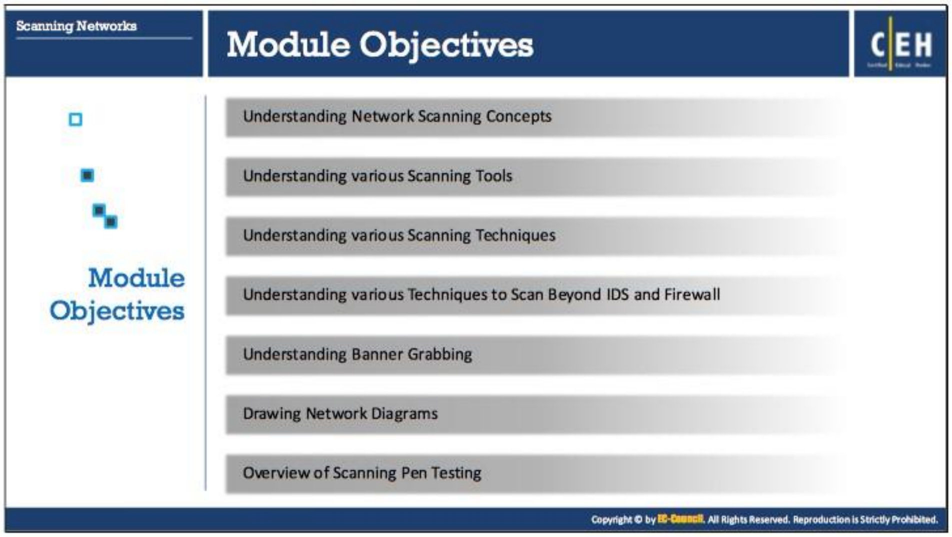
Certified Ethical Hacker

Module 03

Scanning Networks

This page is intentionally left blank.



Scanning Networks

Module Objectives

Module Objectives

- Understanding Network Scanning Concepts
- Understanding various Scanning Tools
- Understanding various Scanning Techniques
- Understanding various Techniques to Scan Beyond IDS and Firewall
- Understanding Banner Grabbing
- Drawing Network Diagrams
- Overview of Scanning Pen Testing

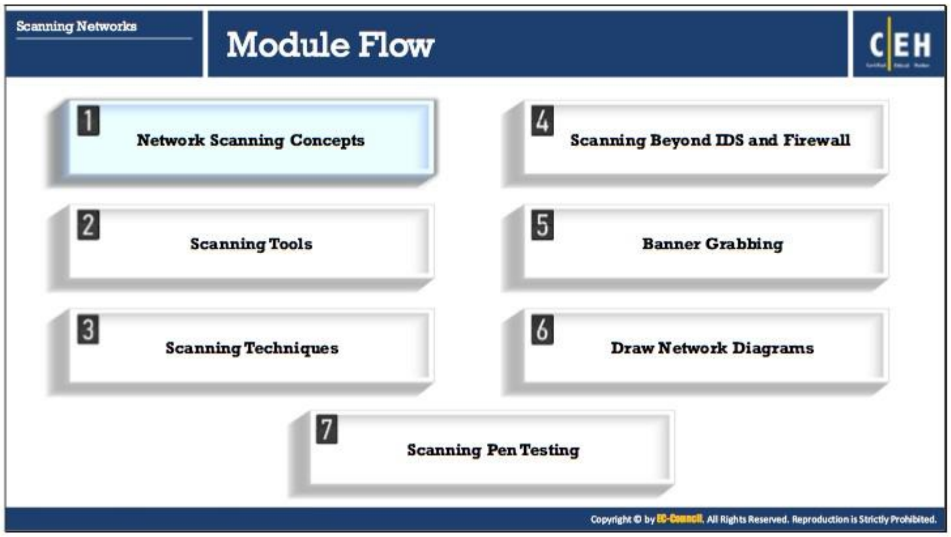
Module Objectives

After identifying the target system and performing the initial reconnaissance as discussed in the Footprinting and Reconnaissance module, attackers begin to search for an entry point into the target system. It should be noted that the scanning itself is not the actual intrusion, but an extended form of reconnaissance in which the attacker learns more about his/her target, including information about operating systems, services, and any configuration lapses. The information gleaned from this reconnaissance helps the attacker select strategies for the attack on the target system or network.

This module starts with an overview of network scanning and provides an insight into various techniques that can be used to check for live systems and open ports. It goes on to discuss various scanning techniques and ends with an overview of penetration testing (also called pen-testing) steps that an ethical hacker should follow to perform the security assessment of the target.

At the end of this module, you will be able to:

- Describe the network scanning concepts
- Use various scanning tools
- Perform scanning to check for live systems and open ports
- Perform scanning by using various scanning techniques
- Scan beyond intrusion detection system (IDS) and firewall
- Perform banner grabbing
- Draw network diagrams using network discovery tools
- Perform scanning penetration testing



Scanning Networks

Module Flow

1 Network Scanning Concepts

2 Scanning Tools

3 Scanning Techniques

4 Scanning Beyond IDS and Firewall

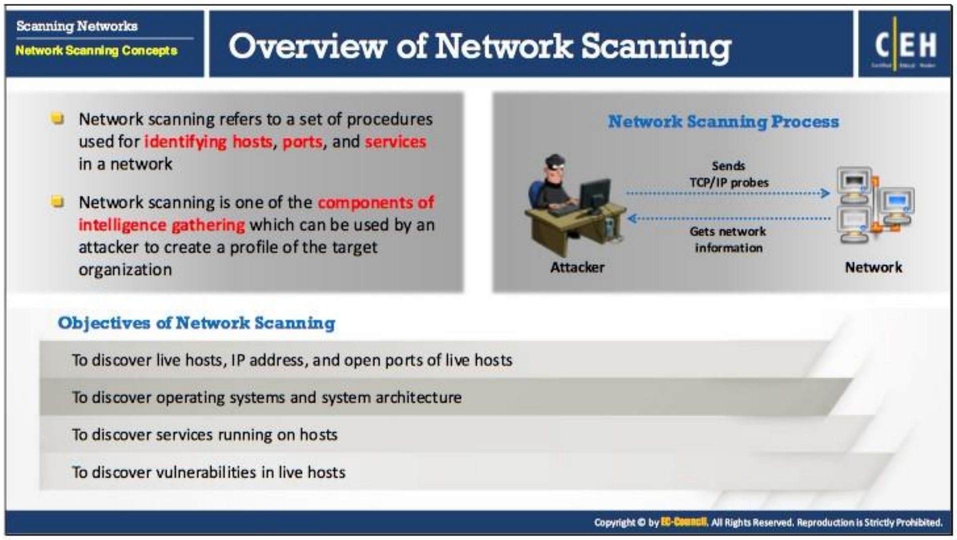
5 Banner Grabbing

6 Draw Network Diagrams

7 Scanning Pen Testing

Network Scanning Concepts

As already discussed, footprinting is the first phase of hacking, in which the attacker gains primary information about a potential target. He/she then uses this information in the scanning phase in order to gather more detailed information about the target.



Scanning Networks
Network Scanning Concepts

Overview of Network Scanning

- Network scanning refers to a set of procedures used for **identifying hosts**, **ports**, and **services** in a network
- Network scanning is one of the **components of intelligence gathering** which can be used by an attacker to create a profile of the target organization

Network Scanning Process
Sends TCP/IP probes
Gets network information
Attacker
Network

Objectives of Network Scanning

- To discover live hosts, IP address, and open ports of live hosts
- To discover operating systems and system architecture
- To discover services running on hosts
- To discover vulnerabilities in live hosts

Overview of Network Scanning

Scanning is the process of gathering additional detailed information about the target by using highly complex and aggressive reconnaissance techniques. Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network. It is one of the most important phases of intelligence gathering for an attacker which enables him/her to create a profile of the target organization. In the process of scanning, the attacker tries to gather information, including the specific IP addresses that can be accessed over the network, the target's operating systems and system architecture, and the services running on each computer.

The purpose of scanning is to discover exploitable communications channels, probe as many listeners as possible, and keep track of the ones that are responsive or useful to an attacker's particular needs. In the scanning phase of an attack, the attacker tries to find various ways to intrude into a target system. The attacker also tries to discover more about the target system to find out if there are any configuration lapses in it. The attacker then uses the information obtained during the scan to develop an attack strategy.

Types of Scanning

- **Port Scanning** – Lists the open ports and services. Port scanning is the process of checking the services running on the target computer by sending a sequence of messages in an attempt to break in. Port scanning involves connecting to or probing TCP and UDP ports on the target system to determine if the services are running or are in a listening state. The listening state provides information about the operating system and the application currently in use. Sometimes, active services that are listening may allow unauthorized user access to misconfigure systems or to run software with vulnerabilities.
- **Network Scanning** – Lists IP addresses. Network scanning is a procedure for identifying active hosts on a network, either to attack them or to assess the security of the network.

- **Vulnerability Scanning** – Shows the presence of known weaknesses. Vulnerability scanning is a method used to check whether a system is exploitable by identifying its vulnerabilities. A vulnerability scanner consists of a scanning engine and a catalog. The catalog includes a list of common files with known vulnerabilities and common exploits for a range of servers. A vulnerability scanner may, for example, look for backup files or directory traversal exploits. The scanning engine maintains logic for reading the exploit list, transferring the request to the Web server, and analyzing the requests to ensure the safety of the server. These tools generally target vulnerabilities that secure host configurations can fix easily, updated security patches, and a clean Web document.

The access points that a thief who wants to break into a house looks for are the doors and windows. These are usually the house's points of vulnerability, as they are easily accessible. When it comes to computer systems and networks, ports are the doors and windows of the system that an intruder uses to gain access. A general rule for computer systems is that more the number of open ports on a system, more vulnerable is the system. However, there are cases, in which a system has fewer open ports compared to another machine, but the open ports present a much higher level of vulnerability.

Objectives of Network Scanning

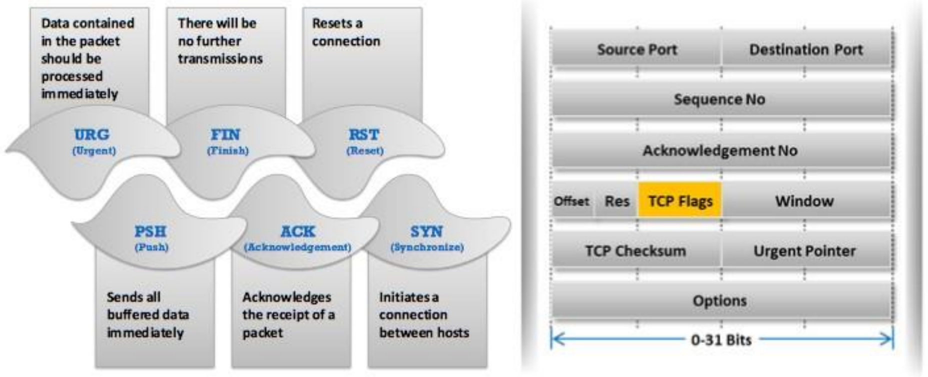
The more the information at hand about a target organization, the greater the chances of knowing a network's security loopholes and consequently, for gaining unauthorized access to it.

Below are some objectives for scanning a network:

- Discover the network's live hosts, IP addresses, and open ports of live. Using open ports, the attacker will determine the best means of entry into the system.
- Discover the operating system and system architecture of the target. This is also known as fingerprinting. An attacker can formulate an attack strategy based on the operating system's vulnerabilities.
- Discover the services running/listening on the target system. Doing so gives the attacker an indication of vulnerabilities (based on the service) exploitation for gaining access to the target system.
- Identify specific applications or versions of a particular service.
- Identify vulnerabilities in any of the network systems. This helps an attacker to compromise the target system or network through various exploits.

Scanning Networks
Network Scanning Concepts

TCP Communication Flags


Data contained in the packet should be processed immediately
URG (Urgent)
There will be no further transmissions
FIN (Finish)
Resets a connection
RST (Reset)
PSH (Push)
Sends all buffered data immediately
ACK (Acknowledgement)
Acknowledges the receipt of a packet
SYN (Synchronize)
Initiates a connection between hosts

Source Port			Destination Port
Sequence No			
Acknowledgement No			
Offset	Res	TCP Flags	Window
TCP Checksum			Urgent Pointer
Options			
0-31 Bits			

Standard TCP communications are controlled by flags in the TCP packet header

TCP Communication Flags

TCP header contains various flags that control the transmission of data across a TCP connection. Six TCP control flags manage the connection between hosts and give instructions to the system. Four of these flags (namely: SYN, ACK, FIN, and RST) govern the establishment, maintenance, and termination of a connection. The other two flags (namely: PSH and URG) provide instructions to the system. The size of each flag is 1 bit. As there are six flags in the TCP Flags section, the size of this section is 6 bits. When a flag value is set to "1,"that flag is automatically turned on.

The following are the TCP communication flags:

- Synchronize alias "SYN": It notifies the transmission of a new sequence number. This flag generally represents the establishment of a connection (3-way handshake) between two hosts.
- Acknowledgement alias "ACK": It confirms the receipt of transmission and identifies next expected sequence number. When the system successfully receives a packet, it sets the value of its flag to "1," implying that the receiver should pay attention to it.
- Push alias "PSH": When its flag is set to "1," it indicates that the sender has raised the push operation to the receiver; this implies that the remote system should inform the receiving application about the buffered data coming from the sender. The system raises the PSH flag at the time of start and end of data transfer and sets it on the last segment of a file to prevent buffer deadlocks.
- Urgent alias "URG": It instructs the system to process the data contained in packets as soon as possible. When the system sets the flag to "1," the remote system gives priority to the urgent data and processes it first, stopping all the other data processing.

- Finish alias "FIN": Its flag is set to "1" to announce that it will not send more transmissions to the remote system and terminates the connection established by the SYN flag.
- Reset alias "RST": When there is an error in the current connection, its flag is set to "1," and it aborts the connection in response to the error. Attackers make use of this to scan hosts in search of open ports.

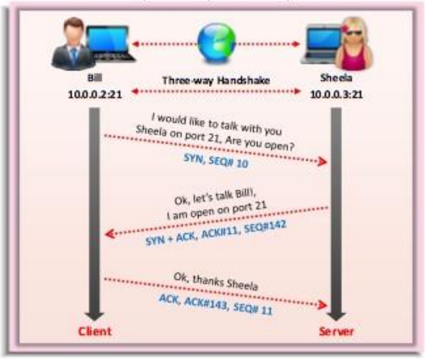
SYN scanning mainly deals with three of the flags: SYN, ACK, and RST. You can use these three flags for gathering illegal information from servers during the enumeration process.

Scanning Networks
Network Scanning Concepts

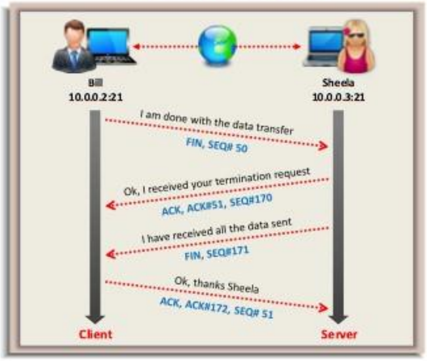
TCP/IP Communication

CEH

TCP Session Establishment
(Three-way Handshake)



TCP Session Termination



TCP/IP Communication

TCP is connection-oriented, which prioritizes connection establishment before data transfer between applications. This connection between protocols is possible through the process of the three-way handshake.

TCP Session initiates using a three-way handshake mechanism:

- To launch a TCP connection, the source (10.0.0.2:21) sends a SYN packet to the destination (10.0.0.3:21).
- On receiving the SYN packet, the destination responds by sending a SYN/ACK packet back to the source.
- The ACK packet confirms the arrival of the first SYN packet to the source.
- To conclude, the source sends an ACK packet for the ACK/SYN packet transmitted by the destination.
- This triggers an "OPEN" connection, allowing communication between the source and the destination; this continues until one of them issues a "FIN" or "RST" packet to close the connection.

The TCP protocol maintains stateful connections for all connection-oriented protocols throughout the Internet, and works like an ordinary telephone communication, in which one picks up a telephone receiver, hears a dial tone, and dials a number that triggers ringing at the other end, until a person picks up the receiver and says, "Hello."

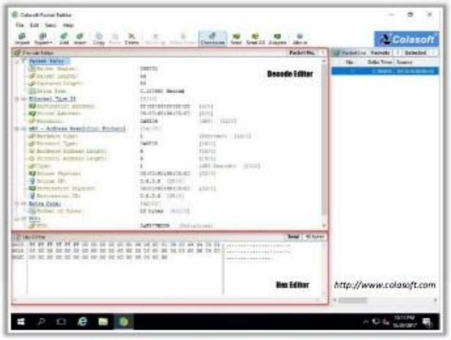
The system terminates the established TCP Session as follows:

After completing all the data transfers through the established TCP connection, the sender sends the connection termination request to the receiver by sending a FIN or RST packet. Upon receiving the connection termination request, the receiver acknowledges the termination requests by sending ACK packet to the sender and finally sends its own FIN packet; then the system will terminate the established connection.

Scanning Networks
Network Scanning Concepts

Creating Custom Packet Using TCP Flags

- Colasoft Packet Builder enables the creation of custom network packets to **audit networks for various attacks**
- Attackers can also use it to create fragmented packets to **bypass firewalls and IDS systems** in a network



http://www.colasoft.com

Packet Crafting Tools

- NetScanTools Pro (*https://www.netscantools.com*)
- Ostinato (*http://ostinato.org*)
- WAN Killer (*http://www.solarwinds.com*)
- Packeth (*http://packeth.sourceforge.net*)
- LANforge FIRE (*http://www.candelatech.com*)

Creating Custom Packet Using TCP Flags

Packet Crafting Tools

Packet crafting tools are used to generate and analyze network traffic. These tools craft and send packet streams by using different protocols at different transfer rates.

- **Colasoft Packet Builder**

  Source: *http://www.colasoft.com*

  Colasoft Packet Builder is a tool that allows an attacker to create custom network packets and helps security professionals to assess the network. The attacker can select a TCP packet from the provided templates and change the parameters in the decoder editor, hexadecimal editor, or ASCII editor to create a packet. In addition to building packets, Colasoft Packet Builder supports saving packets to packet files and sending packets to the network.

  There are three views in the Packet Builder: Packet List, Decode Editor, and Hex Editor.

  - The Packet List displays all constructed packets. When you select one or more packets in the Packet List, the first highlighted packet displays in both Decode Editor and Hex Editor for editing.
  - In the Hex Editor, the data of the packet are represented as hexadecimal values and ASCII characters; nonprintable characters are represented by a dot (".") in the ASCII section. You can edit either the hexadecimal values or the ASCII characters.
  - Decode editor allows the attacker to edit packets without remembering value length, byte order, and offsets. You can select a field and change value in the edit box.

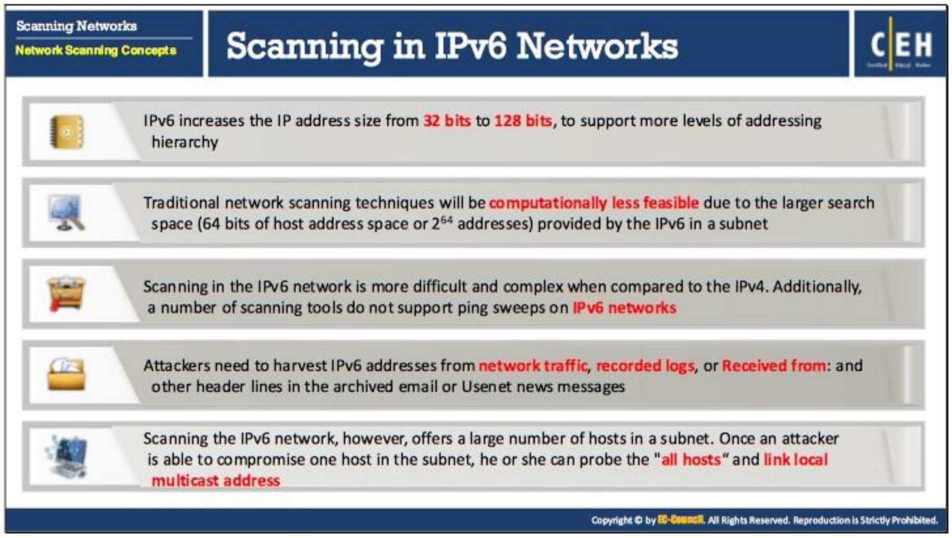
For creating a packet, you can use the add or insert packet command in the Edit menu or on the Toolbar to create a new packet.

The attacker can send a constructed packet to wire directly and control how Colasoft Packet Builder sends the packets, specifying, for example, the interval between every packet, loop times, and the delay time between loops.

This packet builder audits networks and checks network protection against attacks and intruders. Attackers may use this packet builder to create fragmented packets to bypass network firewalls and IDS systems. They can also create packets and flood the victim with a very large number of packets, which could result in denial-of-service attacks.

Some of the packet crafting tools include:

- NetScanTools Pro (*https://www.netscantools.com*)
- Ostinato (*http://ostinato.org*)
- WAN Killer (*http://www.solarwinds.com*)
- Packeth (*http://packeth.sourceforge.net*)
- LANforge FIRE (*http://www.candelatech.com*)
- Bit-Twist (*http://bittwist.sourceforge.net*)
- WireEdit (*https://wireedit.com*)



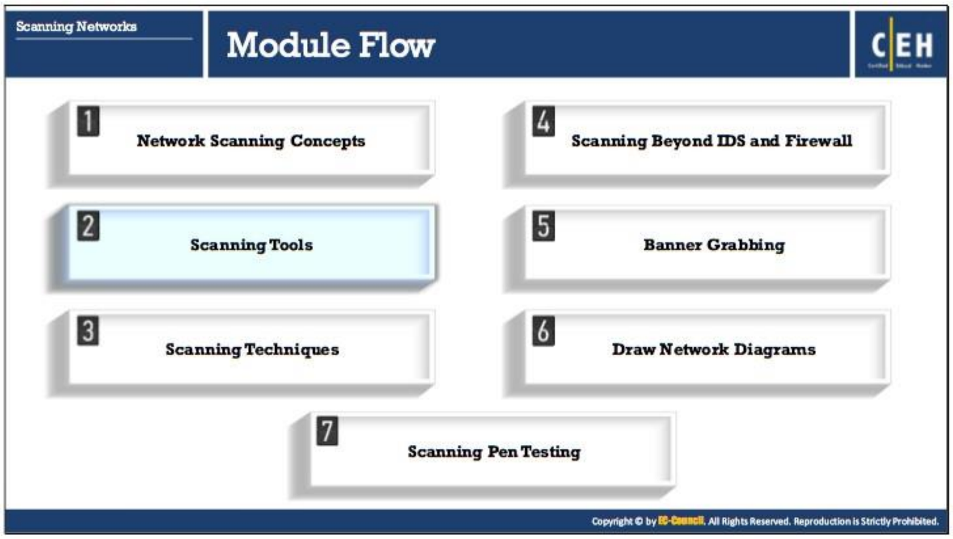
Scanning Networks
Network Scanning Concepts
CEH

Scanning in IPv6 Networks

- IPv6 increases the IP address size from **32 bits** to **128 bits**, to support more levels of addressing hierarchy
- Traditional network scanning techniques will be **computationally less feasible** due to the larger search space (64 bits of host address space or 2^{64} addresses) provided by the IPv6 in a subnet
- Scanning in the IPv6 network is more difficult and complex when compared to the IPv4. Additionally, a number of scanning tools do not support ping sweeps on **IPv6 networks**
- Attackers need to harvest IPv6 addresses from **network traffic**, **recorded logs**, or **Received from**: and other header lines in the archived email or Usenet news messages
- Scanning the IPv6 network, however, offers a large number of hosts in a subnet. Once an attacker is able to compromise one host in the subnet, he or she can probe the "**all hosts**" and **link local multicast address**

Scanning in IPv6 Networks

IPv6 increases the size of IP address space from 32 bits to 128 bits to support more levels of addressing hierarchy. Traditional network scanning techniques are computationally less feasible because of larger search space (64 bits of host address space, or 2^{64} addresses) provided by IPv6 in a subnet. Scanning in the IPv6 network is more difficult and complex when compared to the IPv4.Additionally, a number of scanning tools do not support ping sweeps on IPv6 networks. Attackers need to harvest IPv6 addresses from network traffic, recorded logs, or "Received from" and other header lines in archived email or Usenet news messages to identify IPv6 addresses for subsequent port scanning. Scanning an IPv6 network, however, offers a large number of hosts in a subnet; if an attacker can compromise one subnet host, he can probe the "all hosts" link local multicast address, if hosts numbers are sequential, or use any regular scheme. An attacker needs to analyze 2^{64} addresses to verify if a particular open service is running on a host in that subnet. At a conservative rate of one probe per second, such a scan would take about 5 billion years to complete.



Scanning Networks

Module Flow

1 Network Scanning Concepts

2 Scanning Tools

3 Scanning Techniques

4 Scanning Beyond IDS and Firewall

5 Banner Grabbing

6 Draw Network Diagrams

7 Scanning Pen Testing

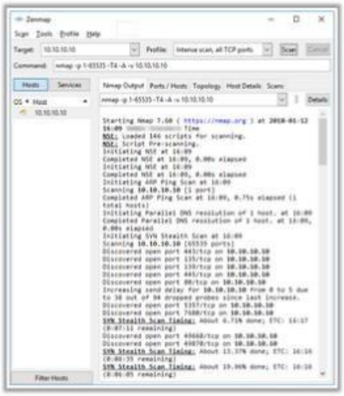
Scanning Tools

Scanning tools scan and identify live hosts, open ports, running services on a target network, location-info, NetBIOS info and information about all TCP/IP, UDP open ports. Information obtained from these tools will assist an ethical hacker in creating the profile of the target organization and to scan the network for open ports of the devices connected.

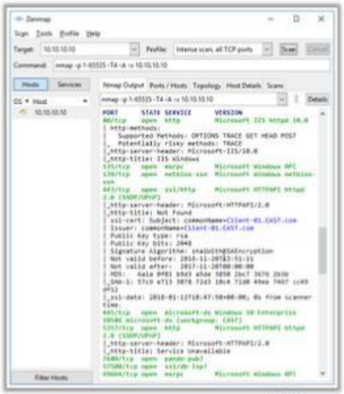
Scanning Networks
Scanning Tools

Scanning Tool: Nmap

- Network administrators can use Nmap for **network inventory**, managing service upgrade schedules, and monitoring host or service uptime
- Attacker uses Nmap to extract information such as **live hosts on the network**, **services** (application name and version), **type of packet filters/firewalls**, **operating systems, and OS versions**



```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12
16:09 [illegible] Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 16:09
Completed NSE at 16:09, 0.00s elapsed
Initiating NSE at 16:09
Completed NSE at 16:09, 0.00s elapsed
Initiating ARP Ping Scan at 16:09
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 16:09, 0.75s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:09
Completed Parallel DNS resolution of 1 host. at 16:09,
0.00s elapsed
Initiating SYN Stealth Scan at 16:09
Scanning 10.10.10.10 [65535 ports]
Discovered open port 443/tcp on 10.10.10.10
Discovered open port 135/tcp on 10.10.10.10
Discovered open port 139/tcp on 10.10.10.10
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 80/tcp on 10.10.10.10
Increasing send delay for 10.10.10.10 from 0 to 5 due
to 38 out of 94 dropped probes since last increase.
Discovered open port 5357/tcp on 10.10.10.10
Discovered open port 7680/tcp on 10.10.10.10
SYN Stealth Scan Timing: About 6.71% done; ETC: 16:17
(0:07:11 remaining)
Discovered open port 49668/tcp on 10.10.10.10
Discovered open port 49670/tcp on 10.10.10.10
SYN Stealth Scan Timing: About 13.37% done; ETC: 16:16
(0:06:35 remaining)
SYN Stealth Scan Timing: About 19.96% done; ETC: 16:16
(0:06:05 remaining)
```



```
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-
ssn
443/tcp   open  ssl/http     Microsoft HTTPAPI httpd
2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
| ssl-cert: Subject: commonName=Client-01.CAST.com
| Issuer: commonName=Client-01.CAST.com
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2016-11-20T13:51:11
| Not valid after:  2017-11-20T00:00:00
| MD5:   6a1a 0f01 b9d3 a5de 5050 2bc7 3670 2b30
|_SHA-1: 57c9 e713 3078 72d3 10c4 71d0 49ee 7407 cc49
e912
|_ssl-date: 2018-01-12T10:47:50+00:00; 0s from scanner
time.
445/tcp   open  microsoft-ds Windows 10 Enterprise
10586 microsoft-ds (workgroup: CAST)
5357/tcp  open  http         Microsoft HTTPAPI httpd
2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
7680/tcp  open  pando-pub?
17500/tcp open  ssl/db-lsp?
49664/tcp open  msrpc        Microsoft Windows RPC
```

https://nmap.org

- **Nmap**

  Source: *https://nmap.org*

  Nmap is a security scanner for network exploration and hacking. It allows you to discover hosts and services on a computer network, thus creating a "map" of the network. It sends specially crafted packets to the target host and then analyzes the responses to accomplish its goal. Either a network administrator or an attacker can use this tool for their specific needs. Network administrators can use Nmap for network inventory, managing service upgrade schedules, and monitoring host or service uptime. Attackers use Nmap to extract information such as live hosts on the network, services (application name and version), type of packet filters/firewalls, operating systems, and OS versions. Nmap includes a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

  Some of the features of Nmap are:

  - It scans vast networks of literally hundreds of thousands of machines.
  - It supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. Nmap includes many port scanning mechanisms (TCP and UDP), OS detection, version detection, ping sweeps, and so on.

