scan a handful of ports and other will scan other handful ports.
- This makes our detection system confused in known if someone is performing a port scan attack because each PC has a different IP address.

Slow Port Scanning Detection

- Our proposed method is mainly composed of two phases:
  1. A feature collection phase that analyzes network traffic and extracts the features needed to classify a certain IP as malicious or not.

  2. A classification phase that divides the IPs, based on the collected features, into three groups: normal IPs, suspicious IPs and scanner IPs.
     - The IPs our approach classify as suspicious are kept for the next (K) time windows for further examination to decide whether they represent scanners or legitimate users.
     - Hence, this approach is different than the traditional approach used by IDSs that classifies IPs as either legitimate or scanners, and thus producing a high number of false positives and false negatives.

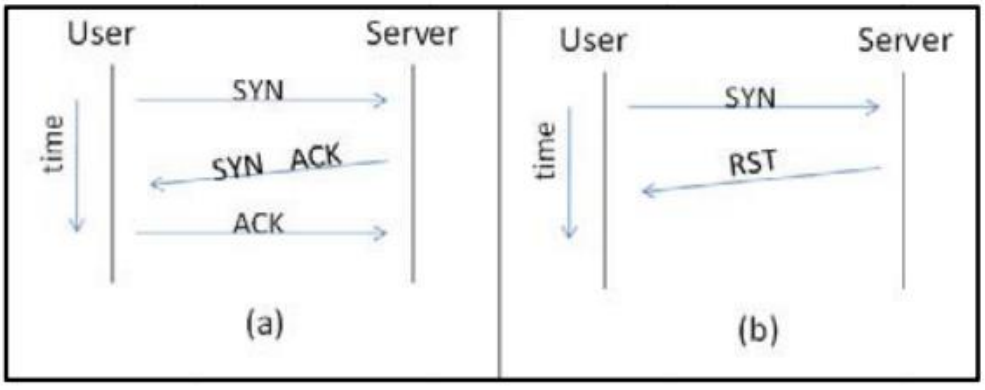
Proposed Method

- To detect slow port scans, we need a large time window so that we can detect any abnormal pattern in the behaviour of the clients. However, choosing a large time window (one hour for example) will require a large memory for storing the traffic and will cause a large delay at the server side as there will be a large amount of traffic that needs to be checked for many IPs which will lead to performance degradation.
- Our IDS will collect different features for each IP address in small time window of T minutes.

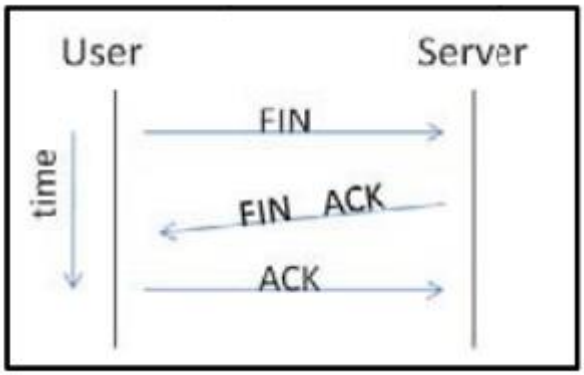
- The IP addresses are then classified based on the collected features that reflect their behaviour, into 3 groups: normal IPs, suspicious IPs, and scanner IPs.
- To allow the detection of any abnormal behaviour in a time that is larger than the time window (T), we keep the suspicious IP addresses in a list for the following (K) time windows and we call that list the suspicious IP list. This allows IDS to watch the behaviour of these IPs in a larger time period.
- If any IP that is a member of the suspicious IPs list is again classified as suspicious IP, the module will notify the administrator or firewall about this abnormal behaviour.

TCP Connection

- A TCP connection is established by a three-way handshake. A client who wants to connect to a specific port sends a TCP segment with the SYN bit in the header of the segment set on.
- The server on the other side receives this segment and checks if there is an available service on that port, then it responds by a TCP segment with the SYN and ACK bit set on.
- Finally, the client completes the three-way handshake and replies by an acknowledgement. Otherwise, if no service is available for that port, the server responds to the received SYN segment by a segment with the RST bit set on.



- To terminate an established TCP connection, the client sends a TCP segment with the FIN bit set on.
- The server responds by sending another TCP segment with the FIN and ACK bits set on. Finally the client acknowledges the reception of that segment by sending a segment with the ACK bit set on.



Common TCP Scanning

- The three most common TCP scanning techniques that an attacker may follow in order to know which ports are open and which ports are closed are: connect scan, half connect scan and FIN scan.

1. Connect Scan:
   - The scanner sends a SYN, and determines whether the port is open or closed based on the server's response. If the server replies by a SYN-ACK then the port is open and the scanner continues the three-way hand shake and establishes a connection by sending an ACK to the server. Otherwise, the server replies by a RST to declare a closed port.

- The first feature that distinguishes legitimate users from scanners is the number of connections made to closed ports (N_{closed}).
- Legitimate users are aware of the offered services and are not expected to try to establish connections to closed ports. On the contrary, there is a great chance that an attacker while scanning the ports will try, unknowingly, to connect to closed ports.

- For each IP, the number of connections attempted to closed ports (N_{closed}) is determined by calculating the difference between the incoming SYN from each IP and the outgoing SYN ACK from the server to each IP.

$$N_{closed}^{ip} = SYN_{in,ip} - SYN_{out,ip}$$

- Where, $SYN_{in,ip}$ is the number of incoming SYN segments from IP to server and $SYN_{out,ip}$ is the number of outgoing SYN-ACKS that are sent from server to IP.

- If no connections are made to closed ports, then the difference is equal to zero. This is based on the fact that when the server receives a connection establishment to a closed port, it responds by an RST indicating that the destination port is closed. Whereas the server responds by a SYN ACK segment if the port is open.

2. Half Connect Scan:

- This technique is also known by the SYN scan. It is similar to the connect scan except that after receiving a reply from the server that the port is open, the scanner doesn't

continue the three-way hand shake and doesn't establish a full connection.

- This type of scan is usually more difficult to detect, since establishing the connection is not completed and isn't logged.
- The number of half connections N_{hc}^{ip} that are made by an IP can be determined directly by calculating the number of SYN-ACKs that were sent from the server to IP and that weren't followed by an ACK.

3. FIN Scan:

- FIN bit is set when the two parties want to terminate a pre-established connection.
- A scanner determines whether a port is open or closed by sending a TCP segment with the FIN bit set to a port. Since no connection to that port was established in the first place, the server responds by RST if the port is closed or the server doesn't respond at all if the port is open. Based on whether the scanner receives an RST or not, he can determine whether the port is closed or open.

- The number of FIN probes that were not preceded by establishing a connection (N_{Fin}) can be calculated for each IP by calculating the difference between the incoming FINs from the IP to the server and the outgoing FINs from the server to the IP.
- The number of FIN probes that are made by an IP and that weren't preceded by establishing a connection can be determined as:

$$N_{Fin}^{ip} = FIN_{in,ip} - FIN_{out,ip}$$

- $FIN_{in,ip}$ is the number of incoming FINs that are sent from IP to the server, and $FIN_{out,ip}$ is the number of outgoing FINs from the server to IP.
- The difference should be zero in normal cases since the connection is terminated by exchanging two FIN segments (one coming from the host and one coming from the server).

Methodology

- We calculate these features (N_{closed}, N_{hc}, N_{Fin}) for each IP at the end of each time window. Then we decide whether each IP represents a scanner or a legitimate user based on these features. The rules for classifying an IP based on the values of its features are shown as follows:

$$state(ip) = \begin{cases} legitimate, N_{closed}^{ip} = 0 \; and \; N_{hc}^{ip} = 0 \; and \; N_{FIN}^{ip} = 0 \\ suspicious, N_{closed}^{ip} = 1 \; or \; N_{hc}^{ip} = 1 \; or \; N_{FIN}^{ip} = 1 \\ scanner, N_{closed}^{ip} > 1 \; or \; N_{hc}^{ip} > 1 \; or \; N_{FIN}^{ip} > 1 \end{cases}$$

- If all the collected features are equal to zero, then the IP is classified as a legitimate user since all the IP's activity in that time window is normal.

- If N_{closed}^{ip} or N_{hc}^{ip} or N_{FIN}^{ip} is equal to one, then we can't decide whether IP is a legitimate user or a scanner.
- If N_{closed}^{ip} is equal to one, then it could be a legitimate user who made by mistake a connection to a closed port or a scanner who is performing a stealthy slow scan.

- Also if N_{hc}^{ip} equals one, then it is possible that the legitimate user tried to connect to an open port by sending a SYN segment, but the legitimate user's machine was shutdown, for some reason, before completing the three-way hand shake causing a half-open connection.
- A normal case where N_{FIN}^{ip} equals one is when the legitimate user wants to terminate an established connection by sending a FIN segment to the server. But in case of network congestions, the server won't reply by a FIN segment directly. The legitimate user will send another FIN segment assuming that the first one was lost. Once the first FIN segment is received by the server, it replies with a FIN segment and terminates the connection, whereas the server ignores the second received FIN segment and the difference between the incoming FINs and outgoing FINs will equal one.
- All of the above mentioned cases occur rarely, but should be taken into account in order to reduce false positive alarms.

- So what we do is add this IP to the suspicious list to further examine its behavior. If the IP isn't classified as suspicious in the following (K) time windows, it is removed from the suspicious list. Otherwise, an alarm is sent to the administrator since this continuous suspicious behavior represents a slow port scanning.
- The suspicious list is a table that contains IP addresses; the table has a small size and is the only thing that is saved after the time window expires.
- If N_{closed}^{ip} or N_{hc}^{ip} or N_{FIN}^{ip} is larger than one, then is classified as a scanner and an alarm is sent to the administrator or a

message is sent directly to the firewall to disable incoming connections from that IP.

- The duration of the time window (T) and the number of consecutive time windows (K) after which the suspicious IP is removed from the suspicious list are specified by the user.
- If T = 3 minutes and K = 10, then the only way to launch an undetected slow port scan will be to scan one port every (T×K= 30 minutes). This is due to the fact that our detection method will classify the IP as suspicious, but after 10 consecutive time windows it will be removed from the suspicious list since it might be a legitimate user who incorrectly made a connection to a closed port. However, if the scanner is willing to spend 30 minutes to scan each port, this requires a very long time for scanning all the 65535 TCP ports to know the available services.