Scanning Networks
Scanning Tools

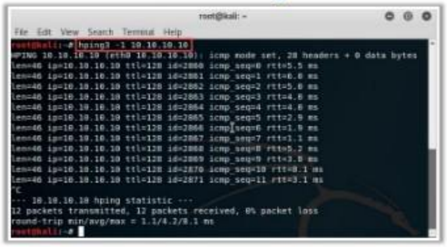
Scanning Tool: Hping2 / Hping3

1 Command line **network scanning** and **packet crafting** tool for the TCP/IP protocol

2 It can be used for **network security auditing**, **firewall testing**, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, etc.

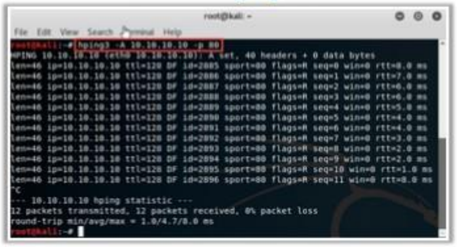
http://www.hping.org

ICMP Scanning



```
root@kali:~# hping3 -1 10.10.10.10
HPING 10.10.10.10 (eth0 10.10.10.10): icmp mode set, 28 headers + 0 data bytes
len=46 ip=10.10.10.10 ttl=128 id=2860 icmp_seq=0 rtt=5.5 ms
len=46 ip=10.10.10.10 ttl=128 id=2861 icmp_seq=1 rtt=6.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2862 icmp_seq=2 rtt=5.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2863 icmp_seq=3 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2864 icmp_seq=4 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2865 icmp_seq=5 rtt=2.9 ms
len=46 ip=10.10.10.10 ttl=128 id=2866 icmp_seq=6 rtt=1.9 ms
len=46 ip=10.10.10.10 ttl=128 id=2867 icmp_seq=7 rtt=1.1 ms
len=46 ip=10.10.10.10 ttl=128 id=2868 icmp_seq=8 rtt=5.2 ms
len=46 ip=10.10.10.10 ttl=128 id=2869 icmp_seq=9 rtt=3.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2870 icmp_seq=10 rtt=8.1 ms
len=46 ip=10.10.10.10 ttl=128 id=2871 icmp_seq=11 rtt=3.1 ms
^C
--- 10.10.10.10 hping statistic ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 1.1/4.2/8.1 ms
root@kali:~#
```

ACK Scanning on port 80



```
root@kali:~# hping3 -A 10.10.10.10 -p 80
HPING 10.10.10.10 (eth0 10.10.10.10): A set, 40 headers + 0 data bytes
len=46 ip=10.10.10.10 ttl=128 DF id=2885 sport=80 flags=R seq=0 win=0 rtt=8.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2886 sport=80 flags=R seq=1 win=0 rtt=7.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2887 sport=80 flags=R seq=2 win=0 rtt=6.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2888 sport=80 flags=R seq=3 win=0 rtt=6.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2889 sport=80 flags=R seq=4 win=0 rtt=5.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2890 sport=80 flags=R seq=5 win=0 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2891 sport=80 flags=R seq=6 win=0 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2892 sport=80 flags=R seq=7 win=0 rtt=3.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2893 sport=80 flags=R seq=8 win=0 rtt=2.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2894 sport=80 flags=R seq=9 win=0 rtt=2.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2895 sport=80 flags=R seq=10 win=0 rtt=1.0 ms
len=46 ip=10.10.10.10 ttl=128 DF id=2896 sport=80 flags=R seq=11 win=0 rtt=8.0 ms
^C
--- 10.10.10.10 hping statistic ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 1.0/4.7/8.0 ms
root@kali:~#
```

- **Hping2 / Hping3**

  Source: *http://www.hping.org*

  Hping2/Hping3 is a command-line-oriented network scanning and packet crafting tool for the TCP/IP protocol that sends ICMP echo requests and supports TCP, UDP, ICMP, and raw-IP protocols. It performs network security auditing, firewall testing, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, and other functions. Hping2/Hping3 has a Traceroute mode which enables you to send files between covert channels. It can send custom TCP/IP packets and display target replies, as does a ping program with ICMP replies. It handles fragmentation, arbitrary packets' body and size, and uses them to transfer encapsulated files under supported protocols. It supports idle host scanning. IP-spoofing and network/host scanning can be used to perform an anonymous probe for services.

  An attacker studies the behavior of an idle host to gain information about the target, such as the services that the host offers, the ports supporting the services, and the operating system of the target. This type of scan is a predecessor to either heavier probing or outright attacks.

  The following are some of the features of Hping2/Hping3:

  - It determines whether the host is up even when the host blocks ICMP packets.
  - It aids advanced port scanning and test net performance using different protocols, packet sizes, TOS, and fragmentation.
  - Manual path MTU discovery
  - Firewalk-like usage allows discovery of open ports behind firewalls.
  - Remote OS fingerprinting and TCP/IP stack auditing

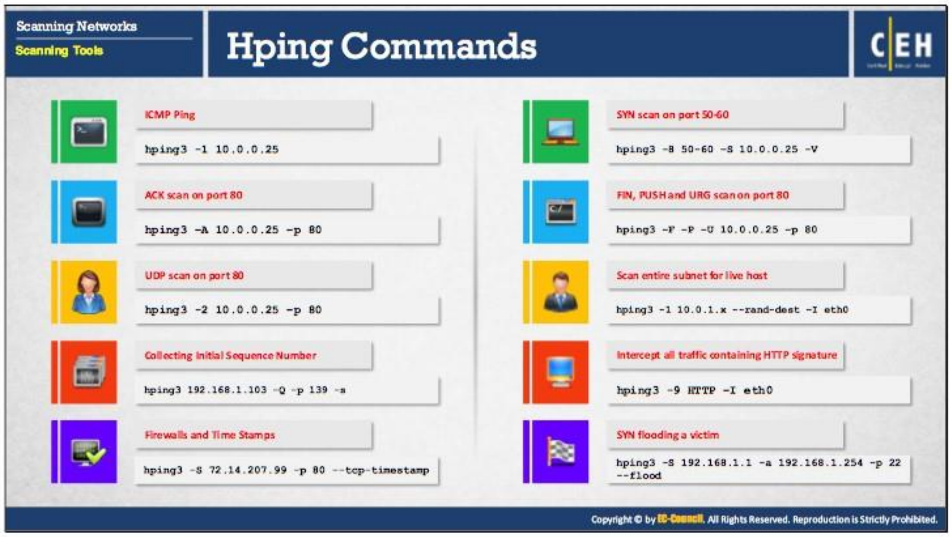
ICMP Scanning

A ping sweep or Internet Control Message Protocol (ICMP) scanning is a process of sending an ICMP request or ping to all hosts on the network to determine which one is up.

The operating system, router, switch, internet-protocol-based devices use this protocol via the ping command to Echo request and Echo response as a connectivity tester between different hosts.

ACK Scanning on Port 80

You can use this scan technique to probe for the existence of a firewall and its rule sets. Simple packet filtering allows you to establish a connection (packets with the ACKbitset), whereas a sophisticated stateful firewall does not allow you to establish a connection.



Scanning Networks
Scanning Tools

Hping Commands

CEH

ICMP Ping

```
hping3 -1 10.0.0.25
```

ACK scan on port 80

```
hping3 -A 10.0.0.25 -p 80
```

UDP scan on port 80

```
hping3 -2 10.0.0.25 -p 80
```

Collecting Initial Sequence Number

```
hping3 192.168.1.103 -Q -p 139 -s
```

Firewalls and Time Stamps

```
hping3 -S 72.14.207.99 -p 80 --tcp-timestamp
```

SYN scan on port 50-60

```
hping3 -8 50-60 -S 10.0.0.25 -V
```

FIN, PUSH and URG scan on port 80

```
hping3 -F -P -U 10.0.0.25 -p 80
```

Scan entire subnet for live host

```
hping3 -1 10.0.1.x --rand-dest -I eth0
```

Intercept all traffic containing HTTP signature

```
hping3 -9 HTTP -I eth0
```

SYN flooding a victim

```
hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22
--flood
```

Hping Commands

Below are various Hping commands:

- **ICMP ping**

  Ex. hping3 -1 10.0.0.25

  Hping performs an ICMP ping scan by specifying the argument -1 on the command line. You may use --ICMP of -1 argument in the command line. By issuing the above command, hping sends ICMP-echo request to 10.0.0.25 and receives ICMP-reply, the same as with a ping utility.

- **ACK scan on port 80**

  Ex. hping3 –A 10.0.0.25 –p 80

  Hping can be configured to perform an ACK scan by specifying the argument -A in the command line. Here, you are setting ACK flag in the probe packets and performing the scan. You perform this scan when a host does not respond to a ping request. By issuing this command, Hping checks if a host is alive on a network. If it finds a live host and an open port, it returns an RST response.

- **UDP scan on port 80**

  Ex. hping3 -2 10.0.0.25 –p 80

  Hping uses TCP as its default protocol. Using the argument -2 in the command line specifies that Hping operates in UDP mode. You may use either –udp of -2 arguments in the command line.

  By issuing the above command, Hping sends UDP packets to port 80 on the host (10.0.0.25). It returns an ICMP port unreachable message if it finds the port closed, and does not respond with a message if the port is open.

- **Collecting Initial Sequence Number**

  ```
  Ex. hping3 192.168.1.103 -Q -p 139 -s
  ```

  By using the argument -Q in the command line, Hping collects all the TCP sequence numbers generated by the target host (192.168.1.103).

- **Firewalls and Time Stamps**

  ```
  Ex. hping3 -S 72.14.207.99 -p 80 --tcp-timestamp
  ```

  Many firewalls drop those TCP packets that do not have TCP Timestamp option set. By adding the --tcp-timestamp argument in the command line, you can enable TCP timestamp option in Hping and try to guess the timestamp update frequency and uptime of the target host (72.14.207.99).

- **SYN scan on port 50-60**

  ```
  Ex. hping3 -8 50-60 –S 10.0.0.25 –V
  ```

  By using the argument -8 (or) --scan in the command, you are operating Hping in scan mode in order to scan a range of ports on the target host. Adding the argument -S allows you to perform a SYN scan.

  Therefore, the above command performs a SYN scan on ports 50-60 on the target host.

- **FIN, PUSH and URG scan on port 80**

  ```
  Ex. hping3 –F –P –U 10.0.0.25 –p 80
  ```

  By adding the arguments –F, -P, and –U in the command, you are setting FIN, PUSH, and URG packets in the probe packets. By issuing this command, you are performing FIN, PUSH, and URG scans on port 80 on the target host (10.0.0.25). If port 80 is open on the target, you will not receive a response. If the port is closed, Hping will return an RST response.

- **Scan entire subnet for live host**

  ```
  Ex. hping3 -1 10.0.1.x --rand-dest –I eth0
  ```

  By issuing this command, Hping performs an ICMP ping scan on the entire subnet 10.0.1.x; in other words, it sends ICMP-echo request randomly (–rand-dest) to all the hosts from 10.0.1.0 – 10.0.1.255 that are connected to the interface eth0. The hosts whose ports are open will respond with an ICMP-reply. In this case, you have not set a port, so Hping sends packets to port 0 on all IP addresses by default.

- **Intercept all traffic containing HTTP signature**

  ```
  Ex. hping3 -9 HTTP –I eth0
  ```

  The argument -9 will set the Hping to listen mode. So, by issuing the command -9 HTTP, Hping starts listening on port 0 (of all the devices connected in the network to interface eth0), intercepts all the packets containing HTTP signature, and dump from signature end to the packet's end.

  For example, on issuing the command `hping2 -9 HTTP`, if Hping reads a packet that contains data 234-09sdflkjs45-HTTPhello_world, it will display the result as hello_world.

- **SYN flooding a victim**

  `Ex. hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22 --flood`

  The attacker employs TCP SYN flooding techniques by using spoofed IP addresses to perform DoS attack.

The following table lists the various scanning methods and their respective Hping commands:

Scan	Commands
ICMP ping	**`hping3 -1 10.0.0.25`**
ACK scan on port 80	**`hping3 –A 10.0.0.25 –p 80`**
UDP scan on port 80	**`hping3 -2 10.0.0.25 –p 80`**
Collecting initial sequence number	**`hping3 192.168.1.103 -Q -p 139 -s`**
Firewalls and timestamps	**`hping3 -S 72.14.207.99 -p 80 --tcp-timestamp`**
SYN scan on port 50-60	**`hping3 -8 50-56 –S 10.0.0.25 -V`**
FIN, PUSH and URG scan on port 80	**`hping3 –F –P –U 10.0.0.25 –p 80`**
Scan entire subnet for live host	**`hping3 -1 10.0.1.x --rand-dest –I eth0`**
Intercept all traffic containing HTTP signature	**`hping3 -9 HTTP –I eth0`**
SYN flooding a victim	**`hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22 --flood`**

TABLE 3.1: Hping command and its respective function

Scanning Networks
Scanning Tools

Scanning Tools

CEH

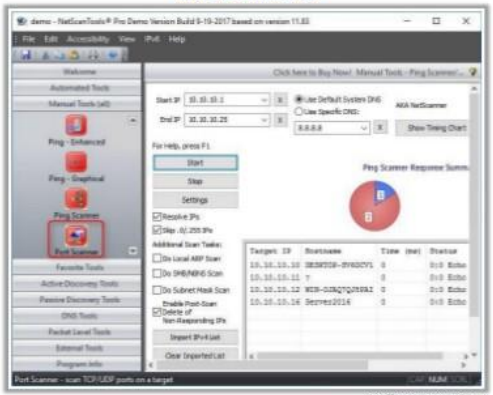
NetScanTools Pro

- Network Tools Pro assists in **troubleshooting**, **diagnosing**, **monitoring**, and **discovering** devices on the network
- It lists **IPv4/IPv6** addresses, hostnames, **domain names**, email addresses, and URLs automatically or manually (using manual tools)

Scanning Tools

- SuperScan (*https://www.mcafee.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- OmniPeek (*https://www.savvius.com*)
- MiTeC Network Scanner (*http://www.mitec.cz*)
- NEWT Professional (*http://www.komodolabs.com*)
- MegaPing (*http://www.magnetosoft.com*)

NetScanTools Pro


demo - NetScanTools® Pro Demo Version Build 9-19-2017 based on version 11.83

Start IP: 10.10.10.1
End IP: 10.10.10.25
● Use Default System DNS
○ Use Specific DNS: 8.8.8.8
AKA NetScanner

☑ Resolve IPs
☑ Skip .0/.255 IPs
Additional Scan Tasks:
☐ Do Local ARP Scan
☐ Do SMB/NBNS Scan
☐ Do Subnet Mask Scan
☑ Enable Post-Scan Delete of Non-Responding IPs

Target IP	Hostname	Time (ms)	Status
10.10.10.10	DESKTOP-8V6DCV1	0	0:0 Echo
10.10.10.11	?	0	0:0 Echo
10.10.10.12	WIN-O2AQ7QJ5SAI	0	0:0 Echo
10.10.10.16	Server2016	0	0:0 Echo

Port Scanner - scan TCP/UDP ports on a target

http://www.netscantools.com

Scanning Tools

- **NetScanTools Pro**

  Source: *http://www.netscantools.com*

  NetScanTools Pro is an investigation tool that allows you to troubleshoot, monitor, discover, and detect devices on your network. You can easily gather information about the local LAN, as well as Internet users, IP addresses, ports, and so on using this tool. You can find vulnerabilities and exposed ports in your system. NetScanTools Pro combines many network tools and utilities categorized by their functions, such as active, passive, DNS, and local computer.

  - **Active Discovery and Diagnostic Tools**: Used for testing and locating devices connected to your network.
  - **Passive Discovery Tools**: Monitor the activities of the devices connected to your network and gather information from third parties.
  - **DNS Tools**: Help to detect DNS problems.
  - **Local Computer and General Information Tools**: Provide details about your local computer's network.

  Benefits:

  - The information gathering process is made simpler and faster by automating the use of many network tools.
  - Clearly produces the result reports in your web browser.

Some of the scanning tools are listed below:

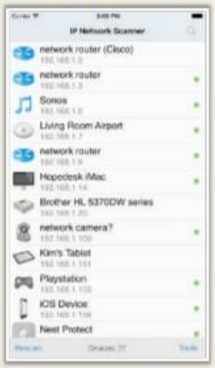
- SuperScan (*https://www.mcafee.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- OmniPeek (*https://www.savvius.com*)
- MiTeC Network Scanner (*http://www.mitec.cz*)
- NEWT Professional (*http://www.komodolabs.com*)
- MegaPing (*http://www.magnetosoft.com*)
- Slitheris Network Discovery (*http://www.komodolabs.com*)
- TamoSoft's CommView (*http://www.tamos.com*)
- IP Scanner (*https://community.spiceworks.com*)
- IP-Tools (*https://www.ks-soft.net*)
- Network Scanner (*http://www.10-strike.com*)
- Global Network Inventory (*http://www.magnetosoft.com*)
- SoftPerfect Network Scanner (*https://www.softperfect.com*)
- Advanced Port Scanner (*https://www.advanced-port-scanner.com*)
- CurrPorts (*http://www.nirsoft.net*)
- Masscan (*https://github.com*)
- DRACNMAP (*https://github.com*)
- NEET (*https://github.com*)

Scanning Networks
Scanning Tools

Scanning Tools for Mobile

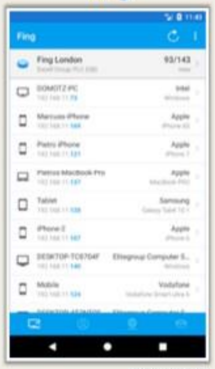
C|EH

IP Scanner



http://10base-t.com

Fing



https://www.fing.io

Hackode
https://play.google.com

zANTI
https://www.zimperium.com

cSploit
http://www.csploit.org

FaceNiff
http://www.effecthacking.com

PortDroid Network Analysis
https://play.google.com

Scanning Tools for Mobile

- **IP Scanner**

  Source: *http://10base-t.com*

  IP Scanner for iOS scans your local area network to determine the identity of all its active machines and Internet devices.

  Features:

  - In-built Ping, Portscan, and WOL tools.
  - Traverse to native VNC, web browser, or any custom service directly from the scan results.
  - Customizable display options for assigning names and icons to discovered devices.
  - Ability to create your custom device categories with your images.
  - Ability to export, email, and print scan results.

- **Fing**

  Source: *https://www.fing.io*

  Fing is a mobile app for Android and iOS that scans and provides complete network information, such as IP address, MAC address, device vendor, and ISP location.

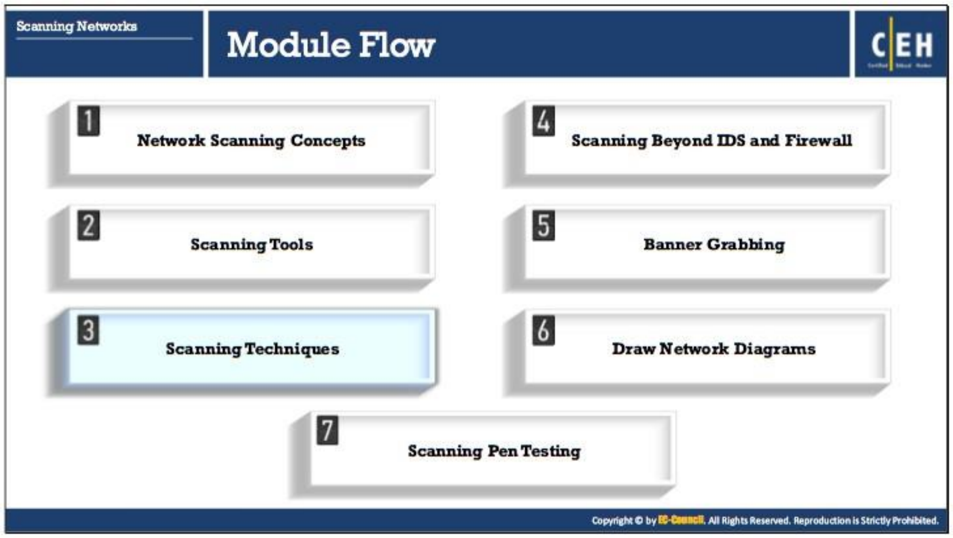
  Features:

  - Discovers all devices connected to a Wi-Fi network
  - Displays MAC Address and device manufacturer

- Full search by IP, MAC, Name, Vendor, and Notes
- Wake On LAN: Allows you to switch on your devices from mobile or tablet
- Ping and traceroute: Assesses network performance
- Automatic DNS lookup and reverse lookup
- Tracks when a device has gone online or offline
- Launch Apps for specific ports, such as Browser, SSH, FTP
- Displays NetBIOS names and properties
- Supports identification by IP address for bridged networks
- Sort by IP, MAC, Name, Vendor, State, and Last Change

Some of the scanning tools for mobile devices include:

- Hackode (*https://play.google.com*)
- zANTI (*https://www.zimperium.com*)
- cSploit (*http://www.csploit.org*)
- FaceNiff (*http://www.effecthacking.com*)
- PortDroid Network Analysis (*https://play.google.com*)
- Pamn IP Scanner (*https://play.google.com*)



Scanning Networks

Module Flow

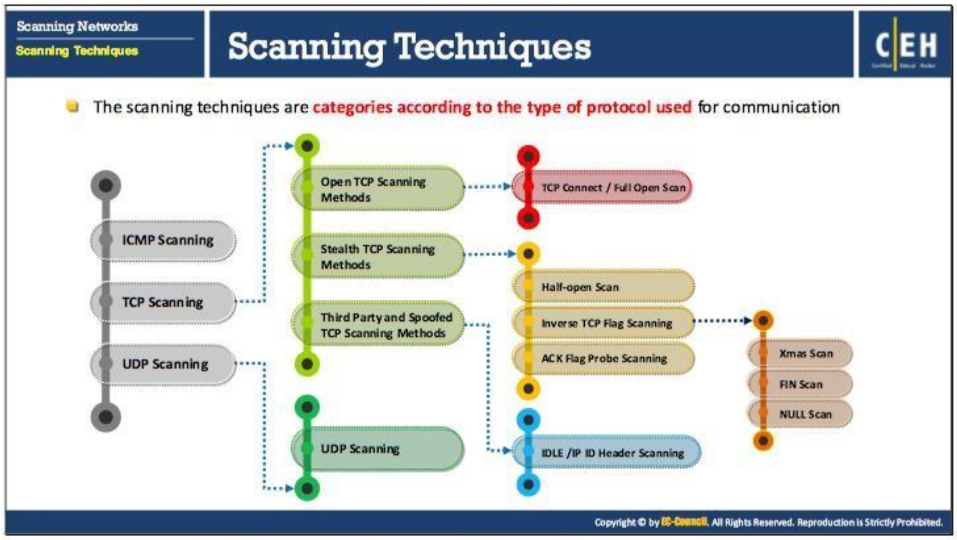
1. Network Scanning Concepts
2. Scanning Tools
3. Scanning Techniques
4. Scanning Beyond IDS and Firewall
5. Banner Grabbing
6. Draw Network Diagrams
7. Scanning Pen Testing

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Scanning Techniques

Scanning is the process of gathering information about systems that are "alive" and responding on the network. Port scanning techniques help an attacker to identify the open ports on a targeted server or host. Administrators often use port scanning techniques to verify security policies of their networks, whereas attackers use them to identify running services on a host with the intent of compromising the network.

The first step in scanning networks is to check for live systems. This section highlights how to check for live systems with the help of ICMP scanning, how to ping a system and various ping sweep tools. Once the attackers detect live systems in the target network, they try to find open ports in the discovered live systems. The next step in the network scanning process involves checking the open ports in live systems. Sometimes users unknowingly keep unnecessary open ports on their systems. Attacker takes advantages of such open ports to launch attacks. This section describes the tools and techniques used by an attacker to do so.



Scanning Networks
Scanning Techniques

Scanning Techniques

- The scanning techniques are **categories according to the type of protocol used** for communication

ICMP Scanning
TCP Scanning
UDP Scanning
Open TCP Scanning Methods
Stealth TCP Scanning Methods
Third Party and Spoofed TCP Scanning Methods
UDP Scanning
TCP Connect / Full Open Scan
Half-open Scan
Inverse TCP Flag Scanning
ACK Flag Probe Scanning
IDLE /IP ID Header Scanning
Xmas Scan
FIN Scan
NULL Scan

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Scanning techniques are further split into three categories as shown below; this is according to the type of protocol used for communication at the transport layer of the network.

Scanning ICMP Network Services:

- ICMP Scanning
- Ping Sweep
- ICMP Echo Scanning

Scanning TCP Network Services:

- Open TCP Scanning Methods
  - TCP Connect / Full Open Scan
- Stealth TCP Scanning Methods
  - Half-open Scan
  - Inverse TCP Flag Scanning
    - Xmas Scan
    - FIN Scan
    - NULL Scan
  - ACK Flag Probe Scanning
- Third Party and Spoofed TCP Scanning Methods
  - IDLE /IP ID Header Scanning

Scanning UDP Network Services:

- UDP Scanning

The following is the list of important reserved ports:

Name	Port/Protocol	Description
echo	7/tcp	
echo	7/udp	
discard	9/tcp	sink null
discard	9/udp	sink null
systat	11/tcp	Users
daytime	13/tcp	
daytime	13/udp	
netstat	15/tcp	
qotd	17/tcp	Quote
chargen	19/tcp	ttytst source
chargen	19/udp	ttytst source
ftp-data	20/tcp	ftp data transfer
ftp	21/tcp	ftp command
ssh	22/tcp	Secure Shell
telnet	23/tcp	
SMTP	25/tcp	Mail
time	37/tcp	Timeserver
time	37/udp	Timeserver
rlp	39/udp	resource location
nickname	43/tcp	who is
domain	53/tcp	domain name server
domain	53/udp	domain name server
sql*net	66/tcp	Oracle SQL*net
sql*net	66/udp	Oracle SQL*net
bootps	67/tcp	bootp server
bootps	67/udp	bootp server
bootpc	68/tcp	bootp client
bootpc	68/udp	bootp client
tftp	69/tcp	Trivial File Transfer
tftp	69/udp	Trivial File Transfer
gopher	70/tcp	gopher server
finger	79/tcp	Finger
www-http	80/tcp	WWW
www-http	80/udp	WWW
kerberos	88/tcp	Kerberos
kerberos	88/udp	Kerberos
pop2	109/tcp	PostOffice V.2

Pop3	110/tcp	PostOffice V.3
sunrpc	111/tcp	RPC 4.0 portmapper
sunrpc	111/udp	RPC 4.0 portmapper
auth/ident	113/tcp	Authentication Service
auth	113/udp	Authentication Service
audionews	114/tcp	Audio News Multicast
audionews	114/udp	Audio News Multicast
nntp	119/tcp	Usenet Network News Transfer
nntp	119/udp	Usenet Network News Transfer
ntp	123/tcp	Network Time Protocol
Name	Port/Protocol	Description
ntp	123/udp	Network Time Protocol
netbios-ns	137/tcp	NETBIOS Name Service
netbios-ns	137/udp	NETBIOS Name Service
netbios-dgm	138/tcp	NETBIOS Datagram Service
netbios-dgm	138/udp	NETBIOS Datagram Service
netbios-ssn	139/tcp	NETBIOS Session Service
netbios-ssn	139/udp	NETBIOS Session Service
imap	143/tcp	Internet Message Access Protocol
imap	143/udp	Internet Message Access Protocol
sql-net	150/tcp	SQL-NET
sql-net	150/udp	SQL-NET
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
snmp	161/tcp	
snmp	161/udp	
snmp-trap	162/tcp	
snmp-trap	162/udp	
cmip-man	163/tcp	CMIP/TCP Manager
cmip-man	163/udp	CMIP
cmip-agent	164/tcp	CMIP/TCP Agent
cmip-agent	164/udp	CMIP
irc	194/tcp	Internet Relay Chat
irc	194/udp	Internet Relay Chat
at-rtmp	201/tcp	AppleTalk Routing Maintenance
at-rtmp	201/udp	AppleTalk Routing Maintenance
at-nbp	202/tcp	AppleTalk Name Binding
at-nbp	202/udp	AppleTalk Name Binding
at-3	203/tcp	AppleTalk
at-3	203/udp	AppleTalk

at-echo	204/tcp	AppleTalk Echo
at-echo	204/udp	AppleTalk Echo
at-5	205/tcp	AppleTalk
at-5	205/udp	AppleTalk
at-zis	206/tcp	AppleTalk Zone Information
at-zis	206/udp	AppleTalk Zone Information
at-7	207/tcp	AppleTalk
at-7	207/udp	AppleTalk
at-8	208/tcp	AppleTalk
at-8	208/udp	AppleTalk
ipx	213/tcp	Novel
ipx	213/udp	Novel
imap3	220/tcp	Interactive Mail Access Protocol v3
imap3	220/udp	Interactive Mail Access Protocol v3
aurp	387/tcp	AppleTalk Update-Based Routing
aurp	387/udp	AppleTalk Update-Based Routing
netware-ip	396/tcp	Novell Netware over IP
netware-ip	396/udp	Novell Netware over IP
Name	Port/Protocol	Description
rmt	411/tcp	Remote mt
rmt	411/udp	Remote mt
kerberos-ds	445/tcp	Microsoft DS
kerberos-ds	445/udp	Microsoft DS
isakmp	500/udp	ISAKMP/IKE
fcp	510/tcp	First Class Server
exec	512/tcp	BSD rexecd(8)
comsat/biff	512/udp	used by mail system to notify users
login	513/tcp	BSD rlogind(8)
who	513/udp	whod BSD rwhod(8)
shell	514/tcp	cmd BSD rshd(8)
syslog	514/udp	BSD syslogd(8)
printer	515/tcp	spooler BSD lpd(8)
printer	515/udp	Printer Spooler
talk	517/tcp	BSD talkd(8)
talk	517/udp	Talk
ntalk	518/udp	New Talk (ntalk)
ntalk	518/udp	SunOS talkd(8)
netnews	532/tcp	Readnews
uucp	540/tcp	uucpd BSD uucpd(8)
uucp	540/udp	uucpd BSD uucpd(8)

klogin	543/tcp	Kerberos Login
klogin	543/udp	Kerberos Login
kshell	544/tcp	Kerberos Shell
kshell	544/udp	Kerberos Shell
ekshell	545/tcp	krcmd Kerberos encrypted remote shell – kfall
pcserver	600/tcp	ECD Integrated PC board srvr
mount	635/udp	NFS Mount Service
pcnfs	640/udp	PC-NFS DOS Authentication
bwnfs	650/udp	BW-NFS DOS Authentication
flexlm	744/tcp	Flexible License Manager
flexlm	744/udp	Flexible License Manager
kerberos-adm	749/tcp	Kerberos Administration
kerberos-adm	749/udp	Kerberos Administration
kerberos	750/tcp	kdc Kerberos authentication—tcp
kerberos	750/udp	Kerberos
kerberos_master	751/udp	Kerberos authentication
kerberos_master	751/tcp	Kerberos authentication
krb_prop	754/tcp	Kerberos slave propagation
	999/udp	Applixware
socks	1080/tcp	
socks	1080/udp	
kpop	1109/tcp	Pop with Kerberos
ms-sql-s	1433/tcp	Microsoft SQL Server
ms-sql-s	1433/udp	Microsoft SQL Server
ms-sql-m	1434/tcp	Microsoft SQL Monitor
ms-sql-m	1434/udp	Microsoft SQL Monitor
pptp	1723/tcp	Pptp
pptp	1723/udp	Pptp
nfs	2049/tcp	Network File System
nfs	2049/udp	Network File System
eklogin	2105/tcp	Kerberos encrypted rlogin
rkinit	2108/tcp	Kerberos remote kinit
kx	2111/tcp	X over Kerberos
kauth	2120/tcp	Remote kauth
lyskom	4894/tcp	LysKOM (conference system)

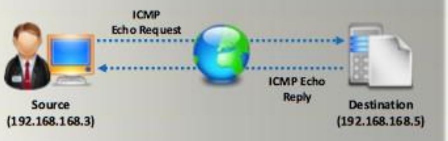
sip	5060/tcp	Session Initiation Protocol
sip	5060/udp	Session Initiation Protocol
x11	6000-6063/tcp	X Window System
x11	6000-6063/udp	X Window System
irc	6667/tcp	Internet Relay Chat
afs	7000-7009/udp	Andrew File System
afs	7000-7009/udp	Andrew File System

TABLE 3.2: Reserved ports table

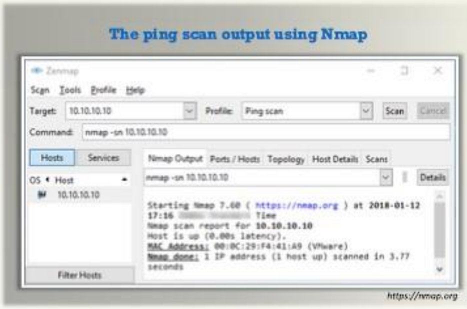
Scanning Networks
Scanning Techniques

ICMP Scanning - Checking for Live Systems

- Ping scan involves sending **ICMP ECHO requests** to a host. If the host is alive, it will return an ICMP ECHO reply
- This scan is useful for **locating active devices** or determining if the **ICMP is passing through a firewall**



The ping scan output using Nmap



Target: 10.10.10.10
Profile: Ping scan
Command: nmap -sn 10.10.10.10

OS Host
10.10.10.10

nmap -sn 10.10.10.10

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12
17:16 [redacted] Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
MAC Address: 00:0C:29:F4:41:A9 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 3.77
seconds
```

https://nmap.org

ICMP Scanning - Checking for Live Systems

Attackers use ICMP scanning to send ICMP packets to the destination system to gather all necessary information about it. This is so because ICMP does not have a port abstraction and it is not the same as port scanning. However, it is useful to determine what hosts in a network are running by pinging them all (Nmap uses the -P option to ICMP scan in parallel, which can happen quickly). The user can also increase the number of pings in parallel using the -L option. It can also be helpful to tweak the ping timeout value using the -T option. Ping scan involves sending ICMP ECHO requests to a host. If the host is alive, it will return an ICMP ECHO reply. This scan is useful for locating active devices or determining if ICMP is passing through a firewall.

ICMP Query

The UNIX tool ICMP query or ICMPush requests the system time (to learn the system time zone) by sending an ICMP type 13 message (TIMESTAMP). The netmask on a particular system can also be determined using ICMP type 17 messages (ADDRESS MARK REQUEST). After finding the netmask of a network card, a user can determine all the subnets in use. Then, the user can target only one particular subnet and avoid hitting the broadcast addresses.

ICMP query has both a timestamp and address mask request option:

```
ICMP query <-query-> [-B] [-f fromhost] [-d delay] [-T time] target
```

Where, **`<query>`** is one of:

- `-t`: ICMP timestamp request (default)
- `-m`: ICMP address mask request
- `-d`: delay to sleep between packets is in microseconds
- `-T` - specifies the number of seconds to wait for a host to respond. The default is 5.

A `target` is a list of hostnames or addresses.

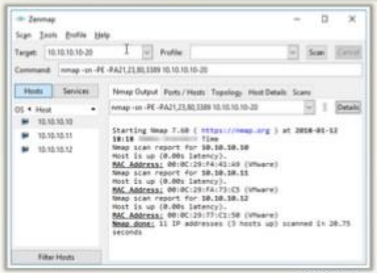
Scanning Networks
Scanning Techniques

Ping Sweep - Checking for Live Systems

CEH

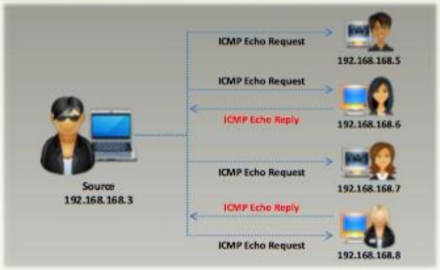
- Ping sweep is used to determine the **live hosts from a range of IP addresses** by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply
- Attackers calculate subnet masks by using the **Subnet Mask Calculators** to identify the number of hosts that are present in the subnet
- Attackers subsequently use ping sweep to create an **inventory of live systems** in the subnet

The ping sweep output using Nmap



```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12
10:10 [illegible] Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
MAC Address: 00:0C:29:F4:43:A9 (VMware)
Nmap scan report for 10.10.10.11
Host is up (0.00s latency).
MAC Address: 00:0C:29:FA:73:C5 (VMware)
Nmap scan report for 10.10.10.12
Host is up (0.00s latency).
MAC Address: 00:0C:29:77:C1:50 (VMware)
Nmap done: 11 IP addresses (3 hosts up) scanned in 20.75
seconds
```

https://nmap.org



Ping Sweep- Checking for Live Systems

A ping sweep (also known as an ICMP sweep) is a basic network scanning technique that is employed to determine which range of IP addresses map to live hosts (computers). Although a single ping will tell the user whether one specified host computer exists on the network, a ping sweep consists of ICMP ECHO requests sent to multiple hosts. If a specified host is active, it will return an ICMP ECHO reply.

Ping sweeps are among the oldest and slowest methods used to scan a network. This utility distributed across almost all the platforms acts as a roll call for systems; a system that is active on the network answers the ping query that another system sends out.

To understand pings better, one should be able to understand the TCP/IP packet. When a system pings, it sends a single packet across the network to a specific IP address. This packet contains 64 bytes (56 data bytes and 8 bytes of protocol header information). The sender then waits or listens for a return packet from the target system. If the connections are good and the target computer is "alive," a good return packet is expected. However, this will not be the case if there is a disruption in the communication. Ping also details the amount of time it takes for a packet to make the complete trip called the "round-trip time." Ping also helps in resolving hostnames. In this case, if the packet bounces back when sent to the IP address, but not when sent to the name, then the system is unable to resolve the name to the specific IP address.

Attackers calculate subnet masks using Subnet Mask Calculators to identify the number of hosts that are present in the subnet. Attackers subsequently use ping sweep to create an inventory of live systems in the subnet.

Scanning Networks
Scanning Techniques

Ping Sweep Tools

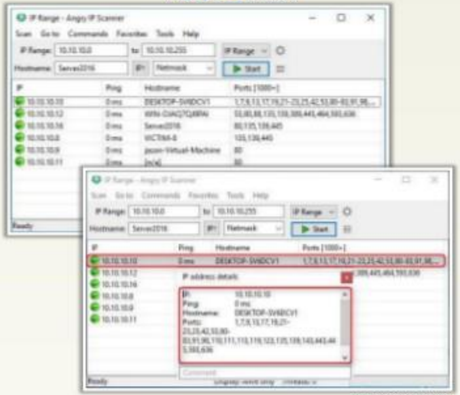
Angry IP Scanner

Angry IP Scanner pings each IP address to check if they are alive, then it optionally resolves its hostname, determines the MAC address, scans ports, etc.

Ping Sweep Tools

- SolarWinds Engineer's Toolset (*http://www.solarwinds.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Colasoft Ping Tool (*http://www.colasoft.com*)
- Visual Ping Tester (*http://www.pingtester.net*)
- OpUtils (*https://www.manageengine.com*)

Angry IP Scanner



IP Range: 10.10.10.0 to: 10.10.10.255
Hostname: Server2016 IP↑ Netmask

IP	Ping	Hostname	Ports [1000+]
10.10.10.10	0 ms	DESKTOP-SV6DCV1	1,7,9,13,17,19,21-23,25,42,53,80-83,91,98,...
10.10.10.12	0 ms	WIN-OJAQ7QJ8RAI	53,80,88,135,139,389,443,464,593,636
10.10.10.16	0 ms	Server2016	80,135,139,445
10.10.10.8	0 ms	VICTIM-8	135,139,445
10.10.10.9	0 ms	jason-Virtual-Machine	80
10.10.10.11	0 ms	[n/a]	80

IP address details

IP: 10.10.10.10
Ping: 0 ms
Hostname: DESKTOP-SV6DCV1
Ports: 1,7,9,13,17,19,21-23,25,42,53,80-83,91,98,110,111,113,119,123,135,139,143,443,445,593,636

http://www.angryip.org

Ping Sweep Tools

Ping sweep tools ping an entire range of network IP addresses to identify the live systems. Given below are ping sweep tools that enable one to determine live hosts on the target network by sending multiple ICMP ECHO requests to various hosts on the network at a time.

- **Angry IP Scanner**

  Source: *http://www.angryip.org*

  Angry IP scanner is an IP address and port scanner. It can scan IP addresses at any range as well as any of their ports. It pings each IP address to check if they are alive, then it optionally resolves its hostname, determines the MAC address, scans ports, and so on. The amount of data gathered about each host extends with plugins. Angry IP scanner has additional features, such as NetBIOS information (computer name, workgroup name, and currently logged in Windows user), favorite IP address ranges, web server detection, and customizable openers. The tool allows the user to save the Scanning results to CSV, TXT, XML, or IP-Port list files. To increase scanning speed, it uses a multithreaded approach: a separate scanning thread created for each scanned IP address.

Listed below are a few more ping sweep tools that an attacker one to determine live hosts on the target network:

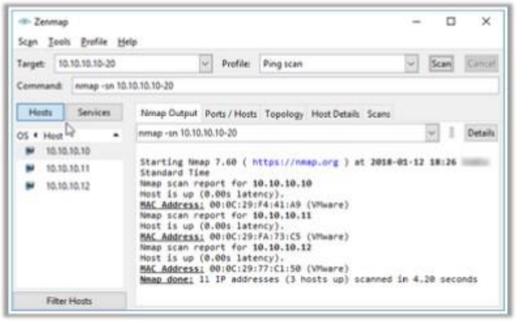
- SolarWinds Engineer's Toolset (*http://www.solarwinds.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Colasoft Ping Tool (*http://www.colasoft.com*)
- Visual Ping Tester (*http://www.pingtester.net*)
- OpUtils (*https://www.manageengine.com*)

- Advanced IP Scanner (*http://www.advanced-ip-scanner.com*)
- PingInfoView (*http://www.nirsoft.net*)
- Ping Monitor (*http://www.niliand.com*)
- Pinkie (*http://www.ipuptime.net*)
- MegaPing (*http://www.magnetosoft.com*)

Scanning Networks
Scanning Techniques

ICMP Echo Scanning

- In the real sense, this is not port scanning, since the **ICMP** does not have a port abstraction
- However, it is sometimes useful in determining which hosts in a network is up by **pinging** all of them
- `nmap -P cert.org/24 152.148.0.0/16`



Target: 10.10.10.10-20 Profile: Ping scan
Command: nmap -sn 10.10.10.10-20

```
nmap -sn 10.10.10.10-20

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12 18:26
Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
MAC Address: 00:0C:29:F4:41:A9 (VMware)
Nmap scan report for 10.10.10.11
Host is up (0.00s latency).
MAC Address: 00:0C:29:FA:73:C5 (VMware)
Nmap scan report for 10.10.10.12
Host is up (0.00s latency).
MAC Address: 00:0C:29:77:C1:50 (VMware)
Nmap done: 11 IP addresses (3 hosts up) scanned in 4.20 seconds
```

ICMP Echo Scanning

ICMP echo scanning pings all the machines in the target network to discover live machines. Attackers send ICMP probes to the broadcast or network address which relays to all the host addresses in the subnet. The live systems will send ICMP echo reply message to the source of the ICMP echo probe.

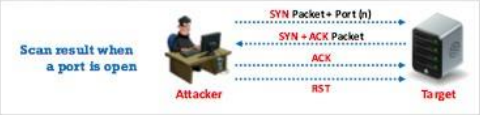
UNIX/Linux and BSD-based machines use ICMP echo scanning; the TCP/IP stack implementations in these operating system respond to the ICMP echo requests to the broadcast addresses. This technique does not work on Windows-based networks, as their TCP/IP stack implementation does not reply to ICMP probes directed at the broadcast address.

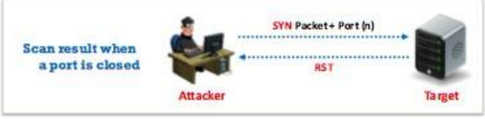
ICMP echo scanning is not same as port scanning because it does not have a port abstraction. ICMP echo scanning is used to determine the particular hosts that are active in a network by pinging all of them. Active hosts are displayed in Zenmap as "Host is up (0.0000s latency)," as shown in the screenshot above.

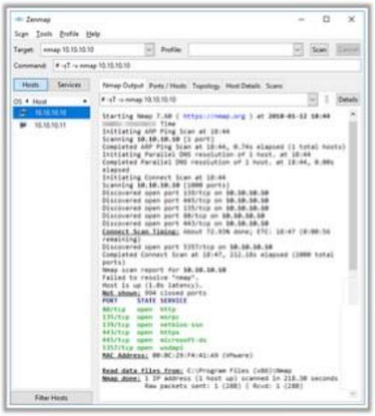
Scanning Networks
Scanning Techniques

TCP Connect / Full Open Scan

- The TCP Connect scan detects when a port is open after completing the **three-way handshake**
- TCP Connect scan **establishes a full connection** and tears it down by sending an **RST packet**
- It does not require the **super user privileges**







Target: nmap 10.10.10.10

Command: # -sT -v nmap 10.10.10.10

```
# -sT -v nmap 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12 10:44
[illegible] Time
Initiating ARP Ping Scan at 10:44
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 10:44, 0.74s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:44
Completed Parallel DNS resolution of 1 host. at 10:44, 0.00s
elapsed
Initiating Connect Scan at 10:44
Scanning 10.10.10.10 [1000 ports]
Discovered open port 139/tcp on 10.10.10.10
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 135/tcp on 10.10.10.10
Discovered open port 80/tcp on 10.10.10.10
Discovered open port 443/tcp on 10.10.10.10
Connect Scan Timing: About 72.93% done; ETC: 10:47 (0:00:56
remaining)
Discovered open port 5357/tcp on 10.10.10.10
Completed Connect Scan at 10:47, 212.18s elapsed (1000 total
ports)
Nmap scan report for 10.10.10.10
Failed to resolve "nmap".
Host is up (1.0s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
5357/tcp open  wsdapi
MAC Address: 00:0C:29:FA:41:A9 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 218.30 seconds
           Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

TCP Connect / Full Open Scan

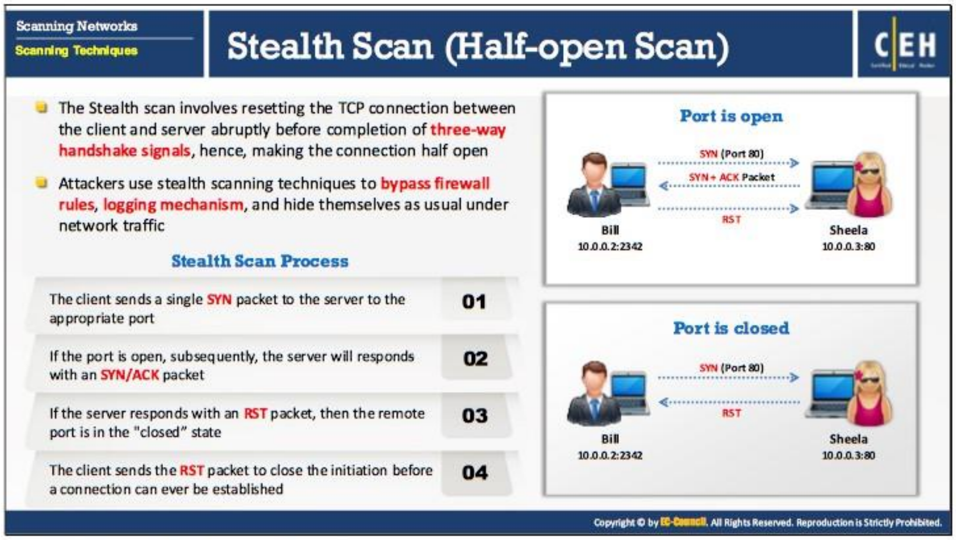
Source: *http://insecure.org*

TCP Connect/Full Open Scan is one of the most reliable forms of TCP scanning. In TCP Connect scanning, the operating system's TCP connect() system call tries to open a connection to every interesting port on the target machine. If the port is listening, the connect() call will result in a successful connection with the host on that particular port; otherwise, it will return an error message stating that the port is not reachable.

TCP Connect scan completes a three-way handshake with the target machine. In the TCP three-way handshake, the client sends a SYN packet, which the recipient acknowledges with a SYN+ACK packet. In turn, the client acknowledges the SYN+ACK packet with an ACK packet to complete the connection. Once the handshake is completed, the scanner sends a RST packet to end the connection.

Making a separate connect() call for every targeted port in a linear fashion would take a long time over a slow connection. The attacker can accelerate the scan by using many sockets in parallel. Using non-blocking, I/O allows the attacker to set a low time-out period and watch all the sockets simultaneously.

The drawback of this type of scan is that it is easily detectable and filterable. The logs in the target system will disclose the connection. This type of scanning does not require the superuser privileges.



Scanning Networks
Scanning Techniques

Stealth Scan (Half-open Scan)

- The Stealth scan involves resetting the TCP connection between the client and server abruptly before completion of **three-way handshake signals**, hence, making the connection half open
- Attackers use stealth scanning techniques to **bypass firewall rules**, **logging mechanism**, and hide themselves as usual under network traffic

Stealth Scan Process

The client sends a single **SYN** packet to the server to the appropriate port	01
If the port is open, subsequently, the server will responds with an **SYN/ACK** packet	02
If the server responds with an **RST** packet, then the remote port is in the "closed" state	03
The client sends the **RST** packet to close the initiation before a connection can ever be established	04

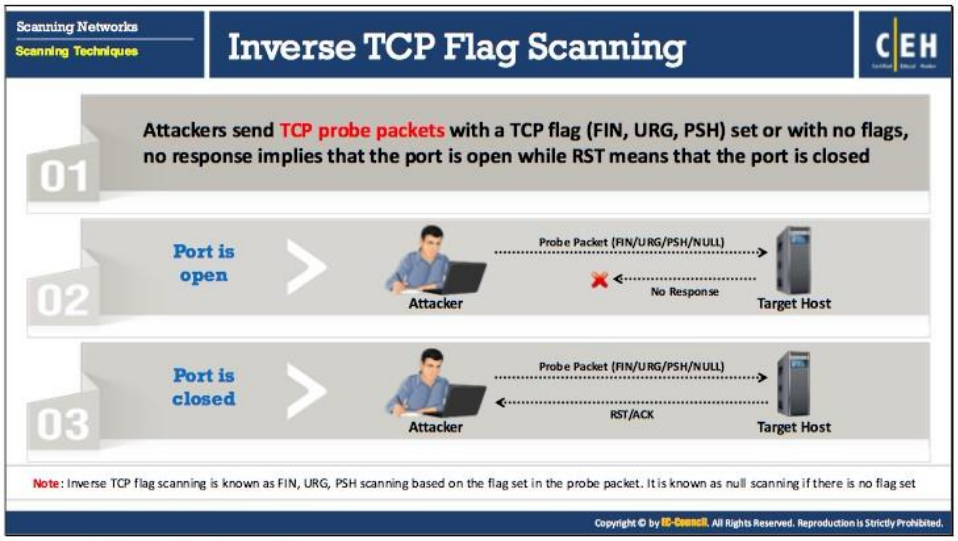
Stealth Scan (Half-open Scan)

The Stealth scan involves resetting the TCP connection between client and server abruptly before completion of the three-way handshake signals, hence, making the connection half open. A stealth scan sends a single frame to a TCP port without any TCP handshaking or additional packet transfers. This type of scan sends a single frame with the expectation of a single response. The half-open scan partially opens a connection but stops halfway through. The stealth scan is also called a "SYN scan," because it only sends the SYN packet. This prevents the service from notifying the incoming connection. TCP SYN or half-open scanning is a stealth method of port scanning.

The stealth scan also implements the three-way handshake methodology. In the last stage, it examines the packets entering the interface and terminating the connection before triggering a new initialization to identify remote ports. The stealth scan process is shown below.

- The client sends a single SYN packet to the server on the appropriate port.
- If the port is open, subsequently, the server responds with an SYN/ACK packet.
- If the server responds with an RST packet, then the remote port is in the "closed" state.
- The client sends the RST packet to close the initiation before a connection can ever be established.

Attackers use stealth scanning techniques to bypass firewall rules, logging mechanism, and hide themselves as usual under network traffic.



Scanning Networks
Scanning Techniques

Inverse TCP Flag Scanning

01 Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags, no response implies that the port is open while RST means that the port is closed

Note: Inverse TCP flag scanning is known as FIN, URG, PSH scanning based on the flag set in the probe packet. It is known as null scanning if there is no flag set

Inverse TCP Flag Scanning

Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set, or with no flags. When the port is open, the attacker does not get any response from the host, whereas when the port is closed, he or she receives the RST from the target host.

Security mechanisms such as firewalls and IDS detect the SYN packets sent to the sensitive ports of the targeted hosts. Programs such as Synlogger and Courtney are available to log half-open SYN flag scan attempts. At times, the probe packets enabled with TCP flags can pass through filters undetected, depending on the security mechanisms installed.

Inverted Technique is an act of probing a target using a half-open SYN flag because the closed ports can only send the response back. According to RFC 793, an RST/ACK packet sent for connection reset, when the host closes a port. Attackers take advantage of this feature to send TCP probe packets to each port of the target host with various TCP flags set.

Common flag configurations used for a probe packet include:

- A FIN probe with the FIN TCP flag set
- An XMAS probe with the FIN, URG, and PUSH TCP flags set
- A NULL probe with no TCP flags set
- A SYN/ACK probe

All closed ports on the targeted host will send an RST/ACK response. Since operating systems such as the Windows completely ignore the RFC 793 standard, you cannot see the RST/ACK response when connected to a closed port on the target host. However, this technique is effective when used with UNIX-based operating systems.

Advantages

- Avoids many IDS and logging systems, highly stealthy

Disadvantages

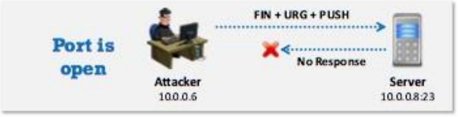
- Needs raw access to network sockets, thus requiring super-user privileges
- Mostly effective against hosts using a BSD-derived TCP/IP stack (not effective against Microsoft Windows hosts, in particular).

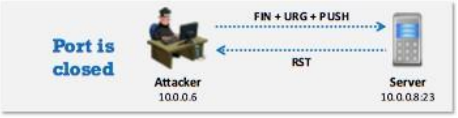
Note: Inverse TCP flag scanning is known as FIN, URG, and PSH scanning based on the flag set in the probe packet. If there is no flag set, it is known as null scanning.

Scanning Networks
Scanning Techniques

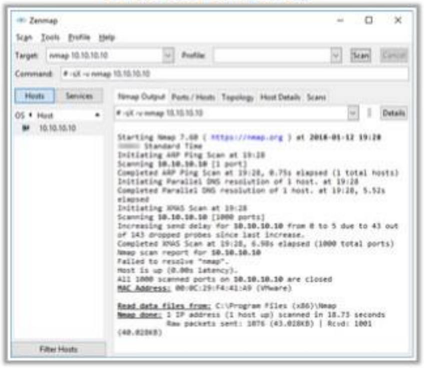
Xmas Scan

- In Xmas scan, attackers send a TCP frame to a remote device with **FIN**, **URG**, and **PUSH** flags set
- FIN scan works only with OSes with **RFC 793-based** TCP/IP implementation
- It will not work against any current version of **Microsoft Windows**





Xmas Scan Using Nmap



```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12 19:28
[illegible] Standard Time
Initiating ARP Ping Scan at 19:28
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 19:28, 0.75s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:28
Completed Parallel DNS resolution of 1 host. at 19:28, 5.52s
elapsed
Initiating XMAS Scan at 19:28
Scanning 10.10.10.10 [1000 ports]
Increasing send delay for 10.10.10.10 from 0 to 5 due to 43 out
of 143 dropped probes since last increase.
Completed XMAS Scan at 19:28, 6.98s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Failed to resolve "nmap".
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.10.10 are closed
MAC Address: 00:0C:29:F4:41:A9 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 18.73 seconds
           Raw packets sent: 1076 (43.028KB) | Rcvd: 1001
(40.028KB)
```

Xmas Scan

Xmas scan is a port scan technique with FIN, URG, and PUSH flags set to send a TCP frame to a remote device. If the target has opened the port, then you will receive no response from the remote system. If the target has closed the port, then you will receive a remote system reply with a RST. You can use this port scanning technique to scan large networks and find which host is up and what services it is offering. It is a technique to describe all TCP flag sets. When all flags are set, some systems hang; so the flags most often set are the nonsense pattern URG-PSH-FIN. Attackers use TCP XMAS scan to determine if ports are closed on the target machine via RST packet. This scan only works when systems are compliant with RFC 793-based TCP/IP implementation. It will not work against any current version of Microsoft Windows.

BSD Networking Code

This method relies on BSD networking code. Thus, you can use this only for UNIX hosts; it does not support Windows NT. If the user scans any Microsoft system, it will show that all the ports on the host are open.

Transmitting Packets

You can initialize all the flags when transmitting the packet to a remote host. If the target system accepts the packet and does not send any response, it means that the port is open. If the target system sends RST flag, then it implies that the port is closed.

Advantages

- It avoids the IDS and TCP three-way handshake.

Disadvantages

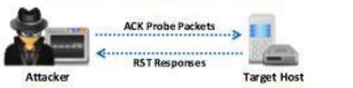
- It works on the UNIX platform only.

Scanning Networks
Scanning Techniques

ACK Flag Probe Scanning

- Attackers send **TCP probe packets with ACK flag** set to a remote device and then **analyzes the header information** (TTL and WINDOW field) of received RST packets to find out if the **port is open or closed**

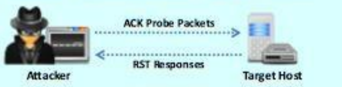
TTL based ACK flag probe scanning



```
1: host 10.2.2.11 port 20: F:RST -> ttl: 80 win: 0
2: host 10.2.2.11 port 21: F:RST -> ttl: 80 win: 0
3: host 10.2.2.11 port 22: F:RST -> ttl: 50 win: 0
4: host 10.2.2.11 port 23: F:RST -> ttl: 80 win: 0
```

If the **TTL value of RST packet** on a particular port is less than the boundary value of **64**, then that **port is open**

WINDOW based ACK flag probe scanning



```
1: host 10.2.2.12 port 20: F:RST -> ttl: 64 win: 0
2: host 10.2.2.12 port 21: F:RST -> ttl: 64 win: 0
3: host 10.2.2.12 port 22: F:RST -> ttl: 64 win: 512
4: host 10.2.2.12 port 23: F:RST -> ttl: 64 win: 0
```

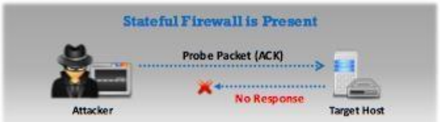
If the **WINDOW value of RST packet** on a particular port has a **non zero value**, then that **port is open**

Scanning Networks
Scanning Techniques

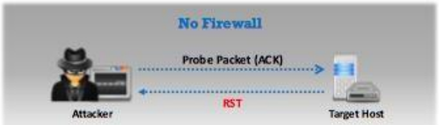
ACK Flag Probe Scanning (Cont'd)

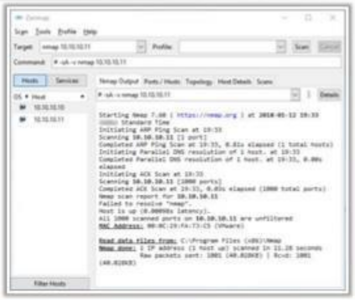
- ACK flag probe scanning can also be used to **check the filtering system of target**
- Attackers send an **ACK probe packet** with a random sequence number, no response implies that the **port is filtered** (stateful firewall is present) and RST response means that the **port is not filtered**

Stateful Firewall is Present



No Firewall





```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-05-12 19:33
      Standard Time
Initiating ARP Ping Scan at 19:33
Scanning 10.10.10.11 [1 port]
Completed ARP Ping Scan at 19:33, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:33
Completed Parallel DNS resolution of 1 host. at 19:33, 0.00s
elapsed
Initiating ACK Scan at 19:33
Scanning 10.10.10.11 [1000 ports]
Completed ACK Scan at 19:33, 0.03s elapsed (1000 total ports)
Nmap scan report for 10.10.10.11
Failed to resolve "nmap".
Host is up (0.00050s latency).
All 1000 scanned ports on 10.10.10.11 are unfiltered
MAC Address: 00:0C:29:FA:73:C5 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 11.28 seconds
           Raw packets sent: 1001 (40.028KB) | Rcvd: 1001
(40.028KB)
```

ACK Flag Probe Scanning

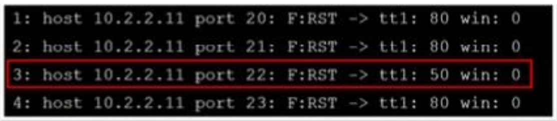
Attackers send TCP probe packets with the ACK flag set to a remote device and then analyze the header information (TTL and WINDOW field) of the received RST packets to find out if the port is open or closed. The ACK flag probe scanning exploits the vulnerabilities within BSD derived TCP/IP stack. Thus, this scanning is effective only on those operating systems and platforms on which the BSD derives TCP/IP stacks.

Categories of ACK Flag Probe Scanning include:

- **TTL-based ACK flag probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (thousands in number) to different TCP ports, and then analyze the TTL field value of the RST packets received.

  If the TTL value of RST packet on a particular port is less than the boundary value of 64, then that port is open. Here is an example displaying a log of the first four RST packets received:



```
1: host 10.2.2.11 port 20: F:RST -> ttl: 80 win: 0
2: host 10.2.2.11 port 21: F:RST -> ttl: 80 win: 0
3: host 10.2.2.11 port 22: F:RST -> ttl: 50 win: 0
4: host 10.2.2.11 port 23: F:RST -> ttl: 80 win: 0
```

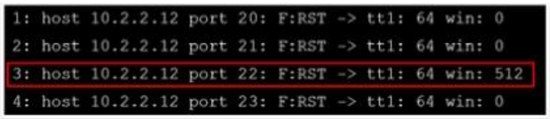
FIGURE 3.1: Screenshot showing the open port based on the TTL value of RST packet

  In the above example, port 22 has returned a TTL value of 50, which is less than 64; all other ports returned a TTL value of 80, which is greater than 64. Therefore, port 22 is open.

- **WINDOW based ACK flag probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (thousands in number) to different TCP ports, and then analyze the Window field value of the received RST packets. The user can use this scanning technique when all the ports return the same TTL value.

  If the WINDOW value of RST packet on a particular port has a non-zero value, then that port is open. Here is an example displaying a log of the first four RST packets received:



```
1: host 10.2.2.12 port 20: F:RST -> ttl: 64 win: 0
2: host 10.2.2.12 port 21: F:RST -> ttl: 64 win: 0
3: host 10.2.2.12 port 22: F:RST -> ttl: 64 win: 512
4: host 10.2.2.12 port 23: F:RST -> ttl: 64 win: 0
```

FIGURE 3.2: Screenshot showing the open port based on the Window value of RST packet

  The above figure shows that the TTL value returned for each packet is the same, so you cannot perform TTL based ACK flag probe scanning to find the open ports. Therefore, when you observe the window value, the third packet has a non-zero window value, which means that the port is open.

Advantages:

- This type of scan can evade IDS in most cases.

Disadvantages:

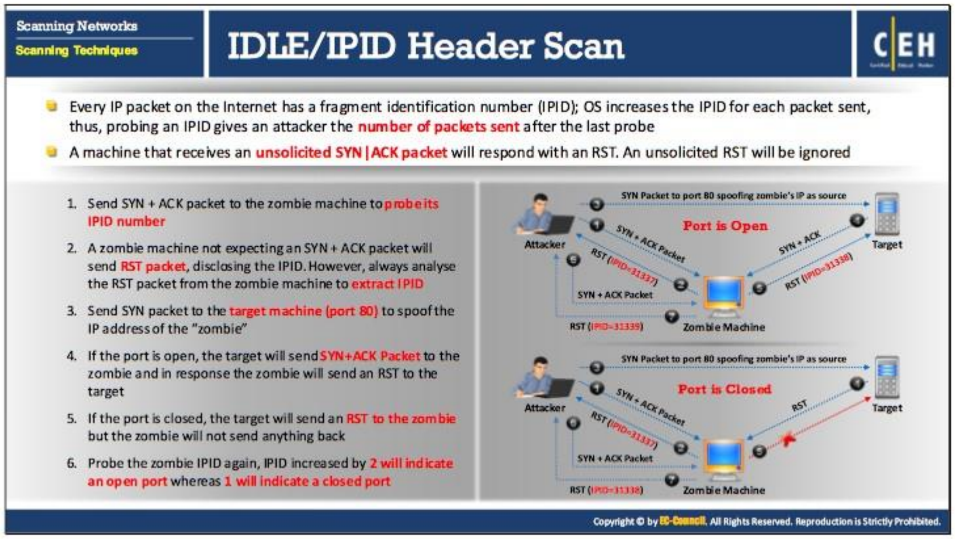
- This scan is very slow and can exploit only older operating systems with vulnerable BSD derived TCP/IP stacks.

Checking the Filtering Systems of Target Networks

The ACK flag probe scanning technique also assists in checking the filtering systems of target networks. The attacker sends an ACK probe packet to check the filtering mechanism (Firewalls) of packets employed by the target network.

Sending an ACK probe packet with a random sequence number and getting No Response from the target means that the port is filtered (stateful firewall is present); an RST response from the target means that the port is not filtered (No Firewall is Present).

```
nmap -sA -P0 10.10.0.25
Starting nmap 6.49DETA4 (https://nmap.org) at 2017-07-2108:02 EDT
Nmap scan report for 10.10.0.25
Host is up (0.00076s latency).
All 1000 scanned ports on 10.10.0.25 are unfiltered.

Nmap done: 1 IP address (1 host up) scanned in 1.72 seconds.
```



Scanning Networks
Scanning Techniques

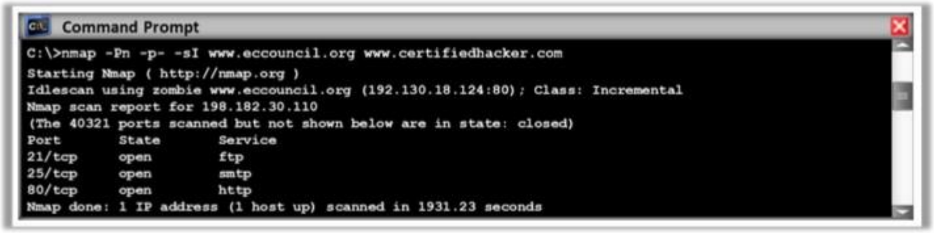
IDLE/IPID Header Scan

- Every IP packet on the Internet has a fragment identification number (IPID); OS increases the IPID for each packet sent, thus, probing an IPID gives an attacker the **number of packets sent** after the last probe
- A machine that receives an **unsolicited SYN|ACK packet** will respond with an RST. An unsolicited RST will be ignored

1. Send SYN + ACK packet to the zombie machine to **probe its IPID number**
2. A zombie machine not expecting an SYN + ACK packet will send **RST packet**, disclosing the IPID. However, always analyse the RST packet from the zombie machine to **extract IPID**
3. Send SYN packet to the **target machine (port 80)** to spoof the IP address of the "zombie"
4. If the port is open, the target will send **SYN+ACK Packet** to the zombie and in response the zombie will send an RST to the target
5. If the port is closed, the target will send an **RST to the zombie** but the zombie will not send anything back
6. Probe the zombie IPID again, IPID increased by **2 will indicate an open port** whereas **1 will indicate a closed port**

IDLE/IPID Header Scan

The IDLE/IPID Header scan is a TCP port scan method that you can use to send a spoofed source address to a computer to find out what services are available. It offers complete blind scanning of a remote host. Most network servers listen on TCP ports, such as web servers on port 80 and mail servers on port 25. Port is considered "open" if an application is listening on the port. One way to determine whether a port is open is to send a "SYN" (session establishment) packet to the port. The target machine will send back a "SYN|ACK" (session request acknowledgment) packet if the port is open, and an "RST" (Reset) packet if the port is closed. A machine that receives an unsolicited SYN|ACK packet will respond with an RST. An unsolicited RST will be ignored. Every IP packet on the Internet has a "fragment identification" number (IPID). OS increases the IPID for each packet sent, thus probing an IPID gives an attacker the number of packets sent since the last probe.



```
C:\>nmap -Pn -p- -sI www.eccouncil.org www.certifiedhacker.com
Starting Nmap ( http://nmap.org )
Idlescan using zombie www.eccouncil.org (192.130.18.124:80); Class: Incremental
Nmap scan report for 198.182.30.110
(The 40321 ports scanned but not shown below are in state: closed)
Port       State      Service
21/tcp     open       ftp
25/tcp     open       smtp
80/tcp     open       http
Nmap done: 1 IP address (1 host up) scanned in 1931.23 seconds
```

FIGURE 3.3: IDLE/IPID Header scan using Nmap

The attacker performs this scan by impersonating another computer through spoofing. The attacker does not send a packet from her/his own IP address; instead, they use another host, often called a "zombie," to scan the remote host and identify any open ports. In this attack, the attacker expects the sequence numbers of the zombie host, and if the remote host checks the IP of the scanning party, the IP of the zombie machine will display.

IDLE Scan

Every IP packet on the Internet has a fragment Internet protocol identification (IPID) number that uniquely identifies fragments of an original IP datagram. As many operating systems simply increase this number for each packet they send, probing for the IPID can tell an attacker how many packets the user sent since the last probe.

- **Step 1**

  The first step in performing an idle scan is to find an appropriate zombie. The zombie that assigns IPID packets incrementally on a global basis is an appropriate or idle zombie to perform the idle scan. The lower the time interval for request/response between the attacker-zombie and the zombie-target, the faster the scan.

  Choose a "Zombie" and Probe for Its Current IP Identification (IPID) Number

  In the first step, you will send the SYN+ACK packet to the zombie machine to probe its IPID number. Here, the reason for sending the SYN+ACK packet is to probe the IPID number but not establish a TCP connection (3-way handshake).

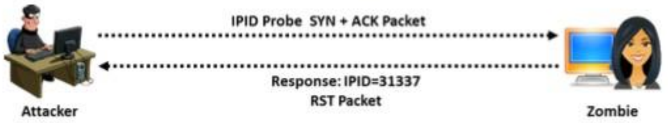


FIGURE 3.4: IDLE scan: Step 1

  As the zombie does not expect a SYN+ACK packet, it will deny the connection by sending back an RST packet. Analyze the RST packet sent by the zombie machine to extract the IPID. In the diagram shown in the slide above, assume the zombie responds with IPID=31337. Assume this IPID is X.

- **Step 2**

  The attacker sends an SYN packet to the target machine on port 80 spoofing the IP address of the Zombie.

  Idle Scan: Step 2.1 (Open Port)

  If the port is open, the target will send the SYN+ACK packet to the zombie (as the IP address was spoofed) to proceed with the 3-way handshake. Since the zombie did not expect a SYN+ACK packet from the target machine, it will respond with a RST packet.

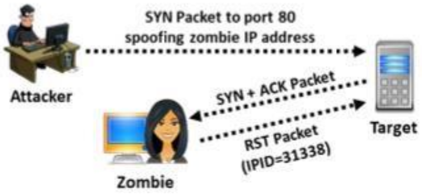


FIGURE 3.5: Port is open

Since every IP packet has a "fragment identification" number, which increases by one for every packet transmission, this time the zombie will use its next available IPID, i.e., 31338 (X + 1).

Idle Scan: Step 2.2 (Closed Port)

Assume that the port on the target is closed. Subsequently, on receiving the SYN packet from the attacker (you), the target will respond with a RST, and the zombie will remain idle without taking any further action.

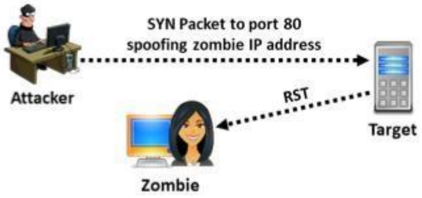


FIGURE 3.6: Port is closed

- **Step 3**

  Now, follow step 1 again to probe the IP ID number.

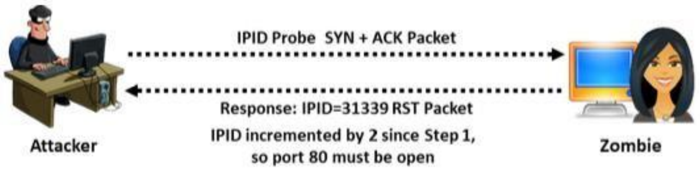
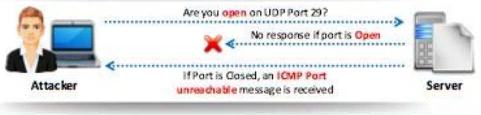

IPID Probe SYN + ACK Packet
Response: IPID=31339 RST Packet
IPID incremented by 2 since Step 1,
so port 80 must be open
Attacker
Zombie

FIGURE 3.7: Idle scan: Step 3

Send a SYN+ACKpacket to the zombie, and it will respond with a RST packet containing the IPID. Assuming that the port on the target was open, and the zombie has already sent a RST packet to the target; then the IPID number has increased by 1. This time the zombie responds with a RST packet to the attacker by using its next IPID, i.e., 31339 (X + 2). Consequently, the IPID has increased by 2, which implies that the port on the target machine was open. Thus, using an Idle scan, an attacker can find out the open ports and services on the target machines by spoofing his /her IP address with a zombie's IP address.

Scanning Networks
Scanning Techniques

UDP Scanning

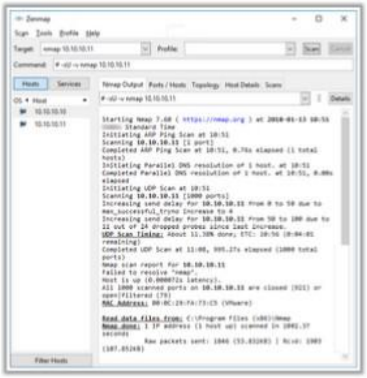

Are you open on UDP Port 29?
No response if port is Open
If Port is Closed, an ICMP Port unreachable message is received
Attacker
Server

UDP Port Open

- There is no **three-way TCP handshake** for UDP scan
- The system does not respond with a message when the **port is open**

UDP Port Closed

- If a UDP packet is sent to a closed port, the system will respond with an **ICMP port unreachable message**
- **Spywares**, **Trojan horses**, and other malicious applications use UDP ports



```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-13 10:51
[illegible] Standard Time
Initiating ARP Ping Scan at 10:51
Scanning 10.10.10.11 [1 port]
Completed ARP Ping Scan at 10:51, 0.76s elapsed (1 total
hosts)
Initiating Parallel DNS resolution of 1 host. at 10:51
Completed Parallel DNS resolution of 1 host. at 10:51, 0.00s
elapsed
Initiating UDP Scan at 10:51
Scanning 10.10.10.11 [1000 ports]
Increasing send delay for 10.10.10.11 from 0 to 50 due to
max_successful_tryno increase to 4
Increasing send delay for 10.10.10.11 from 50 to 100 due to
11 out of 24 dropped probes since last increase.
UDP Scan Timing: About 11.38% done; ETC: 10:56 (0:04:01
remaining)
Completed UDP Scan at 11:08, 995.27s elapsed (1000 total
ports)
Nmap scan report for 10.10.10.11
Failed to resolve "nmap".
Host is up (0.00072s latency).
All 1000 scanned ports on 10.10.10.11 are closed (921) or
open|filtered (79)
MAC Address: 00:0C:29:FA:73:C5 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1002.37
seconds
           Raw packets sent: 1046 (53.832KB) | Rcvd: 1003
(187.852KB)
```

UDP Scanning

UDP Raw ICMP Port Unreachable Scanning

UDP port scanners use the UDP protocol instead of the TCP. There is no three-way handshake for UDP scan. The UDP protocol can be more challenging to use than the TCP scanning because you can send a packet, but you cannot determine whether the host is alive, dead, or filtered. However, you can use one ICMP that checks for open or closed ports. If you send a UDP packet to a port without an application bound to it, the IP stack will return an ICMP port unreachable packet. If any port returns an ICMP error, it will close up thereby, leaving the ports that did not answer if they are open or filtered through the firewall.

This happens because open ports do not have to send an acknowledgement in response to a probe, and closed ports are not even required to send an error packet.

UDP Packets

Source: *https://nmap.org*

When you send a packet to a closed UDP port, most of the hosts send an **ICMP_PORT_UNREACH** error. Thus, you can determine whether a port is NOT open if UDP packets or ICMP errors are not guaranteed to arrive. Thus, UDP scanners of this sort must implement retransmission of packets that appear lost. UDP scanners interpret lost traffic as open ports.

In addition, this scanning technique is slow because it limits the ICMP error message rate as compensation to machines that apply RFC 1812 section 4.3.2.8. A remote host will require access to the raw ICMP socket to distinguish closed from unreachable ports.

UDP RECVFROM () and WRITE () Scanning

Although non-root users cannot read unreachable port errors directly, Linux informs you indirectly when they receive messages.

- **Example:**

  For example, a second write () call to a closed port will usually fail. Various scanners, such as Netcat and Pluvial pscan.c do recvfrom () on non-blocking UDP sockets, and usually return EAGAIN ("Try Again," errno 13) if the ICMP error has not been received, and ECONNREFUSED ("Connection refused," errno 111), if it has. This is the technique used for determining open ports when non-root users use -u (UDP). The root users can also use the -l (lamer UDP scan) options to force this process.

Advantage:

The UDP scan is less informal regarding an open port because there is no overhead of a TCP handshake. However, if ICMP is responding to each unavailable port, the number of total frames can exceed those from a TCP scan. Microsoft-based operating systems do not usually implement any ICMP rate limiting, so this scan operates very efficiently on Windows-based devices.

Disadvantage:

The UDP scan provides port information only. If the additional version of information is needed, the scan must be supplemented with a version detection scan (-sV) or the operating system fingerprinting option (-O).

The UDP scan requires privileged access; hence, this scan option is only available on systems with the appropriate user permissions.

Most networks have huge amounts of TCP traffic; as a result, the efficiency of the UDP scan is lost. The UDP scan will locate these open ports and provide the security manager with valuable information for identifying successful attacker invasions on open UDP ports caused by spyware applications, Trojan horses, and other malicious software.

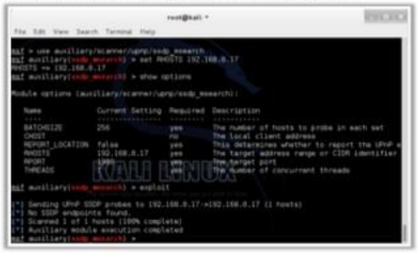
Scanning Networks
Scanning Techniques

SSDP and List Scanning

CEH

SSDP Scanning

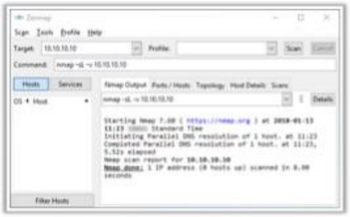
- The Simple Service Discovery Protocol (SSDP) is a network protocol that works in conjunction with the UPnP to detect plug and play devices
- Vulnerabilities in UPnP may allow attackers to launch Buffer overflow or DoS attacks
- Attacker may use UPnP SSDP M-SEARCH information discovery tool to check if the machine is vulnerable to UPnP exploits or not



```
msf > use auxiliary/scanner/upnp/ssdp_msearch
msf auxiliary(ssdp_msearch) > set RHOSTS 192.168.0.17
RHOSTS => 192.168.0.17
msf auxiliary(ssdp_msearch) > show options

Module options (auxiliary/scanner/upnp/ssdp_msearch):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   BATCHSIZE        256              yes       The number of hosts to probe in each set
   CHOST                             no        The local client address
   REPORT_LOCATION  false            yes       This determines whether to report the UPnP e
   RHOSTS           192.168.0.17     yes       The target address range or CIDR identifier
   RPORT            1900             yes       The target port
   THREADS          1                yes       The number of concurrent threads

msf auxiliary(ssdp_msearch) > exploit

[*] Sending UPnP SSDP probes to 192.168.0.17->192.168.0.17 (1 hosts)
[*] No SSDP endpoints found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(ssdp_msearch) >
```

List Scanning

- This type of scan simply generates and prints a list of IPs/Names without actually pinging them
- A reverse DNS resolution is carried out to identify the host names



Target: 10.10.10.10

Command: nmap -sL -v 10.10.10.10

```
nmap -sL -v 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-13
11:23 [illegible] Standard Time
Initiating Parallel DNS resolution of 1 host. at 11:23
Completed Parallel DNS resolution of 1 host. at 11:23,
5.52s elapsed
Nmap scan report for 10.10.10.10
Nmap done: 1 IP address (0 hosts up) scanned in 5.60
seconds
```

List Scanning

In a list scan, the discovery of the active network host is indirect. A list scan simply generates and prints a list of IPs/Names without actually pinging or scanning the hosts. As a result, the list scan shows all IP addresses as “not scanned” (0 hosts up). By default, a reverse DNS resolution is still carried out on each host by Nmap for learning their names.

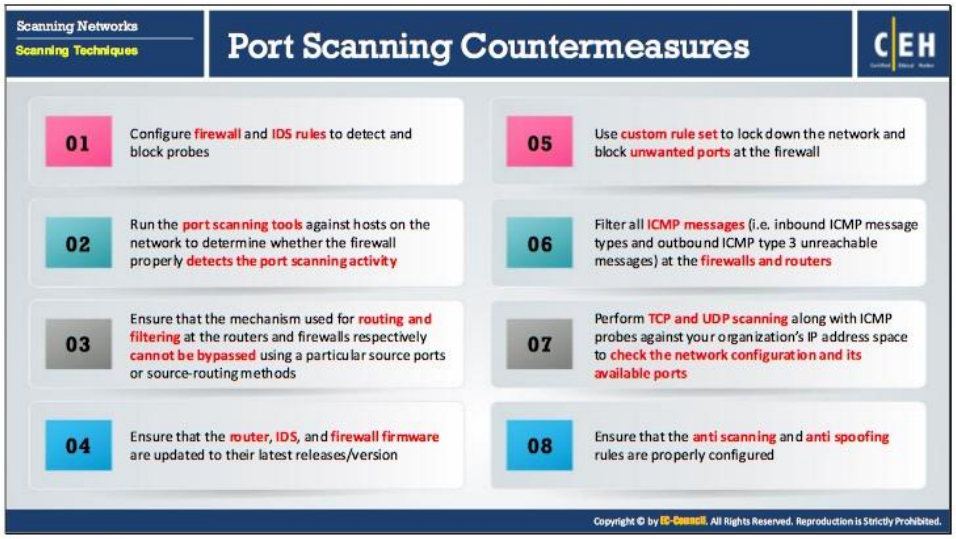
Advantages:

- A list scan can perform a good sanity check.
- The list scan detects incorrectly defined IP addresses on the command line or in an option file. It primarily repairs the detected errors to run any “active” scan.

SSDP Scanning

SSDP (Simple Service Discovery Protocol) is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses. The SSDP service controls communication for the Universal Plug and Play (UPnP) feature. It generally works when the machine is not firewalled; however, it can sometimes work through a firewall. The SSDP service will respond to the query sent over IPv4 or IPv6 broadcast addresses. This response includes information about the Universal Plug and Play (UPnP) feature associated with it. The attacker uses SSDP scanning to detect UPnP vulnerabilities that may allow him/her to launch buffer overflow or DoS attacks.

The attacker may use the UPnP SSDP M-SEARCH information discovery tool to check whether the machine is vulnerable to the UPnP exploits. The UPnP SSDP M-SEARCH information discovery tool gleans information from UPnP-enabled systems as shown in the above slide.



Scanning Networks
Scanning Techniques

Port Scanning Countermeasures

01 Configure firewall and IDS rules to detect and block probes

02 Run the port scanning tools against hosts on the network to determine whether the firewall properly detects the port scanning activity

03 Ensure that the mechanism used for routing and filtering at the routers and firewalls respectively cannot be bypassed using a particular source ports or source-routing methods

04 Ensure that the router, IDS, and firewall firmware are updated to their latest releases/version

05 Use custom rule set to lock down the network and block unwanted ports at the firewall

06 Filter all ICMP messages (i.e. inbound ICMP message types and outbound ICMP type 3 unreachable messages) at the firewalls and routers

07 Perform TCP and UDP scanning along with ICMP probes against your organization's IP address space to check the network configuration and its available ports

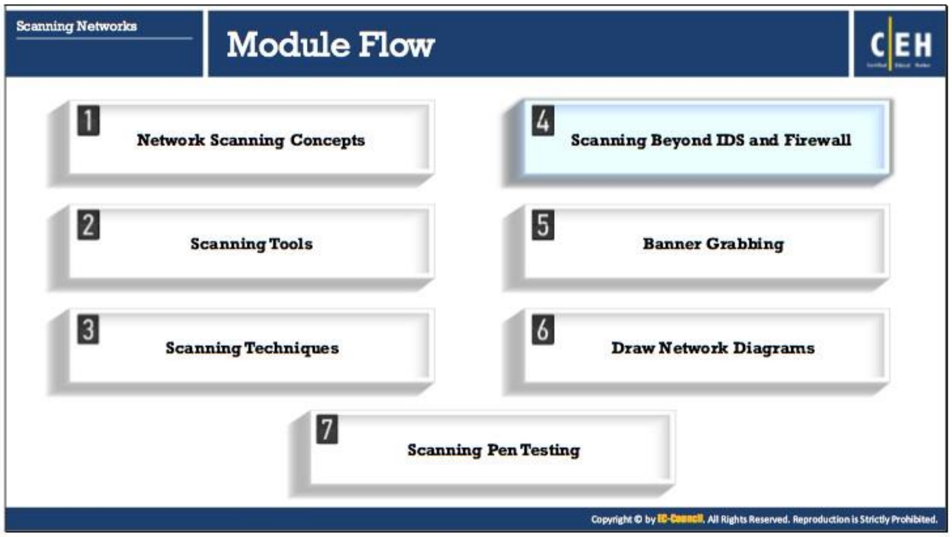
08 Ensure that the anti scanning and anti spoofing rules are properly configured

Port Scanning Countermeasures

As discussed previously, port scanning provides a lot of useful information, such as IP addresses, host names, open ports, and services running on ports to the attacker. Open ports specifically offer an easy means for the attacker to break into the network. But there is nothing to worry about, provided that you secure your system or network against port scanning by applying the following countermeasures:

- Configure firewall and IDS rules to detect and block probes
- The firewall should be capable enough to detect probes sent by the attackers using port scanning tools. It should not allow traffic to pass through it after simply inspecting the TCP header. The firewall should be able to examine the data contained in each packet before allowing the traffic to pass through it.
- Run the port scanning tools against hosts on the network to determine whether the firewall accurately detects the port scanning activity.
- Some firewalls do a better job than others at detecting stealth scans. For example, many firewalls have specific options to detect SYN scans, while others completely ignore the FIN scans.
- Ensure that the router, IDS, and firewall firmware are updated to their latest releases/version.
- Configure commercial firewalls to protect your network against fast port scans and SYN floods. You can run tools such as ports entry to detect and stop port scan attempts on Linux/UNIX systems.

- Hackers use tools such as Nmap and perform OS-detection methods to sniff the details of a remote operating system. Thus, it is important to employ intrusion detection systems in such cases. Snort (http://www.snort.org) is an intrusion detection and prevention technology that can be very useful, mainly because signatures are frequently available from the public authors.
- Keep as few ports open as necessary and filter the rest, as the intruder will try to enter through any open port. Use a custom rule set to lock down the network, block unwanted ports at the firewall, and filter the following ports: 135–159, 256–258, 389, 445, 1080, 1745, and 3268.
- Block inbound ICMP message types and all outbound ICMP type-3 unreachable messages at border routers arranged in front of a company's main firewall.
- Attackers try to perform source routing and send packets to the targets (which may not be reachable via the Internet) by making use of an intermediate host that can interact with the target. Such mechanisms can be adapted for hacking purposes to ensure that your firewall and router can block such source-routing techniques.
- Ensure that the mechanism used for routing and filtering at the routers and firewalls respectively cannot be bypassed using a particular source port or source-routing methods.
- Test your IP address space using TCP and UDP port scans as well as ICMP Probes to determine network configuration and accessible ports.
- Ensure that the anti-scanning and anti-spoofing rules are configured.
- If a commercial firewall is in use, then ensure that:
  - It is patched with the latest updates
  - It has correctly defined antispoofing rules
  - Its Fastmode services are unusable in Check Point Firewall-1 environments



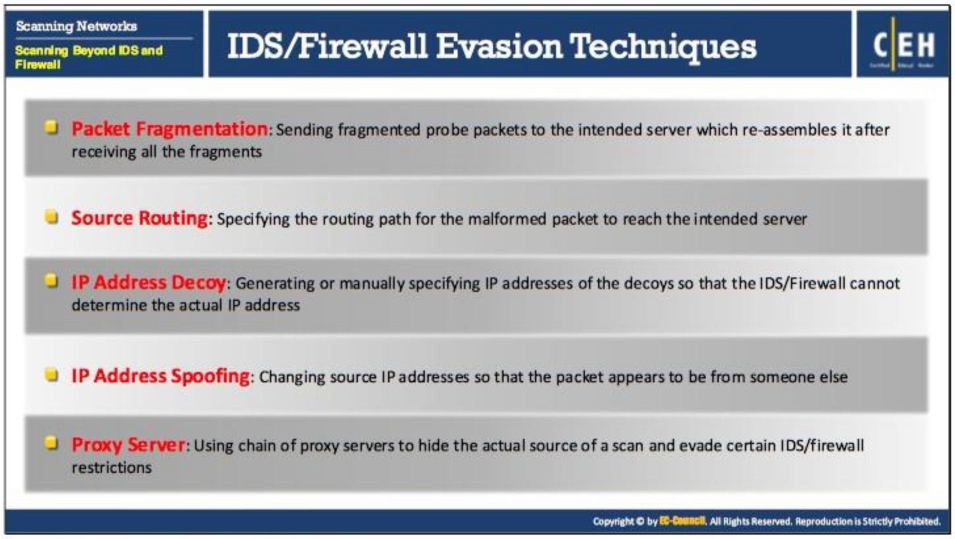
Scanning Networks

Module Flow

1. Network Scanning Concepts
2. Scanning Tools
3. Scanning Techniques
4. Scanning Beyond IDS and Firewall
5. Banner Grabbing
6. Draw Network Diagrams
7. Scanning Pen Testing

Scanning Beyond IDS and Firewall

An Intrusion Detection System (IDS) and firewall are the security mechanism intended to prevent an attacker from accessing a network. But even IDSs and firewalls have some security limitations. Attackers try to launch attacks with the aim of exploiting these limitations. This section highlights various IDS/firewall evasion techniques such as packet fragmentation, source routing, IP address spoofing, etc.



Scanning Networks

Scanning Beyond IDS and Firewall

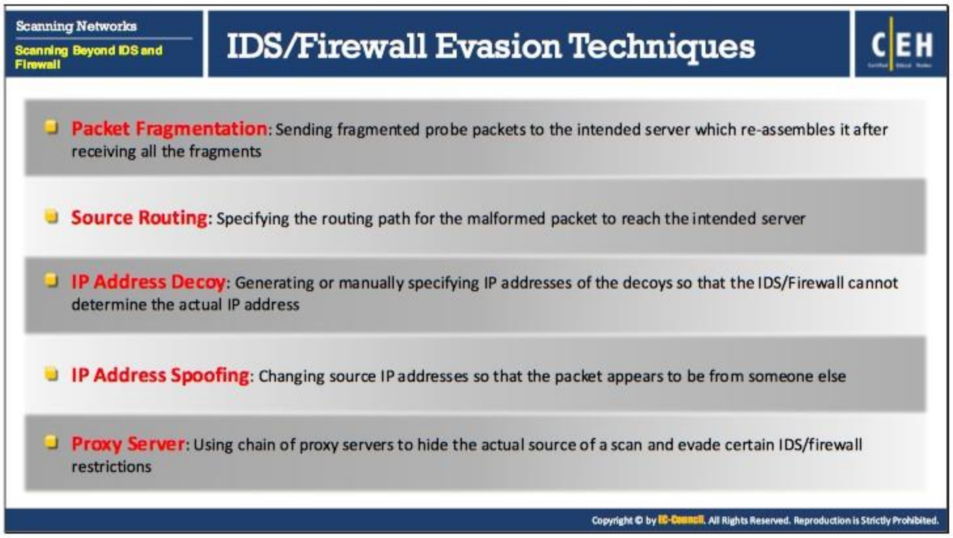
IDS/Firewall Evasion Techniques

- **Packet Fragmentation**: Sending fragmented probe packets to the intended server which re-assembles it after receiving all the fragments
- **Source Routing**: Specifying the routing path for the malformed packet to reach the intended server
- **IP Address Decoy**: Generating or manually specifying IP addresses of the decoys so that the IDS/Firewall cannot determine the actual IP address
- **IP Address Spoofing**: Changing source IP addresses so that the packet appears to be from someone else
- **Proxy Server**: Using chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions

IDS/Firewall Evasion Techniques

Though firewalls and IDSs avoid malicious traffic (packets) from entering a server, attackers manage to send intended packets to the destination server by implementing techniques such as:

- **Packet Fragmentation**: Here, the attacker sends fragmented probe packets to the intended server which re-assembles it after receiving all the fragments.
- **Source Routing**: The attacker specifies the routing path for the malformed packet to reach the intended server.
- **IP Address Decoy**: Generating or manually specifying IP addresses of the decoys so that the IDS/Firewall cannot determine the actual IP address.
- **IP Address Spoofing**: The attacker changes source IP addresses so that the attack appears to be coming in as someone else.
- **Proxy Server**: This is a process in which the attacker uses a chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions.



Scanning Networks

Scanning Beyond IDS and Firewall

IDS/Firewall Evasion Techniques

- **Packet Fragmentation**: Sending fragmented probe packets to the intended server which re-assembles it after receiving all the fragments
- **Source Routing**: Specifying the routing path for the malformed packet to reach the intended server
- **IP Address Decoy**: Generating or manually specifying IP addresses of the decoys so that the IDS/Firewall cannot determine the actual IP address
- **IP Address Spoofing**: Changing source IP addresses so that the packet appears to be from someone else
- **Proxy Server**: Using chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions

IDS/Firewall Evasion Techniques

Though firewalls and IDSs avoid malicious traffic (packets) from entering a server, attackers manage to send intended packets to the destination server by implementing techniques such as:

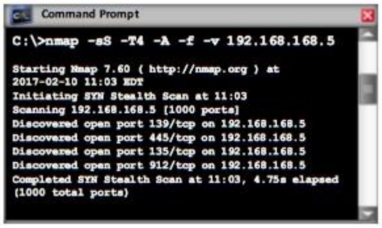
- **Packet Fragmentation**: Here, the attacker sends fragmented probe packets to the intended server which re-assembles it after receiving all the fragments.
- **Source Routing**: The attacker specifies the routing path for the malformed packet to reach the intended server.
- **IP Address Decoy**: Generating or manually specifying IP addresses of the decoys so that the IDS/Firewall cannot determine the actual IP address.
- **IP Address Spoofing**: The attacker changes source IP addresses so that the attack appears to be coming in as someone else.
- **Proxy Server**: This is a process in which the attacker uses a chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions.

Scanning Networks
Scanning Beyond IDS and Firewall

Packet Fragmentation

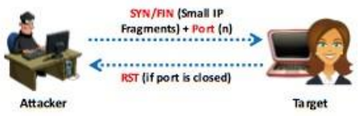
- Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network
- It is not a new scanning method but a modification of the previous techniques

- The TCP header is split into several packets so that the packet filters are not able to detect what the packets intends to do



```
C:\>nmap -sS -T4 -A -f -v 192.168.168.5

Starting Nmap 7.60 ( http://nmap.org ) at
2017-02-10 11:03 EDT
Initiating SYN Stealth Scan at 11:03
Scanning 192.168.168.5 [1000 ports]
Discovered open port 139/tcp on 192.168.168.5
Discovered open port 445/tcp on 192.168.168.5
Discovered open port 135/tcp on 192.168.168.5
Discovered open port 912/tcp on 192.168.168.5
Completed SYN Stealth Scan at 11:03, 4.75s elapsed
(1000 total ports)
```

SYN/FIN Scanning Using IP Fragments



Packet Fragmentation

Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network. When these packets reach a host, IDSs and firewalls behind the host generally queue all of them and process them one by one. However, since this method of processing involves greater CPU consumption as well as network resources, the configuration of most of the IDSs makes it skip fragmented packets during port scans.

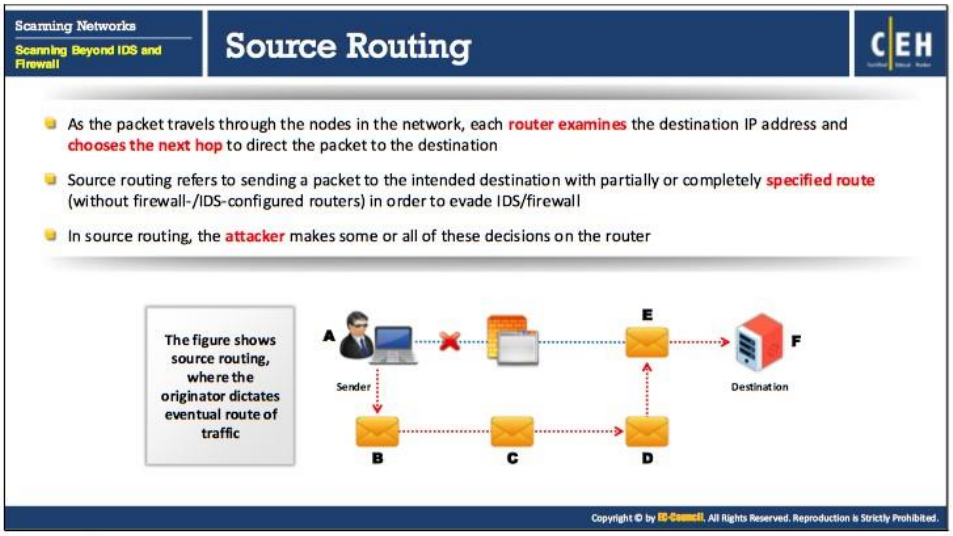
Therefore, attackers use packet fragmentation tools such as Nmap and fragroute to split the probe packet into smaller packets that circumvent the port-scanning techniques employed by intrusion detection systems. Once these fragments reach the destined host, they again re-assemble to form a single packet.

SYN/FIN Scanning Using IP Fragments

SYN/FIN Scanning using IP fragments is not a new scanning method but a modification of the previous techniques. This process of scanning was created to avoid false positives generated by other scans because of a packet filtering device on the target system. The TCP header splits into several packets to evade the packet filter. For any transmission, every TCP header must have the source and destination port for the initial packet (8-octet, 64-bit). The initialized flags in the next packet let the remote host reassemble the packet upon receipt via an Internet protocol module that detects the fragmented data packets using field equivalent values of the source, destination, protocol, and identification.

In this scan, the system splits the TCP header into several fragments and transmits them over the network. However, IP reassembly on the server-side may result in unpredictable and abnormal results, such as fragmentation of IP header data. Some hosts may fail to parse and reassemble the fragmented packets, thus leading to crashes, reboots, or even network device monitoring dumps.

Some firewalls might have rule sets that block IP fragmentation queues in the Kernel (e.g., CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel), although this is not widely implemented because of adverse effects on performance. Since many IDSs use signature-based methods to indicate scanning attempts on IP and/or TCP headers, the use of fragmentation will often evade this type of packet filtering and detection, resulting in a high probability of causing problems on the target network. Attackers use SYN/FIN scanning method with IP fragmentation to evade this type of filtering and detection.



Scanning Networks

Scanning Beyond IDS and Firewall

Source Routing

- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- Source routing refers to sending a packet to the intended destination with partially or completely **specified route** (without firewall-/IDS-configured routers) in order to evade IDS/firewall
- In source routing, the **attacker** makes some or all of these decisions on the router

The figure shows source routing, where the originator dictates eventual route of traffic

Source Routing

An IP datagram contains various fields, including the IP options field, which stores source routing information and includes a list of IP addresses through which the packet travels to its destination. As the packet travels through the nodes in the network, each router examines the destination IP address and chooses the next hop to direct the packet to the destination.

When attackers send malformed packets to a target, these packets hop through various routers and gateways to reach the destination. In some cases, routers in the path might include configured firewalls and IDSs that blocks such packets. To avoid this, attackers enforce a loose or strict source routing mechanism, in which they manipulate the IP address path in the IP options field so that the packet takes the attacker-defined path (without firewall-/IDS-configured routers) to reach the destination, thereby evading firewalls and IDSs.

Scanning Networks

Scanning Beyond IDS and Firewall

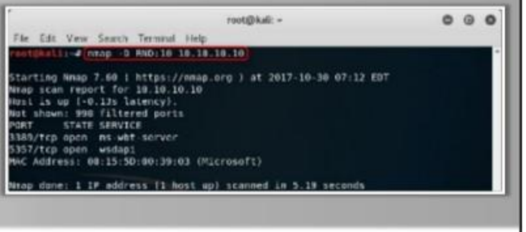
IP Address Decoy

- IP address decoy technique refers to generating or manually specifying IP addresses of the decoys in order to evade IDS/firewall
- It appears to the target that the decoys as well as the host(s) are scanning the network
- This technique makes it difficult for the IDS/firewall to determine which IP address was actually scanning the network and which IP addresses were decoys

Decoy Scanning using Nmap

Nmap has two options for decoy scan:

- `nmap -D RND:10 [target]`
  (Generates a random number of decoys)
- `nmap -D decoy1,decoy2,decoy3,..` etc.
  (Manually specify the IP addresses of the decoys)



```
root@kali:~# nmap -D RND:10 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-10-30 07:12 EDT
Nmap scan report for 10.10.10.10
Host is up (-0.12s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

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

IP Address Decoy

IP address decoy technique refers to generating or manually specifying IP addresses of the decoys in order to evade IDS/firewall. It appears to the target that the decoys, as well as the host(s), are scanning the network. This technique makes it difficult for the IDS/firewall to determine which IP address was actually scanning the network and which IP addresses were decoys.

Nmap scanning tool comes with a built-in scan function called decoy scan, which cloaks a scan with decoys. This technique generates multiple IP addresses to perform a scan function, thus, making it difficult for the target security establishments like IDS, firewall, etc. to identify the original source from the registered logs. The target IDS might report scanning from 5 – 10 IP addresses, however, it cannot differentiate between the actual scanning IP address to the innocent decoy IPs.

You can perform two types of decoy scans using Nmap:

- **nmap -D RND:10 [target]**

  By using this command, Nmap automatically generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IPs.

  Ex. Consider 192.168.100.50 is the target IP address to be scanned. Thus, the Nmap decoy scan command will be:

  ```
  # nmap -D RND:10 192.168.100.50
  ```

- **nmap -D decoy1,decoy2,decoy3,...,ME,... [target]**

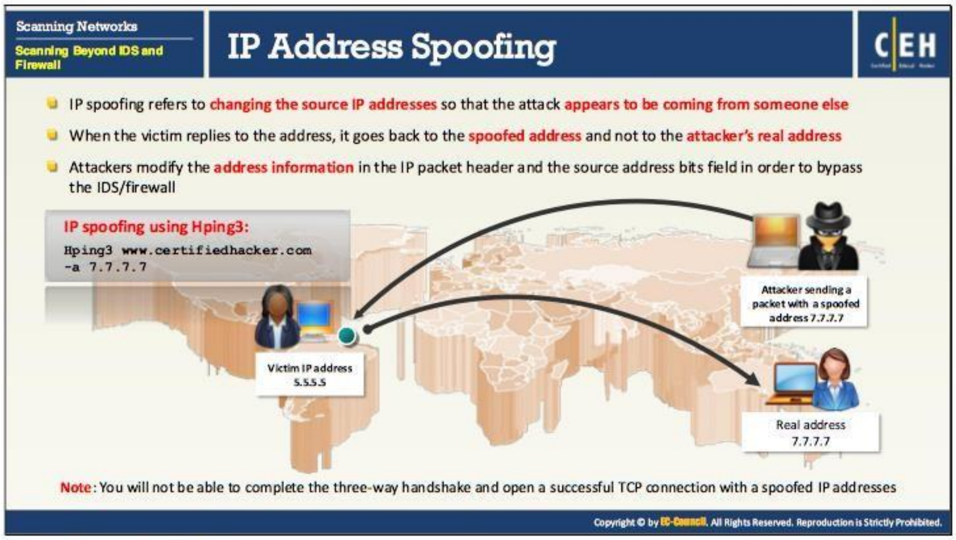
  Using this command, you can manually specify the IP addresses of the decoys to scan the victim's network. Here, you have to separate each decoy IP's with commas (,) and you can optionally use the ME command in order to position your real IP in the decoy list. If you place ME in the 4th position of the command, your real IP will be positioned at 4th position accordingly. This is an optional command, and if you do not mention ME in your scan command, then Nmap will automatically place your real IP in any random position.

Ex. consider 192.168.100.1 is the real source IP and 192.168.100.50 is the target IP address to be scanned. Then the Nmap decoy command will be:

```
#    nmap    -D    192.168.1.10,10.5.1.5,172.1.5.3,192.168.100.1,
3.4.2.1,192.168.111.15,192.168.100.50
```

These decoys can be generated in both initial ping scans like ICMP, SYN, ACK, etc., and during the actual port scanning phase.

IP address decoy is a useful technique for hiding your IP address. However, this cannot be successful if the target employs any of the active mechanisms like router path tracing, response-dropping, etc. Also, using many decoys can slow down the scanning process and affect the accuracy of scan performance.



Scanning Networks

Scanning Beyond IDS and Firewall

IP Address Spoofing

CEH

- IP spoofing refers to **changing the source IP addresses** so that the attack **appears to be coming from someone else**
- When the victim replies to the address, it goes back to the **spoofed address** and not to the **attacker's real address**
- Attackers modify the **address information** in the IP packet header and the source address bits field in order to bypass the IDS/firewall

IP spoofing using Hping3:

```
Hping3 www.certifiedhacker.com
-a 7.7.7.7
```

Attacker sending a packet with a spoofed address 7.7.7.7

Victim IP address 5.5.5.5

Real address 7.7.7.7

Note: You will not be able to complete the three-way handshake and open a successful TCP connection with a spoofed IP addresses

IP Address Spoofing

Most of the firewalls filter packets are based on the source IP address. These firewalls examine the source IP address and decide whether the packet is coming from a legitimate source or an illegitimate source. The IDS filters packets from illegitimate sources. Attackers use the IP spoofing technique to bypass such IDSs/firewalls.

IP address spoofing is a hijacking technique in which an attacker obtains a computer's IP address, alters the packet headers, and sends request packets to a target machine, pretending to be a legitimate host. The packets appear to be sent from the legitimate machine but are actually sent from the attacker's machine, while his/her machine's IP address is concealed. When the victim replies to the address, it goes back to the spoofed address and not to the attacker's real address. Attackers mostly use IP address spoofing to perform DoS attacks.

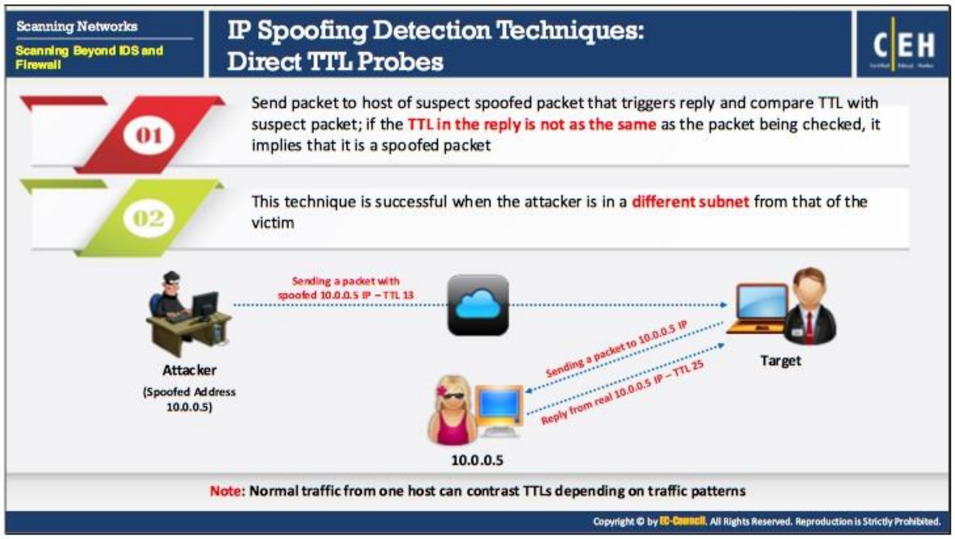
When the attacker sends a connection request to the target host, the target host replies and sends it to the spoofed IP address. When spoofing a nonexistent address, the target replies to a nonexistent system, and then hangs until the session times out, thus consuming the target's resources.

IP spoofing using Hping3:

```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```

You can use Hping3 to perform IP spoofing. The above command helps you to send arbitrary TCP/IP packets to network hosts.

Note: You will not be able to complete the three-way handshake and open a successful TCP connection with a spoofed IP addresses.



Scanning Networks
Scanning Beyond IDS and Firewall

IP Spoofing Detection Techniques: Direct TTL Probes

01 Send packet to host of suspect spoofed packet that triggers reply and compare TTL with suspect packet; if the **TTL in the reply is not as the same** as the packet being checked, it implies that it is a spoofed packet

02 This technique is successful when the attacker is in a **different subnet** from that of the victim

Sending a packet with spoofed 10.0.0.5 IP – TTL 13
Attacker
(Spoofed Address 10.0.0.5)
Target
Sending a packet to 10.0.0.5 IP
Reply from real 10.0.0.5 IP – TTL 25
10.0.0.5

Note: Normal traffic from one host can contrast TTLs depending on traffic patterns

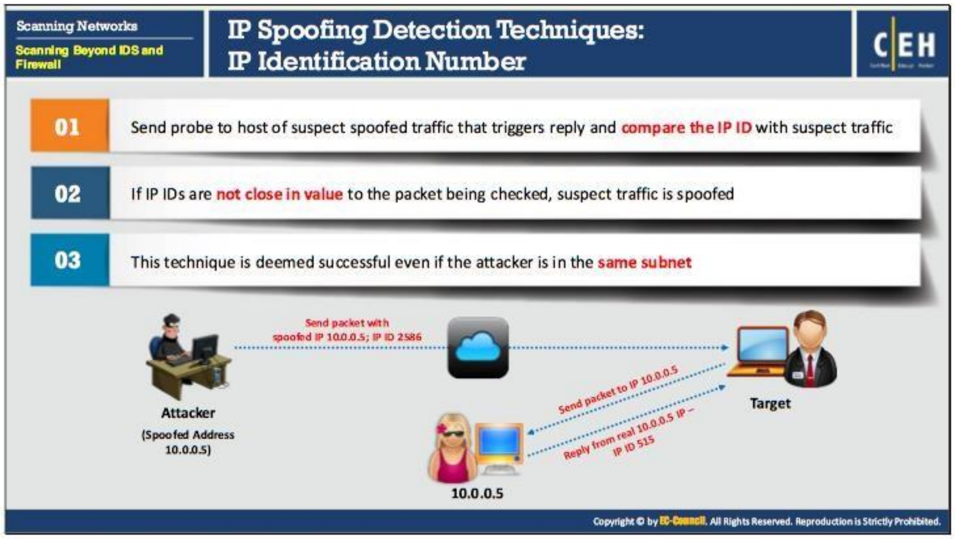
IP Spoofing Detection Techniques

- **Direct TTL Probes**

  In this technique, you initially send a packet (ping request) to the legitimate host and wait for a reply. Check whether the TTL value in the reply matches that of the packet you are checking. Both will have the same TTL if they are using the same protocol. Although the initial TTL values vary according to the protocol used, a few initial TTL values are commonly used: for TCP/UDP, the values are 64 and 128; for ICMP, 128 and 255.

  If the reply is from a different protocol, then you should check the actual hop count to detect the spoofed packets. Deduct the TTL value in the reply from the initial TTL value to determine the hop count. It is a spoofed packet if the reply TTL does not match the TTL of the packet you are checking. It will be very easy to launch an attack if the attacker knows the hop count between the source and the host. In this case, the test result is a false negative. This technique is successful when the attacker is in a different subnet from that of the victim.

  Note: Normal traffic from one host can contrast TTLs depending on traffic patterns



Scanning Networks
Scanning Beyond IDS and Firewall

IP Spoofing Detection Techniques: IP Identification Number

01 Send probe to host of suspect spoofed traffic that triggers reply and **compare the IP ID** with suspect traffic

02 If IP IDs are **not close in value** to the packet being checked, suspect traffic is spoofed

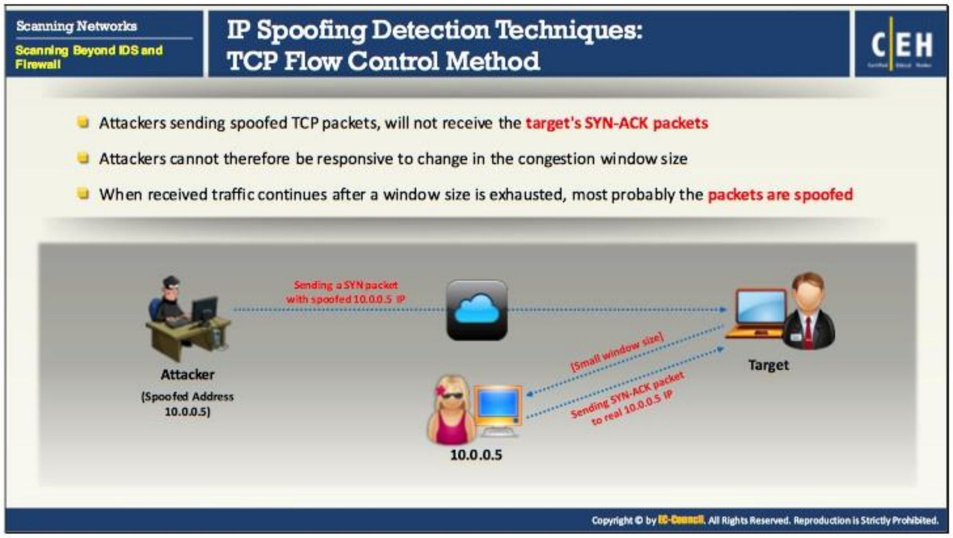
03 This technique is deemed successful even if the attacker is in the **same subnet**

Send packet with spoofed IP 10.0.0.5; IP ID 2586
Attacker
(Spoofed Address 10.0.0.5)
Target
Send packet to IP 10.0.0.5
Reply from real 10.0.0.5 IP – IP ID 515
10.0.0.5

- **IP Identification Number**

  Users can identify spoofed packets by monitoring the IP identification (IPID) number in the IP packet headers. The IPID increases incrementally each time a system sends a packet. Every IP packet on the network has a "fragment identification" number, which is increased by one for every packet transmission. To identify whether a packet is spoofed, send a probe packet to the source IP address of the packet and observe the IPID number in the reply. The IPID value in the response packet must be close to, but slightly higher than the IPID value of the probe packet. The source address of the IP packet is spoofed if the IPID of the response packet is not close to that of the probe packet.

  This method is effective even when both the attacker and the target are on the same subnet.



Scanning Networks
Scanning Beyond IDS and Firewall

IP Spoofing Detection Techniques: TCP Flow Control Method

- Attackers sending spoofed TCP packets, will not receive the **target's SYN-ACK packets**
- Attackers cannot therefore be responsive to change in the congestion window size
- When received traffic continues after a window size is exhausted, most probably the **packets are spoofed**

Sending a SYN packet with spoofed 10.0.0.5 IP
Attacker
(Spoofed Address 10.0.0.5)
(Small window size)
Target
Sending SYN-ACK packet to real 10.0.0.5 IP
10.0.0.5

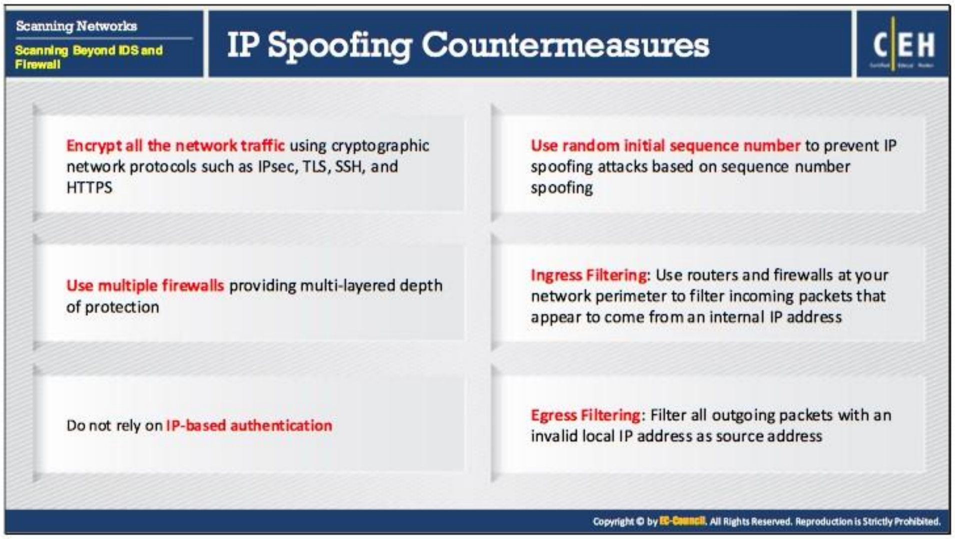
- **TCP Flow Control Method**

  The TCP can optimize the flow control on both the sender and the receiver's end with its algorithm. The algorithm accomplishes the flow control using the sliding window principle. The user can control the flow of IP packets by the window size field in the TCP header. This field represents the maximum amount of data that the recipient can receive and the maximum amount of data that the sender can transmit without acknowledgement. Thus, this field helps us to control data flow. The sender should stop sending data whenever the window size is set to zero.

  In general flow control, the sender should stop sending data once the initial window size is exhausted. The attacker who is unaware of the ACK packet containing window size information might continue to send data to the victim. If the victim receives data packets beyond the window size, they are spoofed packets. For effective flow control method and early detection of spoofing, the initial window size must be very small.

  Most spoofing attacks occur during the handshake, as it is challenging to build multiple spoofing replies with the correct sequence number. Therefore, apply the flow control spoofed packet detection at the handshake. In a TCP handshake, the host sending the initial SYN packet waits for SYN-ACK before sending the ACK packet. To check whether you are getting the SYN request from a genuine client or a spoofed one, set the SYN-ACK to zero. If the sender sends an ACK with any data, it means that the sender is the spoofed one. This is because when the SYN-ACK is set to zero, the sender must respond to it only with the ACK packet, without additional data.

  Attackers sending spoofed TCP packets will not receive the target's SYN-ACK packets. Attackers cannot be responsive to change in the congestion window size. When received traffic continues after a window size is exhausted, most probably the packets are spoofed.



Scanning Networks
Scanning Beyond IDS and Firewall

IP Spoofing Countermeasures

Encrypt all the network traffic using cryptographic network protocols such as IPsec, TLS, SSH, and HTTPS

Use random initial sequence number to prevent IP spoofing attacks based on sequence number spoofing

Use multiple firewalls providing multi-layered depth of protection

Ingress Filtering: Use routers and firewalls at your network perimeter to filter incoming packets that appear to come from an internal IP address

Do not rely on **IP-based authentication**

Egress Filtering: Filter all outgoing packets with an invalid local IP address as source address

IP Spoofing Countermeasures

In ethical hacking, the ethical hacker, also known as the "pen tester", has to perform an additional task that a normal hacker does not follow (i.e., applying countermeasures to the respective vulnerabilities determined through hacking). This is essential because knowing security loopholes in your network is worthless unless you take measures to protect them against real hackers. As mentioned previously, IP spoofing is one of the techniques that a hacker employs to break into the target network. Therefore, to protect your network from external hackers, you should apply IP spoofing countermeasures to your network security settings. The following are a few IP spoofing countermeasures that you can apply:

- **Avoid trust relationships**

  Do not rely on IP-based authentication. Attackers may spoof themselves as a trusted host and send malicious packets to you. If you accept these packets, under the assumption that they are "clean" because they are from your trusted host, the malicious code will infect your system. Therefore, it is advisable to test all packets, even when they come from one of your trusted hosts. You can avoid this problem by implementing password authentication along with trust-relationship-based authentication.

- **Use firewalls and filtering mechanisms**

  As stated above, you should filter all the incoming and outgoing packets to avoid attacks and sensitive information loss. A firewall can keep malicious packets from entering your private network and prevent severe data loss. You can use access control lists (ACLs) to block unauthorized access. At the same time, there is the possibility of an insider attack. Inside attackers could send sensitive information about your business to your competitors, which could lead to monetary loss and other issues. Another risk of outgoing packets is that an attacker will succeed in installing a malicious sniffing program running in hidden mode on your network. These programs gather and send all your network information to the attacker without any notification after filtering the outgoing packets. Therefore, you should assign the same importance to the scanning of outgoing packets as you would do with incoming packets.

- **Use random initial sequence numbers**

  Most of the devices choose their ISN based on timed counters. This makes the ISNs predictable, as it is easy for a malicious person to determine the concept of generating the ISN. An attacker can determine the ISN of the next TCP connection by analyzing the ISN of the current session or connection. If the attacker can predict the ISN, then he/she can make a malicious connection to the server and sniff out your network traffic. To avoid this risk, use random initial sequence numbers.

- **Ingress filtering**

  Ingress filtering prohibits spoofed traffic from entering the Internet. It is applied on routers because it enhances the functionality of the routers and blocks spoofed traffic. Configuring and using access control lists (ACLs) that drop packets with the source address outside the defined range is one method to implement ingress filtering.

- **Egress filtering**

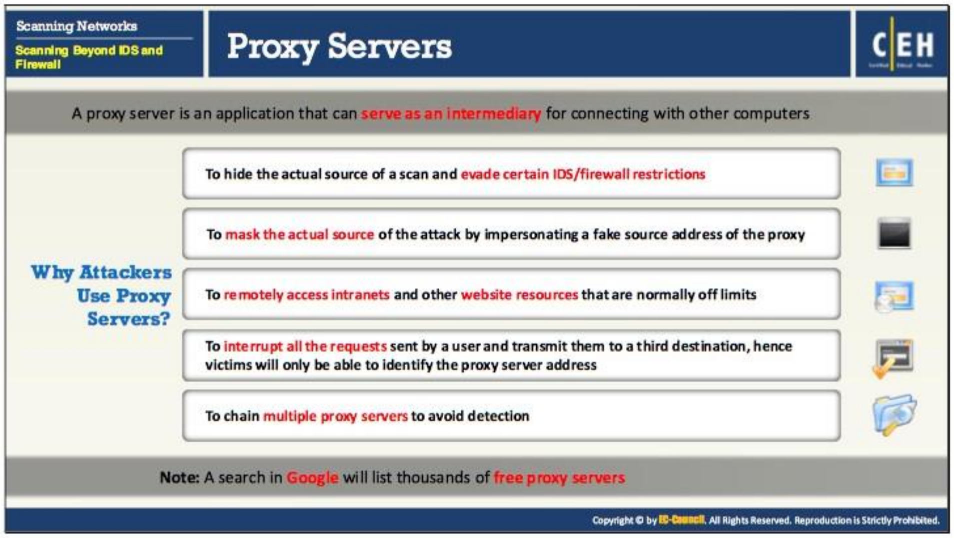
  Egress filtering refers to a practice that aims at IP spoofing prevention by blocking the outgoing packets with a source address that is not inside.

- **Use encryption**

  If you want to attain maximum network security, then use strong encryption for all the traffic placed onto the transmission media, without considering its type and location. This is the best prevention against IP spoofing attacks. Attackers tend to focus on easy-to-compromise targets. If an attacker wants to break into the encrypted network, he or she has to face decrypting a whole slew of encrypted packets, which is a difficult task. Therefore, the attacker is likely to move on to try and find another target that is easy to compromise or will simply abort the attempt. Use the latest encryption algorithms that provide strong security.

- **SYN flooding countermeasures**

  Countermeasures against SYN flooding attacks can also help you to avoid IP spoofing attacks.



Scanning Networks
Scanning Beyond IDS and Firewall

Proxy Servers

A proxy server is an application that can serve as an intermediary for connecting with other computers

Why Attackers Use Proxy Servers?

- To hide the actual source of a scan and evade certain IDS/firewall restrictions
- To mask the actual source of the attack by impersonating a fake source address of the proxy
- To remotely access intranets and other website resources that are normally off limits
- To interrupt all the requests sent by a user and transmit them to a third destination, hence victims will only be able to identify the proxy server address
- To chain multiple proxy servers to avoid detection

Note: A search in Google will list thousands of free proxy servers

Proxy Servers

A proxy server is an application that can serve as an intermediary for connecting with other computers.

A proxy server is used:

- As a firewall and to protect the local network from external attacks.
- As an IP address multiplexer, allowing a number of computers to connect to the Internet when you have only one IP address (NAT/PAT).
- To anonymize web surfing (to some extent).
- To extract unwanted content, such as ads or "unsuitable" material (using specialized proxy servers).
- To provide some protection against hacking attacks.
- To save bandwidth

How does a proxy server work?

Initially, when you use a proxy to request a particular web page on an actual server, the proxy server receives it. The proxy server then sends your request to the actual server on behalf of your request. It mediates between you and the actual server to transmit and respond to the request, as shown in the figure below.

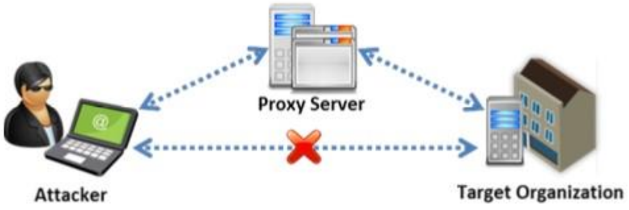


FIGURE 3.8: Attacker using a proxy server for connecting to the target

In this process, the proxy receives the communication between the client and the destination application. To take advantage of a proxy server, an attacker must configure client programs so that they can send their requests to the proxy server instead of the final destination.

Why Attackers Use Proxy Servers?

It is easier for an attacker to attack or hack a particular system than to conceal the attack source. Therefore, the primary challenge for an attacker is to hide his/her identity so that he/she cannot be traced. Thus, the attacker uses the proxy server to avoid detection of attack evidence by masking his/her IP address. When the attacker uses a proxy to connect to the target system, the server logs will record the proxy's source address, rather than the attacker's source address.

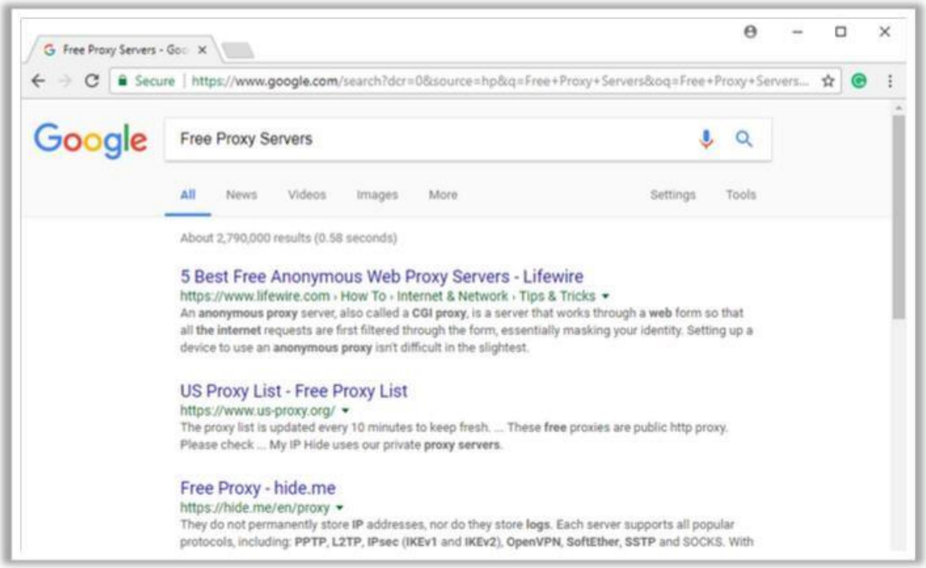
Proxy sites help the attacker to browse the Internet anonymously and unblock blocked sites (i.e., evade firewall restrictions). By using these sites, the attacker can surf restricted sites anonymously, without using the source IP address.

Attackers use proxy servers:

- To hide the actual source of a scan and evade certain IDS/firewall restrictions.
- To hide the source IP address so that they can hack without any legal corollary.
- To mask the actual source of the attack by impersonating a fake source address of the proxy.
- To remotely access intranets and other website resources that are normally off limits.
- To interrupt all the requests sent by a user and transmit them to a third destination, hence, victims will only be able to identify the proxy server address.
- To chain multiple proxy servers to avoid detection.

Free Proxy Servers

There are some free proxy servers available on the Internet that can help you to access restricted sites without revealing your IP address. In the **Google** search engine, type **"Free Proxy Servers"** to see a listing of them. Select one from the list to download, and browse anonymously without revealing your legitimate IP address.


Free Proxy Servers - Goo
Secure
https://www.google.com/search?dcr=0&source=hp&q=Free+Proxy+Servers&oq=Free+Proxy+Servers...
Google
Free Proxy Servers
All
News
Videos
Images
More
Settings
Tools

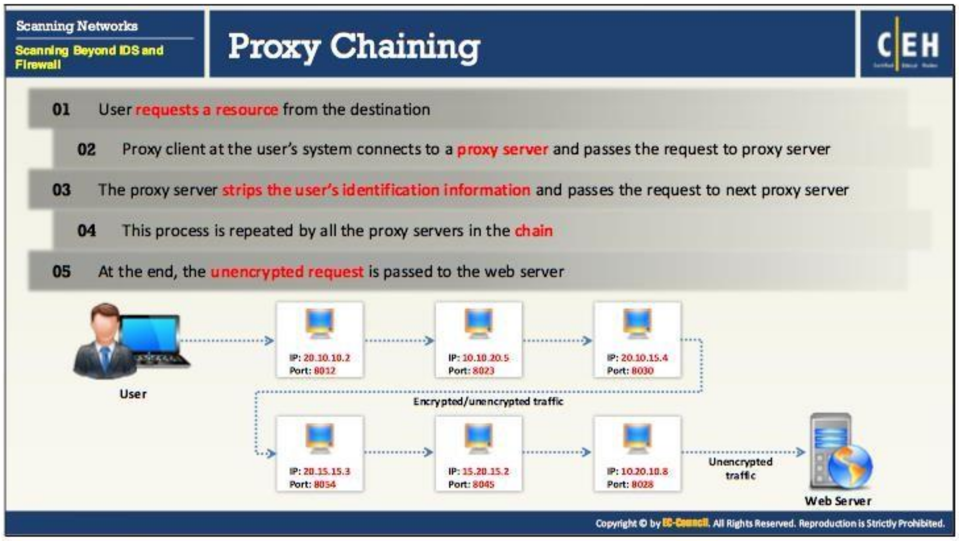
About 2,790,000 results (0.58 seconds)

5 Best Free Anonymous Web Proxy Servers - Lifewire
https://www.lifewire.com › How To › Internet & Network › Tips & Tricks ▾
An **anonymous proxy** server, also called a **CGI proxy**, is a server that works through a **web** form so that all **the internet** requests are first filtered through the form, essentially masking your identity. Setting up a device to use an **anonymous proxy** isn't difficult in the slightest.

US Proxy List - Free Proxy List
https://www.us-proxy.org/ ▾
The proxy list is updated every 10 minutes to keep fresh. ... These **free** proxies are public http proxy. Please check ... My IP Hide uses our private **proxy servers**.

Free Proxy - hide.me
https://hide.me/en/proxy ▾
They do not permanently store **IP** addresses, nor do they store **logs**. Each server supports all popular protocols, including: **PPTP**, **L2TP**, **IPsec** (**IKEv1** and **IKEv2**), **OpenVPN**, **SoftEther**, **SSTP** and SOCKS. With

FIGURE 3.9: Free Proxy Servers



Scanning Networks
Scanning Beyond IDS and Firewall

Proxy Chaining

01 User requests a resource from the destination

02 Proxy client at the user's system connects to a proxy server and passes the request to proxy server

03 The proxy server strips the user's identification information and passes the request to next proxy server

04 This process is repeated by all the proxy servers in the chain

05 At the end, the unencrypted request is passed to the web server

Proxy Chaining

Proxy chaining helps an attacker to increase his/her Internet anonymity. Internet anonymity depends on the number of proxies used for fetching the target application; the larger the number of proxy servers used, the greater the attacker's anonymity.

Proxy chaining process is described as below:

- User requests a resource from the destination.
- A proxy client at the user's system connects to a proxy server and passes the request to the proxy server.
- The proxy server strips the user's identification information and passes the request to the next proxy server.
- This process is repeated by all the proxy servers in the chain.
- In the end, the unencrypted request is passed to the web server.

Scanning Networks

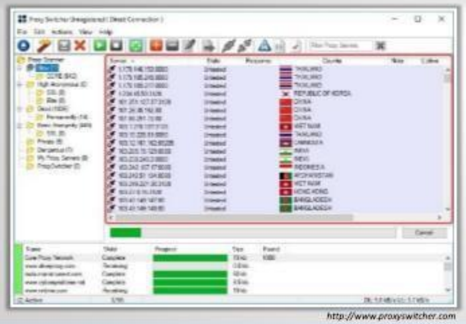
Scanning Beyond IDS and Firewall

Proxy Tools: Proxy Switcher and Proxy Workbench

CEH

Proxy Switcher

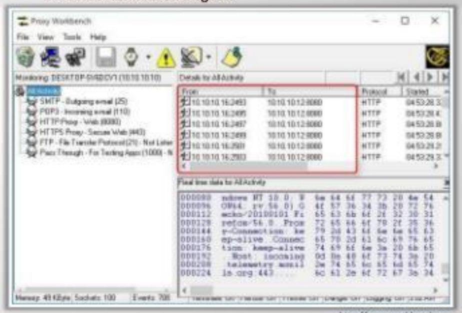
- Proxy Switcher allows you to surf anonymously on the Internet without disclosing your IP address



http://www.proxyswitcher.com

Proxy Workbench

- Proxy Workbench is a proxy server that displays data passing through it in real time, allows you to drill into a particular TCP/IP connection, view their history, save the data to a file, and view the socket connection diagram


Proxy Workbench
File View Tools Help
SMTP - Outgoing e-mail (25)
POP3 - Incoming e-mail (110)
HTTP Proxy - Web (8080)
HTTPS Proxy - Secure Web (443)
FTP - File Transfer Protocol (21) - Not Listening
Pass Through - For Testing Apps (1000)
Details for All Activity
Real time data for All Activity

http://proxyworkbench.com

Proxy Tools

Proxy tools are intended to allow users to surf the Internet anonymously by keeping their IP hidden through a chain of SOCKS or HTTP proxies. These tools can also act as an HTTP, mail, FTP, SOCKS, news, telnet, and HTTPS proxy server.

- **Proxy Switcher**

  Source: *http://www.proxyswitcher.com*

  Proxy Switcher allows you to surf the Internet anonymously without disclosing your IP address. It also helps you to access various blocked sites in the organization. It avoids all sorts of limitations imposed by target sites.

  Features:

  - Hides your IP address from the websites you visit
  - Penetrate bans and blocks on forums, classifieds, and download sites (e.g., RapidShare)
  - Automatic proxy server switching for improved anonymous surfing
  - Full support for password-protected servers.
  - Full support of Socks v5 and Elite servers

- **Proxy Workbench**

  Source: *http://proxyworkbench.com*

  Proxy Workbench is a proxy server that displays the data passing through it in real time and allows you to drill into particular TCP/IP connections, view their history, save the data to a file, and view the socket connection diagram. The socket connection diagram is an animated graphical history of all of the events that took place on the socket connection.

Features:

- Connection failure simulation strategies allow you to simulate:
  - Slow or asymmetric Internet connections (bandwidth throttling)
  - Servers that are underpowered, overloaded or under attack (connection refusal)
  - Intermittent connections (connection termination)
  - Disconnected network cables (connection dangling)
  - Data floods and droughts
- Native ability to analyze HTTP, FTP, SOAP, HTTPS (secure sockets), POP3, Web services, and "pass-through" communications
- The data can be presented in 5 formats: ASCII, hexadecimal, octal, decimal, or binary

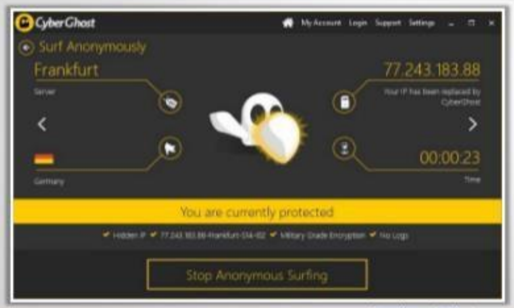
Scanning Networks

Scanning Beyond IDS and Firewall

Proxy Tools

CyberGhost VPN

- CyberGhost allows you to protect your online privacy, surf anonymously, and access blocked or censored contents
- It hides your IP and replaces it with one of your choice, allowing you to surf anonymously



https://www.cyberghostvpn.com

- **Tor** https://www.torproject.org
- **Burp Suite** https://www.portswigger.net
- **Hotspot Shield** https://www.hotspotshield.com
- **Proxifier** https://www.proxifier.com
- **Charles** https://www.charlesproxy.com

- **CyberGhost**

  Source: *https://www.cyberghostvpn.com*

  CyberGhost VPN allows users to protect their online privacy, surf anonymously, and access blocked or censored content.

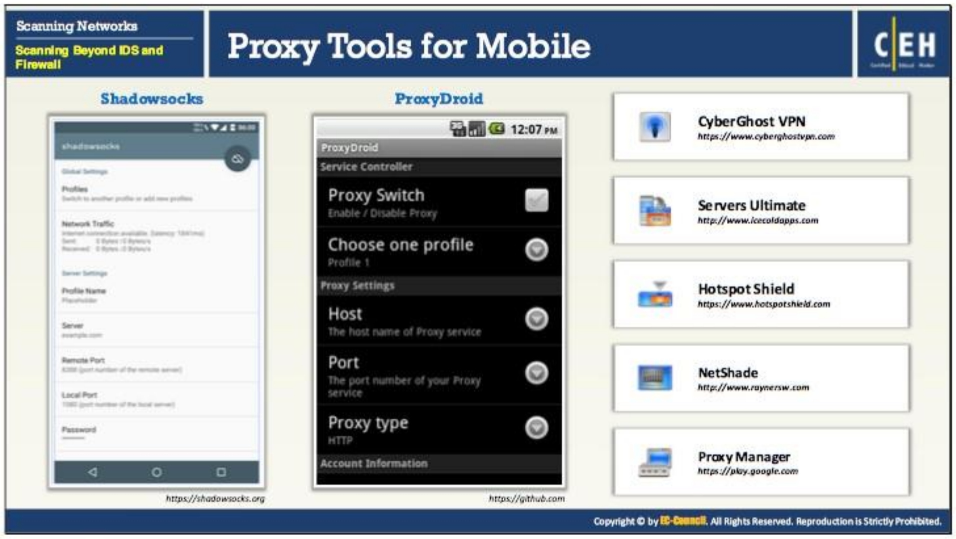
  Features:

  - **Privacy**: This hides your IP and replaces it with one of your choices. This way, you can surf anonymously.
  - **Security**: It encrypts your connection and does not keep logs, thus securing data.
  - **Freedom**: This allows access to censored or geo-restricted content.

In addition to the proxy tools mentioned earlier, there are many other proxy tools intended to allow users to surf the Internet anonymously. Some additional proxy tools are discussed below:

- Tor (*https://www.torproject.org*)
- Burp Suite (*https://www.portswigger.net*)
- Hotspot Shield (*https://www.hotspotshield.com*)
- Proxifier (*https://www.proxifier.com*)
- Charles (*http://www.charlesproxy.com*)
- Fiddler (*http://www.telerik.com*)
- Protoport Proxy Chain (*http://www.protoport.com*)
- ProxyCap (*http://www.proxycap.com*)

- CCProxy (*http://www.youngzsoft.net*)
- Privoxy (*https://www.privoxy.org*)
- SocksChain (*http://ufasoft.com*)



Scanning Networks

Scanning Beyond IDS and Firewall

Proxy Tools for Mobile

Shadowsocks

shadowsocks

Global Settings

Profiles
Switch to another profile or add new profiles

Network Traffic
Internet connection available. (latency: 1841ms)
Sent: 0 Bytes (0 Bytes/s)
Received: 0 Bytes (0 Bytes/s)

Server Settings

Profile Name
Placeholder

Server
example.com

Remote Port
8388 (port number of the remote server)

Local Port
1080 (port number of the local server)

Password

https://shadowsocks.org

ProxyDroid

12:07 PM

ProxyDroid

Service Controller

Proxy Switch
Enable / Disable Proxy

Choose one profile
Profile 1

Proxy Settings

Host
The host name of Proxy service

Port
The port number of your Proxy service

Proxy type
HTTP

Account Information

https://github.com

CyberGhost VPN
https://www.cyberghostvpn.com

Servers Ultimate
http://www.icecoldapps.com

Hotspot Shield
https://www.hotspotshield.com

NetShade
http://www.raynersw.com

Proxy Manager
https://play.google.com

Proxy Tools for Mobile

- **Shadowsocks**

  Source: *https://shadowsocks.org*

  Shadowsocks is a high-performance, cross-platform secured socks5 proxy. It helps one to surf the internet privately and securely.

  Features

  - Bleeding edge techniques with Asynchronous I/O and Event-driven programming
  - Low resource consumption, suitable for low-end boxes and embedded devices
  - Available on multiple platforms, including PC, MAC, Mobile (Android and iOS), and Routers (OpenWRT)
  - Open source implementations in python, node.js, golang, C#, and pure C

- **ProxyDroid**

  Source: *https://github.com*

  ProxyDroid is an app that can help you to set the proxy (http / socks4 / socks5) on your android devices.

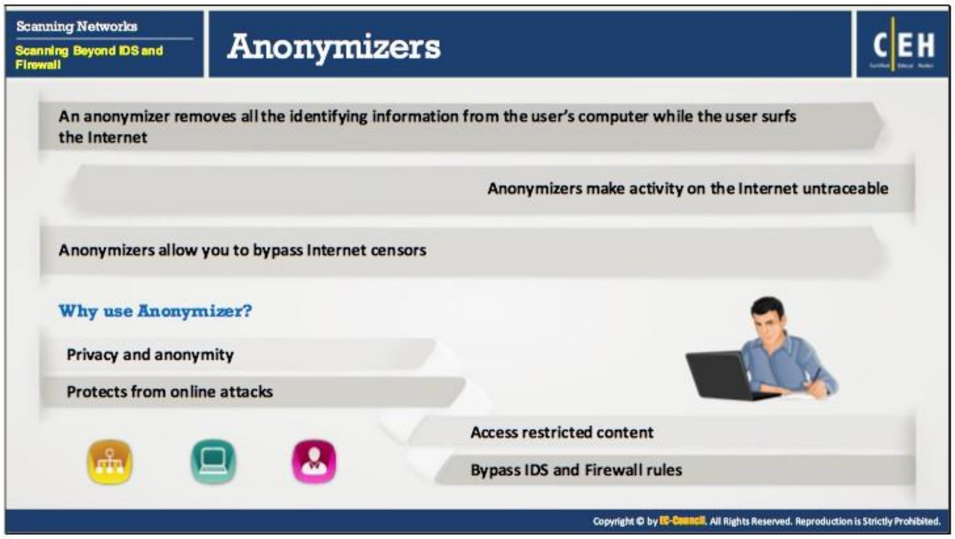
  Features

  - Supports HTTP / HTTPS / SOCKS4 / SOCKS5 proxy
  - Supports basic / NTLM / NTLMv2 authentication methods
  - Individual proxy for only one or several apps

- Supports multiple profiles
- Binds configuration to WIFI's SSID / Mobile Network (2G / 3G)
- Widgets for quickly switching on/off proxy
- Low battery and memory consumption (written in C and compiled as native binary)
- Bypass custom IP address
- PAC file support (only basic support, thanks to Rhino)
- DNS proxy for guys behind the firewall that disallows to resolve external addresses

Some of the proxy tools for mobile include:

- CyberGhost VPN (*https://www.cyberghostvpn.com*)
- Servers Ultimate (*http://www.icecoldapps.com*)
- Hotspot Shield (*https://www.hotspotshield.com*)
- NetShade (*http://www.raynersw.com*)
- Proxy Manager (*https://play.google.com*)



Scanning Networks

Scanning Beyond IDS and Firewall

Anonymizers

An anonymizer removes all the identifying information from the user's computer while the user surfs the Internet

Anonymizers make activity on the Internet untraceable

Anonymizers allow you to bypass Internet censors

Why use Anonymizer?

Privacy and anonymity

Protects from online attacks

Access restricted content

Bypass IDS and Firewall rules

Anonymizers

An anonymizer is an intermediate server placed between you as the end user and the website to accesses the website on your behalf and make your web surfing untraceable. Anonymizers allow you to bypass Internet censors. An anonymizer eliminates all the identifying information (IP address) from your system while you are surfing the Internet, thereby ensuring privacy. Most anonymizers can anonymize the web (HTTP:), file transfer protocol (FTP :), and gopher (gopher :) Internet services.

To visit a page anonymously, you can visit your preferred anonymizer site, and enter the name of the target website in the Anonymization field. Alternately, you can set your browser home page to point to an anonymizer, in order to anonymize subsequent web access. Apart from this, you can choose to anonymously provide passwords and other information to sites without revealing any additional information, such as your IP address. Crackers may configure an anonymizer as a permanent proxy server by making the site name the setting for the HTTP, FTP, Gopher, and other proxy options in their applications configuration menu, thereby cloaking their malicious activities.

Why Use an Anonymizer?

The reasons for using anonymizers include:

- **Ensuring privacy**: Protect your identity by making your web navigation activities untraceable. Your privacy is maintained until and unless you disclose your personal information on the web, for example, by filling out forms.
- **Accessing government-restricted content**: Most governments prevent their citizens from accessing certain websites or content deemed inappropriate or containing sensitive

information. However, these sites can still be accessed using an anonymizer located outside the target country.

- **Protection against online attacks**: An anonymizer can protect you from all instances of online pharming attacks by routing all customer Internet traffic via its protected DNS server.
- **Bypassing IDS and firewall rules**: Firewalls are typically bypassed by employees or students accessing websites that they are not supposed to access. An anonymizer service gets around your organization's firewall by setting up a connection between your computer and the anonymizer service. By so doing, firewalls see only the connection from your computer to the anonymizer's web address. The anonymizer will then connect to any website (e.g., Twitter) with the help of an Internet connection, and then direct the content back to you. To your organization, your system appears to be simply connected to the anonymizer's web address, but not to the actual site to which you have browsed.

In addition to protecting users' identities, anonymizers can also be used to attack a website without being traced.

Types of Anonymizers

An anonymizer is a service through which one can hide their identity when using certain Internet services. It encrypts the data from your computer to the Internet service provider. Anonymizers are of two basic types; Networked anonymizers and Single-point anonymizers.

- **Networked Anonymizers**

  A networked anonymizer first transfers your information through a network of Internet-connected computers before passing it on to the website. Because the information passes through several Internet computers, it becomes more cumbersome for anyone trying to track your information to establish the connection between you and the anonymizer.

  Example: If you want to visit any web page, you have to make a request. The request will first pass through A, B, and C Internet computers before going to the website.

  Advantage: Complication of the communications makes traffic analysis complex.

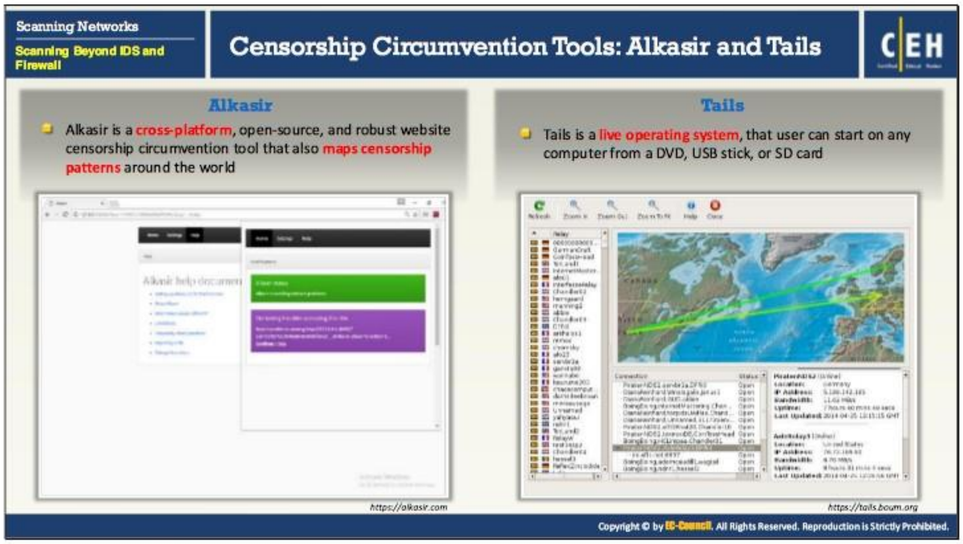
  Disadvantage: Any multi-node network communication incurs some degree of risk of compromising confidentiality at each node.

- **Single-Point Anonymizers**

  Single-point anonymizers first transfer your information through a website before sending it to the target website, and then pass back information gathered from the targeted website, to you via the website to protect your identity.

  Advantage: Arms-length communication protects IP address and related identifying information.

  Disadvantage: It offers less resistance to sophisticated traffic analysis.



Scanning Networks

Scanning Beyond IDS and Firewall

Censorship Circumvention Tools: Alkasir and Tails

Alkasir

- Alkasir is a cross-platform, open-source, and robust website censorship circumvention tool that also maps censorship patterns around the world

https://alkasir.com

Tails

- Tails is a live operating system, that user can start on any computer from a DVD, USB stick, or SD card

https://tails.boum.org

Censorship Circumvention Tools

- **Alkasir**

  Source: *https://alkasir.com*

  Alkasir is a cross-platform, open-source and robust website censorship circumvention tool that also maps censorship patterns around the world.

  Features

  - Optimized bandwidth usage
  - Keeps you informed about links that are still blocked and links that are not blocked.
  - Less prone to an interception in day-to-day access.

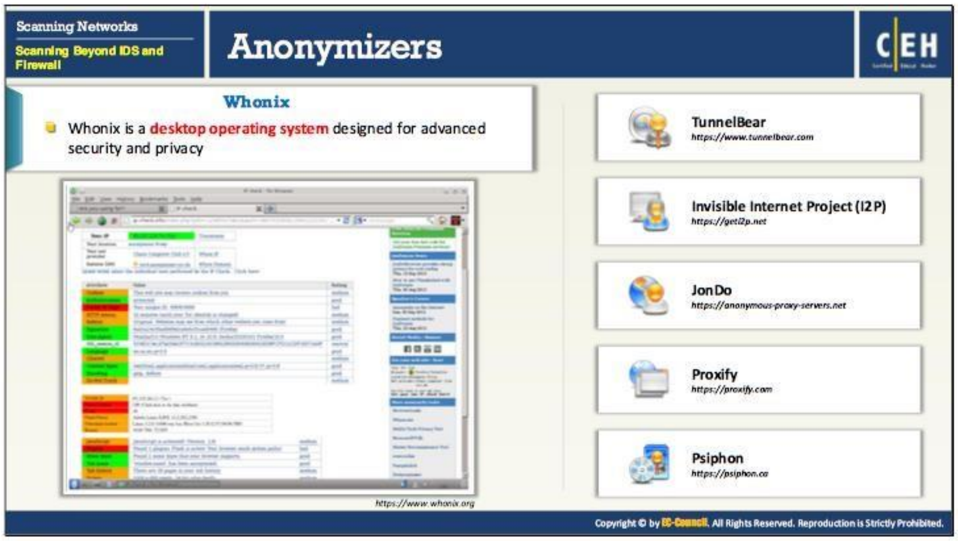
- **Tails**

  Source: *https://tails.boum.org*

  Tails is a live operating system that users can start on any computer from a DVD, USB stick, or SD card.

  Features

  - Use the Internet anonymously and circumvent censorship
  - Leave no trace on the computer
  - Use state-of-the-art cryptographic tools to encrypt files, emails, and instant messaging



Scanning Networks

Scanning Beyond IDS and Firewall

Anonymizers

Whonix

- Whonix is a **desktop operating system** designed for advanced security and privacy

https://www.whonix.org

TunnelBear
https://www.tunnelbear.com

Invisible Internet Project (I2P)
https://geti2p.net

JonDo
https://anonymous-proxy-servers.net

Proxify
https://proxify.com

Psiphon
https://psiphon.ca

Anonymizers

An anonymizer assists you to mask your IP address to visit websites without being tracked or identified while keeping your activity and identity protected. It uses various techniques such as SSH, VPN, and HTTP proxy that allow you to access blocked or censored content on the Internet with omitted advertisements.

- **Whonix**

  Source: *https://www.whonix.org*

  Whonix is a desktop operating system designed for advanced security and privacy. It mitigates the threat of common attack vectors while maintaining usability. Online anonymity is realized via fail-safe, automatic, and desktop-wide use of the Tor network.

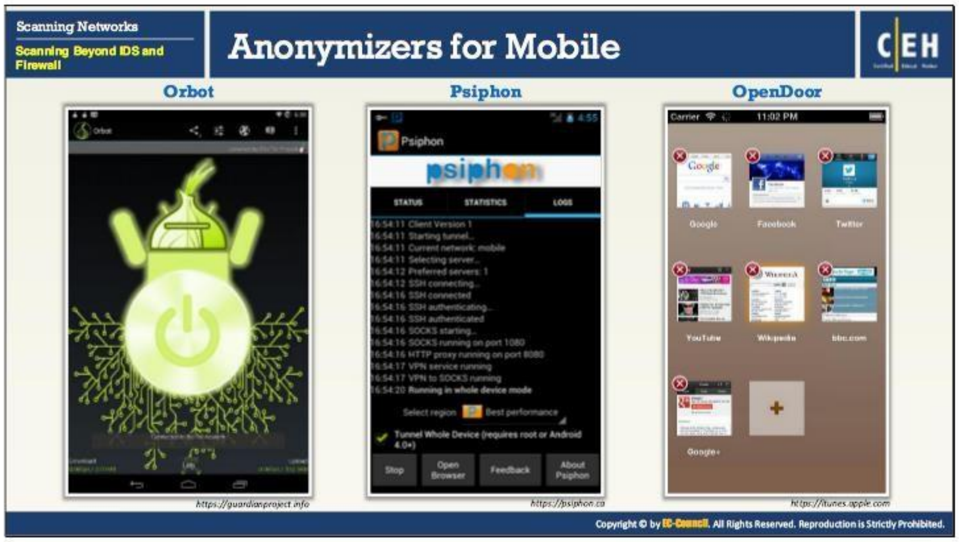
  Features:

  - A heavily reconfigured Debian base is run inside multiple virtual machines, providing a substantial layer of protection from malware and IP address leaks.
  - Commonly used applications are pre-installed and safely pre-configured for immediate use.
  - It is under active development, and it is the only operating system designed to be run inside a VM and paired with Tor.

Some of the anonymizers are as follows:

- TunnelBear (*https://www.tunnelbear.com*)
- Invisible Internet Project (I2P) (*https://geti2p.net*)
- JonDo (*https://anonymous-proxy-servers.net*)

- Proxify (*https://proxify.com*)
- Psiphon (*https://psiphon.ca*)
- Anonymizer Universal (*https://www.anonymizer.com*)
- Anonymous Web Surfing (*http://www.anonymous-surfing.com*)
- Guardster (*http://www.guardster.com*)
- Ultrasurf (*https://ultrasurf.us*)
- Anonym8 (*https://securityonline.info*)
- Web Proxy Server (*http://www.webproxyserver.net*)



Scanning Networks

Scanning Beyond IDS and Firewall

Anonymizers for Mobile

Orbot

Psiphon

OpenDoor

```
16:54:11 Client Version 1
16:54:11 Starting tunnel...
16:54:11 Current network: mobile
16:54:11 Selecting server...
16:54:12 Preferred servers: 1
16:54:12 SSH connecting...
16:54:16 SSH connected
16:54:16 SSH authenticating...
16:54:16 SSH authenticated
16:54:16 SOCKS starting...
16:54:16 SOCKS running on port 1080
16:54:16 HTTP proxy running on port 8080
16:54:17 VPN service running
16:54:17 VPN to SOCKS running
16:54:20 Running in whole device mode
```

Select region: Best performance

✓ Tunnel Whole Device (requires root or Android 4.0+)

https://guardianproject.info

https://psiphon.ca

https://itunes.apple.com

Anonymizers for Mobile

- **Orbot**

  Source: *https://guardianproject.info*

  Orbot is a proxy app that allows other apps to use the Internet more securely. It uses Tor to encrypt Internet traffic, and then hides it by bouncing through a series of computers around the world. Tor is a free software that provides an open network to help defend your system against any form of network surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and a kind of state security monitoring known as "traffic analysis." Orbot creates a truly private Internet connection.

- **Psiphon**

  Source: *https://psiphon.ca*

  Psiphon is a circumvention tool from Psiphon Inc. that utilizes VPN, SSH, and HTTP Proxy technology to provide you with open and uncensored access to Internet content. However, Psiphon does not increase online privacy and is not an online security tool.

  Features:

  - **Browser** or **VPN** (whole-device) **mode**: one can choose whether to tunnel everything or just the web browser.
  - **In-app stats**: This lets you know how much traffic you have been using.

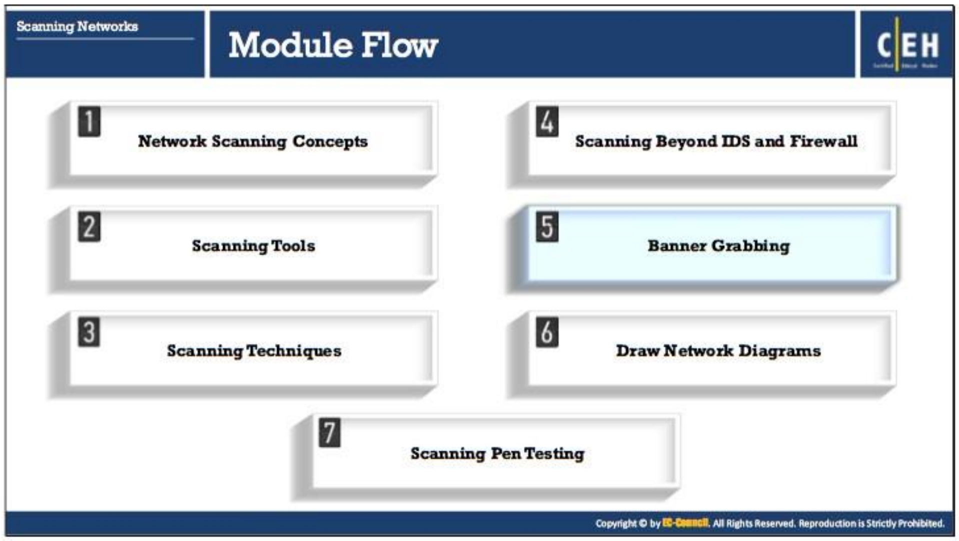
- **OpenDoor**

  Source: *https://itunes.apple.com*

  OpenDoor is an app designed for both iPhone and iPad; it allows you to browse websites smoothly and anonymously.

  Features:

  - **Safe and Smooth Browsing**: OpenDoor caches website contents on its fast and secure servers to speed up access and minimize interruptions.
  - **Total Anonymity**: It protects users' identity on the web via randomized IP address.
  - **No "Content Striping"**: OpenDoor supports JavaScript, video, and multimedia streaming for a rich browsing experience.
  - **Multi-tab Browsing**: It allows you to view multiple websites simultaneously.
  - **Easy to use interface**: It has complete browser functionalities.



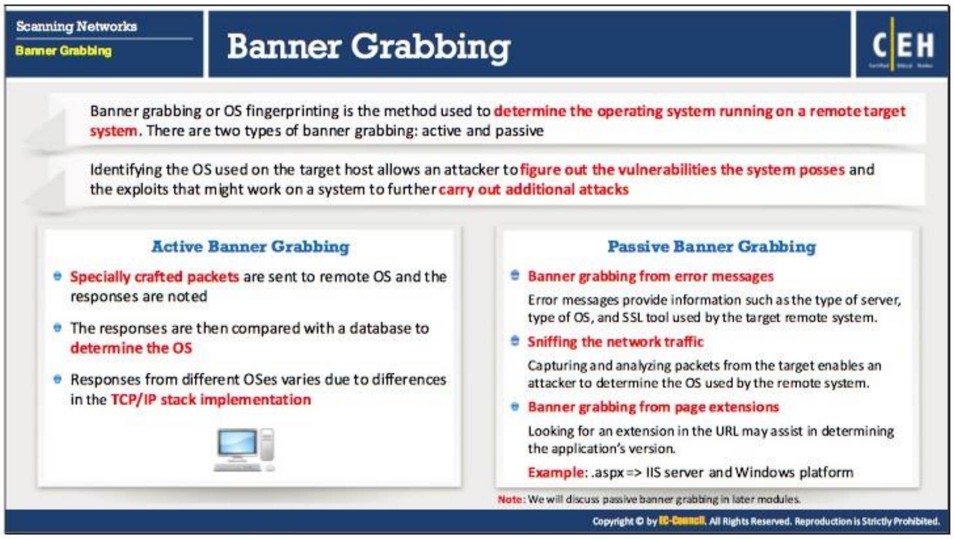
Scanning Networks

Module Flow

1. Network Scanning Concepts
2. Scanning Tools
3. Scanning Techniques
4. Scanning Beyond IDS and Firewall
5. Banner Grabbing
6. Draw Network Diagrams
7. Scanning Pen Testing

Banner Grabbing

An attacker uses banner grabbing techniques to identify network hosts running versions of applications and OSs with known exploits. This section introduces you to banner grabbing, its types, and its tools, as well as useful countermeasures you can employ against it.



Scanning Networks
Banner Grabbing

Banner Grabbing

CEH

Banner grabbing or OS fingerprinting is the method used to **determine the operating system running on a remote target system**. There are two types of banner grabbing: active and passive

Identifying the OS used on the target host allows an attacker to **figure out the vulnerabilities the system posses** and the exploits that might work on a system to further **carry out additional attacks**

Active Banner Grabbing

- **Specially crafted packets** are sent to remote OS and the responses are noted
- The responses are then compared with a database to **determine the OS**
- Responses from different OSes varies due to differences in the **TCP/IP stack implementation**

Passive Banner Grabbing

- **Banner grabbing from error messages**

  Error messages provide information such as the type of server, type of OS, and SSL tool used by the target remote system.
- **Sniffing the network traffic**

  Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
- **Banner grabbing from page extensions**

  Looking for an extension in the URL may assist in determining the application's version.

  Example: .aspx => IIS server and Windows platform

Note: We will discuss passive banner grabbing in later modules.

Banner grabbing, or "OS fingerprinting," is a method used to determine the operating system that is running on a remote target system. It is an important scanning method, as the attacker will have a higher probability of success if the OS of the target system is known (many vulnerabilities are OS-specific). The attacker can then formulate an attack strategy based on the OS of the target system.

There are two methods of banner grabbing: spotting the banner while trying to connect to a service such as FTP site, or downloading the binary file/bin/ls to check the system architecture.

A more advanced fingerprinting technique depends on stack querying, which transfers the packets to the network host and evaluates them by the reply. The first stack-querying method designed with regard to the TCP mode of communication evaluates the response to connection requests.

The next method, known as ISN (Initial Sequence Number) analysis, identifies the differences in random number generators found in the TCP stack.

ICMP response analysis is another method used to fingerprint an OS. It consists of sending ICMP messages to a remote host and evaluating the reply.

The two different types of banner grabbing techniques are presented below:

- **Active Banner Grabbing**

  Active banner grabbing applies the principle that an operating system's IP stack has a unique way of responding to specially crafted TCP packets. This happens because of different interpretations that vendors apply while implementing the TCP/IP stack on the particular OS. In an active banner grabbing, the attacker sends a variety of malformed packets to the remote host, and the responses are compared to a database. Response from different OSes varies due to differences in TCP/IP stack implementation.

For instance, the scanning utility Nmap uses a series of nine tests to determine an OS fingerprint or banner grabbing. The tests listed below provide some ideas about an active banner grabbing attack, as described in *www.packetwatch.net*:

- **Test 1:** A TCP packet with the SYN and ECN-Echo flags enabled is sent to an open TCP port.
- **Test 2:** A TCP packet with no flags enabled is sent to an open TCP port. This type of packet is a NULL packet.
- **Test 3:** A TCP packet with the URG, PSH, SYN, and FIN flags enabled is sent to an open TCP port.
- **Test 4:** A TCP packet with the ACK flag enabled is sent to an open TCP port.
- **Test 5:** A TCP packet with the SYN flag enabled is sent to a closed TCP port.
- **Test 6:** A TCP packet with the ACK flag enabled is sent to a closed TCP port.
- **Test 7:** A TCP packet with the URG, PSH, and FIN flags enabled is sent to a closed TCP port.
- **Test 8 PU (Port Unreachable):** A UDP packet is sent to a closed UDP port. The objective is to extract an "ICMP port unreachable" message from the target machine.
- **Test 9 TSeq (For TCP Sequence ability test):** This test tries to determine the sequence generation patterns of the TCP initial sequence numbers (also known as TCP ISN sampling), the IP identification numbers (also known as IPID sampling), and the TCP timestamp numbers. It sends six TCP packets with the SYN flag enabled to an open TCP port.

The objective of these tests is to find patterns in the initial sequence of numbers that the TCP implementations chose while responding to a connection request. These can be categorized into groups, such as the traditional 64K (many old UNIX boxes), random increments (newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others), or true random (Linux 2.0.*, OpenVMS, newer AIX, etc.). Windows boxes use a "time-dependent" model in which the ISN is incremented by a fixed amount for each occurrence.

- **Passive Banner Grabbing**

  Source: *https://www.symantec.com*

  Like active banner grabbing, passive banner grabbing also depends on the differential implementation of the stack and the various ways an OS responds to packets. However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host via sniffing to study for telltale signs that can reveal an OS.

  Passive banner grabbing includes:

  - **Banner grabbing from error messages**: Error messages provide information, such as type of server, type of OS, and SSL tool used by the target remote system.

- **Sniffing the network traffic**: Capturing and analyzing packets from the target enables an attacker to determine OS used by the remote system.
- **Banner grabbing from page extensions**: Looking for an extension in the URL may assist in determining the application version. Example: .aspx => IIS server and Windows platform.

Given below are the four areas that typically determine the operating system:

- TTL (time to live) of the packets: What does the operating system sets as the Time To Live on the outbound packet?
- Window Size: What is the Window size set by the operating system?
- Whether the DF (Don't Fragment) bit is set: Does the operating system set the Don't Fragment bit?
- TOS (Type of Service): Does the operating system set the Type of Service, and if so, what setting is it?

Passive fingerprinting has to be neither fully accurate nor be limited to these four signatures. However, one can improve accuracy by looking at several signatures, and combining the information. The following is an analysis of a sniffed packet dissected by Lance Spitzner in his paper on passive fingerprinting (*https://www.symantec.com/connect/articles/passive-fingerprinting*):

```
04/20-21:41:48.129662 129.142.224.3:659 -> 172.16.1.107:604
TCP TTL:45 TOS:0x0 ID:56257
***F**A* Seq: 0x9DD90553
Ack: 0xE3C65D7 Win: 0x7D78
```

According to the four criteria, the following are identified:

- TTL: 45
- Window Size: 0x7D78 (or 32120 in decimal)
- DF: The Don't Fragment bit is set
- TOS: 0x0

Compare this information to a database of signatures.

TTL: The TLL from the analysis is 45. The original packet went through 19 hops to get to the target, so it sets the original TTL to 64. Based on this TTL, it appears that the user sent the packet from a Linux or FreeBSD box (however, more system signatures need to be added to the database). This TTL confirms it by implementing a traceroute to the remote host. If the trace needs to be done stealthily, the traceroute TTL (default 30 hops) can be set to one or two hops less than the remote host (-m option). Setting the traceroute in this manner reveals path information (including the upstream provider) without actually touching the remote host.

Window Size: In this step, window sizes are compared. Window size is another effective tool for determining precisely what window size is used and how often it is changed. In the previous signature, the window size is set at 0x7D78, a default window size commonly used by Linux. In addition, FreeBSD and Solaris tend to maintain the same window size throughout a session. However, Cisco routers and Microsoft Windows NT window sizes constantly change. Window size is more accurate when measured after the initial three-way handshake (due to TCP slow start).

DF bit: Most systems use the DF bit set, so this is of limited value. However, this does make it easier to identify few systems that do not use the DF flag (such as SCO or OpenBSD).

TOS: TOS is also of limited value, as it seems to be more session-based than OS-based. In other words, it is not so much the OS that determines the TOS, but the protocol used determines it to a large extent.

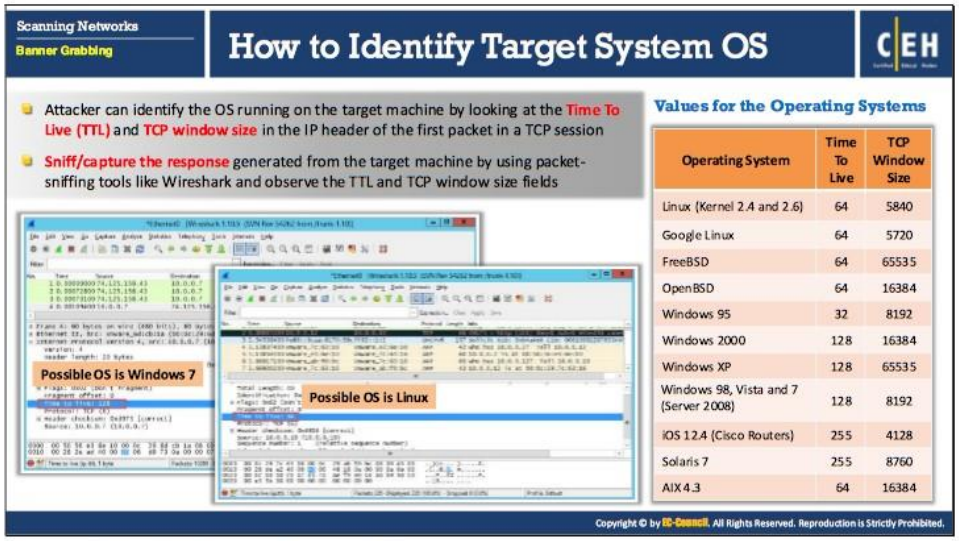
From the information obtained from the packet, specifically the TTL and the window size, one can compare the results to the database of signatures, and with some degree of confidence, determine the OS (in this case, Linux kernel 2.2.x).

Passive fingerprinting, like active fingerprinting, has some limitations. First, applications that build their own packets (e.g., Nmap, Hunt, Nemesis, etc.) will not use the same signatures as the OS. Second, it is relatively simple for a remote host to adjust the TTL, window size, DF, or TOS setting on packets.

Passive fingerprinting has several other uses. For example, Crackers can use stealthy fingerprinting to determine the operating system of a potential target such as a Web server. A user only needs to request a Web page from the server and then analyze the sniffer traces. This bypasses the need for using an active tool that various IDS systems can detect. Passive fingerprinting also helps in identifying remote proxy firewalls. It may be possible to ID proxy firewalls from the signatures as discussed above, simply because proxy firewalls rebuild connections for clients. Similarly, passive fingerprinting can be used to identify rogue systems.

Why Banner Grabbing?

An attacker uses banner grabbing to identify the OS used on the target host and thus determine system vulnerabilities and the exploits that might work on that system to further carry out additional attacks.



Scanning Networks
Banner Grabbing

How to Identify Target System OS

CEH

- Attacker can identify the OS running on the target machine by looking at the **Time To Live (TTL)** and **TCP window size** in the IP header of the first packet in a TCP session
- **Sniff/capture the response** generated from the target machine by using packet-sniffing tools like Wireshark and observe the TTL and TCP window size fields

Possible OS is Windows 7

Possible OS is Linux

Values for the Operating Systems

Operating System	Time To Live	TCP Window Size
Linux (Kernel 2.4 and 2.6)	64	5840
Google Linux	64	5720
FreeBSD	64	65535
OpenBSD	64	16384
Windows 95	32	8192
Windows 2000	128	16384
Windows XP	128	65535
Windows 98, Vista and 7 (Server 2008)	128	8192
iOS 12.4 (Cisco Routers)	255	4128
Solaris 7	255	8760
AIX 4.3	64	16384

How to Identify Target System OS

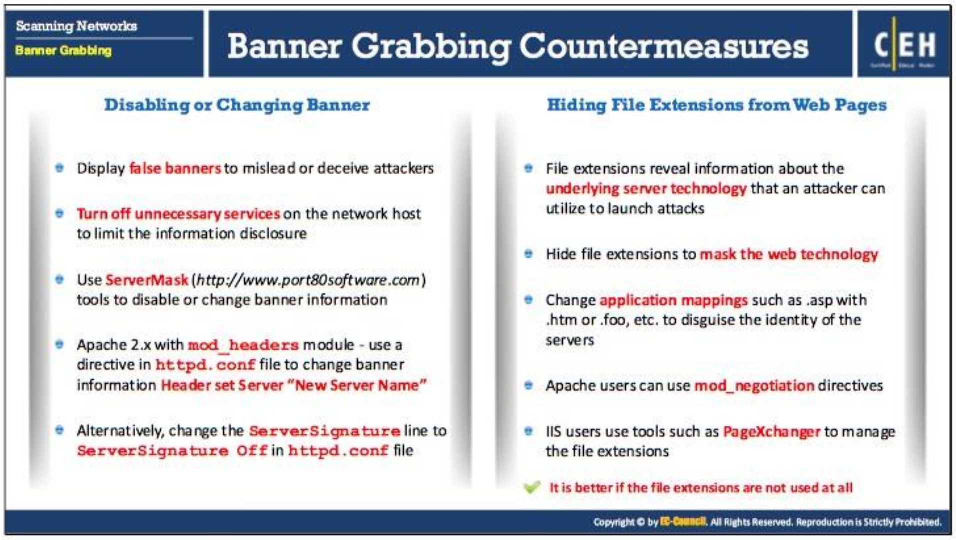
Finding out the target operating system is one of the important tasks for an attacker to compromise the target network/machine. In a network, various standards are implemented in order to allow different operating systems to communicate with each other. These standards govern the functioning of various protocols like IP, TCP, UDP, etc. By analyzing certain parameters/fields in these protocols, one can reveal the details of the operating system. Parameters like Time to Live (TTL) and TCP window size in the IP header of the first packet in a TCP session are helpful in identifying the operating system running on the target machine. The TTL field determines the maximum time a packet can remain in a network, and the TCP window size determines the length of the packet reported. These values differ for different operating systems as presented in the following table:

Operating System	Time To Live	TCP Window Size
Linux (Kernel 2.4 and 2.6)	64	5840
Google Linux	64	5720
FreeBSD	64	65535
OpenBSD	64	16384
Windows 95	32	8192
Windows 2000	128	16384
Windows XP	128	65535
Windows 98, Vista and 7 (Server 2008)	128	8192

iOS 12.4 (Cisco Routers)	255	4128
Solaris 7	255	8760
AIX 4.3	64	16384

TABLE 3.3: Table showing TTL and TCP Window size values for OS

In order to identify the target OS, sniff/capture the response generated from the target machine to the request-originated machine using packet-sniffing tools like Wireshark, etc. and observe the TTL and TCP window size fields in the captured first TCP packet. Comparing these values to that of the above table, you can determine the target operating system that has generated the response.



Scanning Networks
Banner Grabbing

Banner Grabbing Countermeasures

Disabling or Changing Banner

- Display **false banners** to mislead or deceive attackers
- **Turn off unnecessary services** on the network host to limit the information disclosure
- Use **ServerMask** (*http://www.port80software.com*) tools to disable or change banner information
- Apache 2.x with `mod_headers` module - use a directive in `httpd.conf` file to change banner information **Header set Server "New Server Name"**
- Alternatively, change the `ServerSignature` line to `ServerSignature Off` in `httpd.conf` file

Hiding File Extensions from Web Pages

- File extensions reveal information about the **underlying server technology** that an attacker can utilize to launch attacks
- Hide file extensions to **mask the web technology**
- Change **application mappings** such as .asp with .htm or .foo, etc. to disguise the identity of the servers
- Apache users can use **mod_negotiation** directives
- IIS users use tools such as **PageXchanger** to manage the file extensions

✓ **It is better if the file extensions are not used at all**

Banner Grabbing Countermeasures

- **Disabling or Changing Banner**

  Whenever a port is open, it implies that a service/banner is running on it. When attackers connect to the open port using banner grabbing techniques, the system presents a banner containing sensitive information such as OS, server type, and version. With the help of the information gathered, the attacker identifies specific vulnerabilities to exploit and thereafter launches attacks. The countermeasures to defend against banner grabbing attacks are as follows:

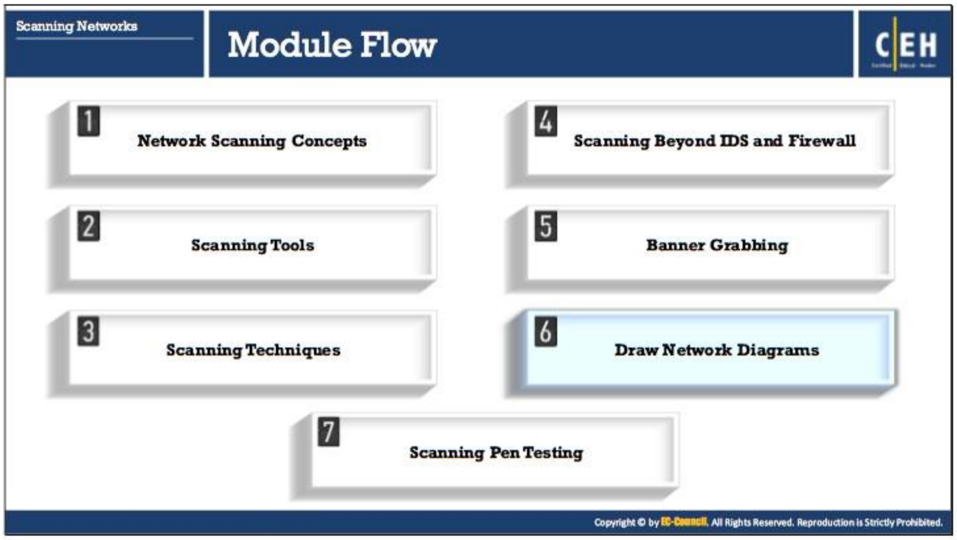
  - Display false banners to mislead or deceive attackers.
  - Turn off unnecessary services on the network host to limit information disclosure.
  - Use ServerMask (*https://www.port80software.com*) tools to disable or change banner information.
  - ServerMask removes unnecessary HTTP header and response data and camouflages the server by providing false signatures. It also provides you with the option of eliminating file extensions such as .asp or .aspx, and it clearly indicates that a site is running on a Microsoft server.
  - Apache 2.x with mod_headers module: use a directive in httpd.conf file to change banner information Header set Server "New Server Name”.
  - Alternatively, change the `ServerSignature` line to `ServerSignatureOff` in the `httpd.conf` file.

- **Hiding File Extensions from Web Pages**

  File extensions reveal information about the underlying server technology that an attacker can utilize to launch attacks. The countermeasures to defend against banner grabbing attacks are as follows:

  - Hide file extensions to mask the web technology.
  - Change application mappings such as .asp with .htm or .foo, etc. to disguise the identity of the servers.
  - Apache users can use mod_negotiation directives.
  - IIS users use tools such as PageXchanger to manage the file extensions.

  Note: It is better if the file extensions are not used at all.



Scanning Networks

Module Flow

1 Network Scanning Concepts

2 Scanning Tools

3 Scanning Techniques

4 Scanning Beyond IDS and Firewall

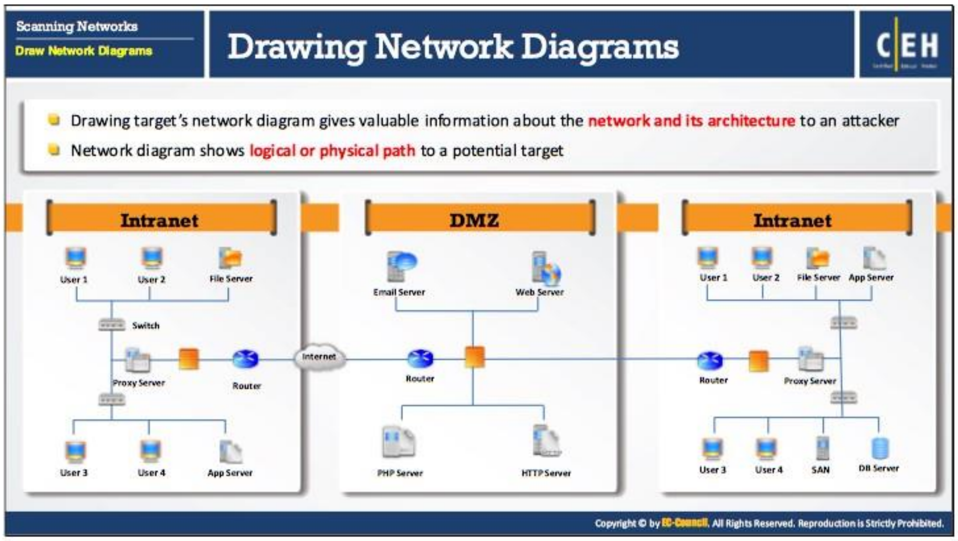
5 Banner Grabbing

6 Draw Network Diagrams

7 Scanning Pen Testing

Draw Network Diagrams

A network diagram helps in analyzing complete network topology. This section highlights the importance of the network diagram, how to draw one, how an attacker uses one to launch an attack, and the tools that help in drawing network maps.



Scanning Networks
Draw Network Diagrams

Drawing Network Diagrams

CEH

- Drawing target's network diagram gives valuable information about the **network and its architecture** to an attacker
- Network diagram shows **logical or physical path** to a potential target

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Drawing Network Diagrams

Drawing a network diagram helps an attacker identify the topology or architecture of a target network. The network diagram also helps to trace out the path to the target host in the network and enables the attacker to understand the position of firewalls, IDSs, routers, and other access control devices. Once the attacker has this information, he/she can try to find the vulnerabilities or weak points of those security mechanisms. Then, the attacker can exploit those security weaknesses to find his/her way into the victim's network.

The network diagram also helps the network administrators to manage their networks. Attackers use network discovery or mapping tools to draw network diagrams of target networks.

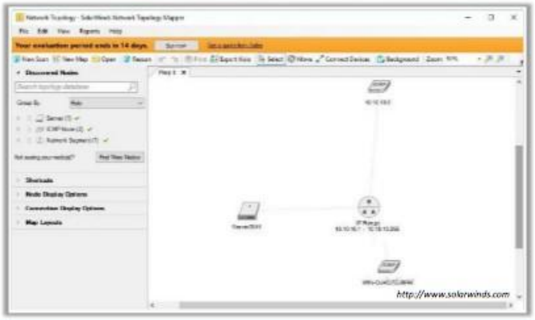
Scanning Networks
Draw Network Diagrams

Network Discovery and Mapping Tools

CEH

Network Topology Mapper

- Network Topology Mapper discovers a network and produces a comprehensive network diagram
- It displays in-depth connections such as OSI Layer 2 and Layer 3 topology data



http://www.solarwinds.com

- **OpManager** *https://www.manageengine.com*
- **The Dude** *https://mikrotik.com*
- **NetSurveyor** *http://nutsaboutnets.com*
- **NetBrain** *https://www.netbraintech.com*
- **Spiceworks Inventory** *https://www.spiceworks.com*

Network Discovery and Mapping Tools

Network discovery and mapping tools allow you to view the map of your network. They help you detect rogue hardware and software violations and notify you whenever a particular host becomes active or goes down. Thus, you can also determine server outages or problems related to performance. An attacker can use the same tools to draw a diagram of the target network, analyze the topology, find the vulnerabilities or weak points, and launch an attack by exploiting them.

- **Network Topology Mapper**

  Source:*http://www.solarwinds.com*

  Network Topology Mapper tool allows one to automatically discover and create a network map of the target network. It is also able to display in-depth connections such as OSI Layer 2 and Layer 3 topology data (e.g. displaying switch-to-switch, switch-to-node, and switch-to-router connections). It can keep track of network changes and allow the user to perform inventory management of hardware and software assets.

  Features:

  - **Network topology discovery and mapping**

    Automatically discovers the entire network and creates comprehensive and detailed network maps

  - **Export network diagrams to Visio**

    Exports network diagrams to Microsoft Office® Visio®, Orion Network Atlas, PDF, and PNG formats

- **Network mapping for regulatory compliance**

  Allows one to directly address PCI compliance and other regulations that require maintenance of an up-to-date network diagram.

- **Multi-level network discovery**

  Performs multi-level network discovery to produce an integrated OSI Layer 2 and Layer 3 network map that includes detailed device information

- **Auto-detection of changes to network topology**

  Automatically detects new devices and changes to network topology with scheduled network scanning

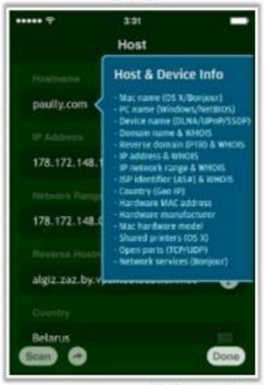
Some of the network discovery and mapping tools an attacker uses to create a network map are discussed below:

- OpManager (*https://www.manageengine.com*)
- The Dude (*https://www.mikrotik.com*)
- NetSurveyor (*http://nutsaboutnets.com*)
- NetBrain (*https://www.netbraintech.com*)
- Spiceworks Inventory (*https://www.spiceworks.com*)
- LANState (*http://www.10-strike.com*)
- Friendly Pinger (*http://www.kilievich.com*)
- WhatsConnected (*https://www.ipswitch.com*)
- Lan-Secure Switch Center (*http://www.lan-secure.com*)
- Intermapper (*https://www.helpsystems.com*)
- SteelCentral NetAuditor (*https://www.riverbed.com*)
- IPsonar (*http://www.lumeta.com*)

Scanning Networks
Draw Network Diagrams

Network Discovery Tools for Mobile

Scany



3:31
Host

Hostname
paully.com

IP Address
178.172.148.1

Network Range
178.172.148.0

Reverse Hostname
algiz.zaz.by.v

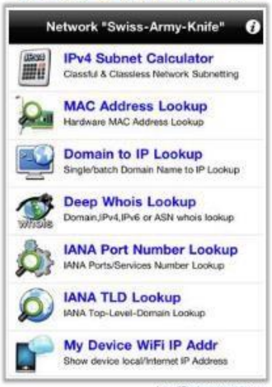
Country
Belarus

Host & Device Info

- Mac name (OS X/Bonjour)
- PC name (Windows/NetBIOS)
- Device name (DLNA/UPnP/SSDP)
- Domain name & WHOIS
- Reverse domain (PTR) & WHOIS
- IP address & WHOIS
- IP network range & WHOIS
- ISP identifier (AS#) & WHOIS
- Country (Geo IP)
- Hardware MAC address
- Hardware manufacturer
- Mac hardware model
- Shared printers (OS X)
- Open ports (TCP/UDP)
- Network services (Bonjour)

http://happymagenta.com

Network "Swiss-Army-Knife"



Network "Swiss-Army-Knife"

IPv4 Subnet Calculator
Classful & Classless Network Subnetting

MAC Address Lookup
Hardware MAC Address Lookup

Domain to IP Lookup
Single/batch Domain Name to IP Lookup

Deep Whois Lookup
Domain,IPv4,IPv6 or ASN whois lookup

IANA Port Number Lookup
IANA Ports/Services Number Lookup

IANA TLD Lookup
IANA Top-Level-Domain Lookup

My Device WiFi IP Addr
Show device local/Internet IP Address

http://foobang.weebly.com

PortDroid Network Analysis
https://play.google.com

NetX - Network Discovery Tools
https://play.google.com

Network Mapper
https://play.google.com

Fing - Network Tools
https://www.fing.io

ezNetScan
https://play.google.com

Network Discovery Tools for Mobile

Given below are network discovery tools for mobile devices:

- **Scany**

  Source: *http://happymagenta.com*

  Scany, a network scanner app for iPhone and iPad, scans LAN, Wi-Fi networks, websites, open ports, discovers network devices, and digs network info. It supports a number of networking protocols and anti-stealth technologies. It is a multifunctional networking instrument for finding connected devices, looking up detailed device information, network troubleshooting, scanning ports, and testing network security and firewalls.

  Features:

  - Scan both LAN and the Internet
  - Scan any IP address or network range
  - Bonjour hostnames lookup
  - Windows hostnames lookup (NetBIOS, Samba)
  - Device names lookup (UPnP, SSDP, DLNA)
  - Detects country of origin and the network owner
  - Network range and AS number lookup
  - MAC address and hardware vendor lookup
  - Wake on LAN or Wi-Fi (by MAC address)

- Wake over the Internet (using proper routers)
- Ping/Trace hosts with integrated tools
- WHOIS hostnames, IP addresses, ASNs
- Know Wi-Fi, VPN, 3G/EDGE, and external IPs
- Checks ICMP, TCP, and UDP
- Faster asynchronous network I/O
- Works with Wi-Fi, 3G, EDGE, and GPRS

- **Network "Swiss-Army-Knife"**

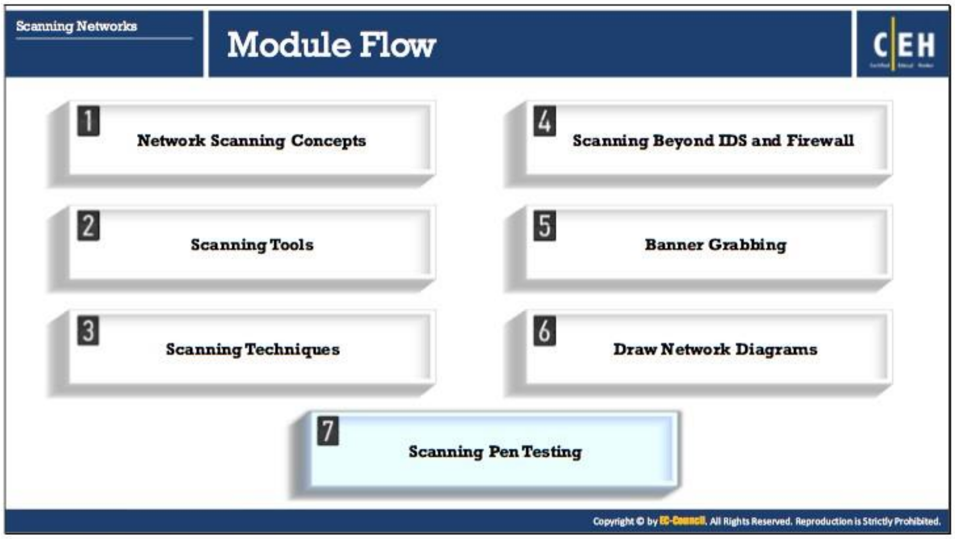
  Source: *http://foobang.weebly.com*

  Network "Swiss-Army-Knife" is a network application for iPhone to perform a number of tasks mentioned below:

  - Calculate IPv4 subnet (Classful and Classless) and all the related valid subnet information
  - Find Offline hardware MAC address to organization lookup and vice versa.
  - Perform Single/Batch Domain-name lookup: For a list of domain names, equivalent IP address can be found
  - Perform WHOIS lookup directly from the idevice. WHOIS lookup permits one to query the list of NICes for detailed information. Lookup relays on IP addresses- either IPv4 or IPv6, domain name or AS Number. Results can be stored in a local repository for future reference.
  - Offline IANA Port number lookup: allows IANA assigned port number to the name and vice versa
  - IANA Top level domain lookup: identifies which countries domain end with .cz, .cv, .su etc.
  - My device Wi-Fi IP addr: allows identification of your local device Wi-Fi IP address

Some of the network discovery tools for mobile include:

- PortDroid Network Analysis (*https://play.google.com*)
- NetX - Network Discovery Tools (*https://play.google.com*)
- Network Mapper (*https://play.google.com*)
- Fing - Network Tools (*https://www.fing.io*)
- ezNetScan (*https://play.google.com*)



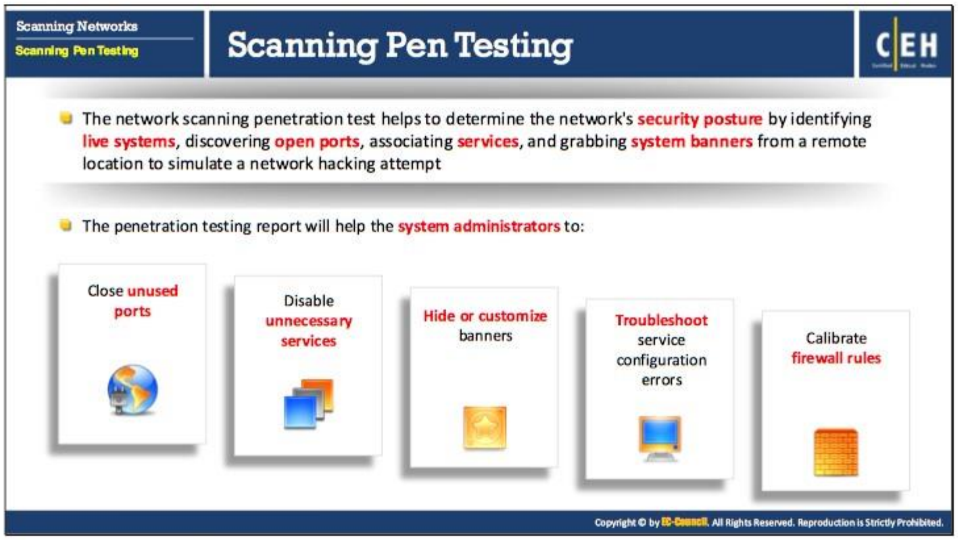
Scanning Networks

Module Flow

1. Network Scanning Concepts
2. Scanning Tools
3. Scanning Techniques
4. Scanning Beyond IDS and Firewall
5. Banner Grabbing
6. Draw Network Diagrams
7. Scanning Pen Testing

Scanning Pen Testing

It is advisable to pen-test the target network to identify its security posture. Pen-testing in anticipation of a possible problem helps to find and fix any security loopholes present in the target network. Such proactive prevention practices can keep an entire network from being compromised. This section describes the steps involved in pen-testing the target network and the various scanning tools used to accomplish this task.

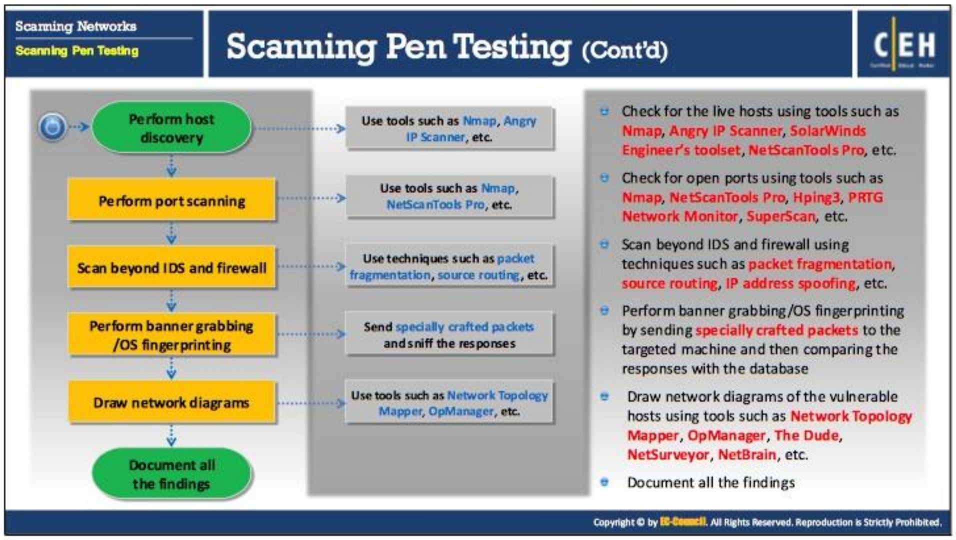


Scanning Networks
Scanning Pen Testing

Scanning Pen Testing

- The network scanning penetration test helps to determine the network's **security posture** by identifying **live systems**, discovering **open ports**, associating **services**, and grabbing **system banners** from a remote location to simulate a network hacking attempt
- The penetration testing report will help the **system administrators** to:

Close **unused ports**
Disable **unnecessary services**
Hide or customize banners
Troubleshoot service configuration errors
Calibrate **firewall rules**



Scanning Networks
Scanning Pen Testing

Scanning Pen Testing (Cont'd)

Perform host discovery
Use tools such as Nmap, Angry IP Scanner, etc.
Perform port scanning
Use tools such as Nmap, NetScanTools Pro, etc.
Scan beyond IDS and firewall
Use techniques such as packet fragmentation, source routing, etc.
Perform banner grabbing /OS fingerprinting
Send specially crafted packets and sniff the responses
Draw network diagrams
Use tools such as Network Topology Mapper, OpManager, etc.
Document all the findings

- Check for the live hosts using tools such as **Nmap**, **Angry IP Scanner**, **SolarWinds Engineer's toolset**, **NetScanTools Pro**, etc.
- Check for open ports using tools such as **Nmap**, **NetScanTools Pro**, **Hping3**, **PRTG Network Monitor**, **SuperScan**, etc.
- Scan beyond IDS and firewall using techniques such as **packet fragmentation**, **source routing**, **IP address spoofing**, etc.
- Perform banner grabbing/OS fingerprinting by sending **specially crafted packets** to the targeted machine and then comparing the responses with the database
- Draw network diagrams of the vulnerable hosts using tools such as **Network Topology Mapper**, **OpManager**, **The Dude**, **NetSurveyor**, **NetBrain**, etc.
- Document all the findings

The network scanning penetration test helps to determine a network's security posture by identifying live systems, discovering open ports and associated services, and grabbing system banners from a remote location to simulate a network hacking attempt. You, as an ethical hacker or pen-tester, should scan and test the network in every manner possible to ensure that there is no security loophole in the system.

Once you are done with the penetration testing, document all your findings at every stage of the testing. This documentation will help the system administrators to:

- Close unused ports if unnecessary/unknown open ports are found
- Disable unnecessary services
- Hide or customize banners
- Troubleshoot service configuration errors
- Calibrate firewall rules to impose more restriction

The more ports that are open on the server, the easier it will be for an attacker to connect to it. The first thing an attacker does is monitor network traffic for vulnerabilities such as open ports and services running, through which the network could be compromised. Admins may install, configure some unwanted services, leave services with default settings, and turn them on during OS and application installations. This can cause unwanted traffic to the server or a way for an attacker to intrude into the system. Attackers might also "banner grab" to trace the server name and its version, and then use this information to break into a network. Therefore, close all the unused/unnecessary open ports, unwanted services, and so on, and configure the server in such a way that it hides the display of the banner. Also create inbound and outbound firewall rules to block all the unwanted ports from allowing any connections from outside the network.

Here is how you can conduct a pen-test of a target network.

- **Step 1: Perform host discovery**

  The first step of network penetration testing is to detect live hosts on the target network. You can attempt to detect the live hosts (i.e., accessible hosts in the target network), using network scanning tools such as Nmap, Angry IP Scanner, SolarWinds Engineer's toolset, and NetScanTools Pro. It is difficult to detect live hosts behind a firewall.

- **Step 2: Perform port scanning**

  Perform port scanning using tools such as Nmap, NetScanTools Pro, Hping3, PRTG Network Monitor, and SuperScan. These tools help to probe a server or host on the target network for open ports. Open ports are the doorways through which an attacker installs malware on a system. Therefore, you should always check for open ports and close them if they are not necessary.

- **Step 3: Scan beyond IDS and firewall**

  Scan beyond IDS and firewall; this helps you to understand the organization's security limitations. Use IDS/firewall evasion techniques such as packet fragmentation, source routing, IP address spoofing, etc. to bypass IDS and firewall rules.

  Use proxy tools such as Proxy Switcher, Proxy Workbench, CyberGhost, Tor, and Burp Suite to hide yourself from detection.

- **Step 4: Perform banner grabbing or OS fingerprinting**

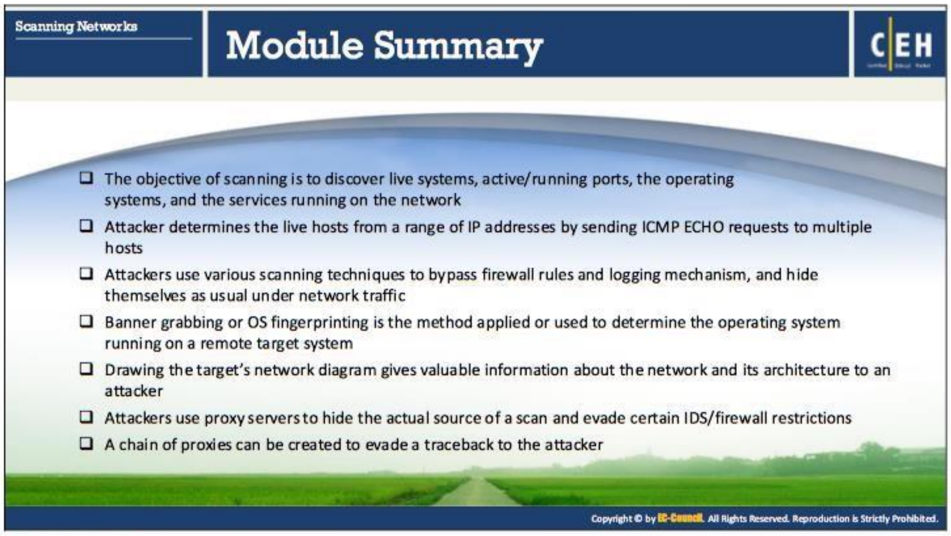
  Perform banner grabbing/OS fingerprinting by sending specially crafted packets to the target machine and then comparing the responses with the database. This determines the operating system running on the target host of a network and its version. Once you know the version and the operating system running on the target system, find and exploit the vulnerabilities related to that OS. Try to gain control over the system and compromise the whole network.

- **Step 5: Draw network diagrams**

  Draw a network diagram of the vulnerable hosts that helps you to understand the logical connection and path to them in the network. You can draw the network diagram with the help of tools such as Network Topology Mapper, OpManager, The Dude, NetSurveyor, and NetBrain. The network diagrams provide valuable information about the network and its architecture.

- **Step 6: Document all the findings**

  The last but the most important step in penetration testing is to preserve all the outcomes of tests conducted in previous steps in a document. This document will assist in finding potential vulnerabilities in the network which you can use to suggest countermeasures. Thus, penetration testing helps in assessing the security posture of the network and fixing any security loopholes before they can cause trouble and result in severe organizational loss.



Scanning Networks

Module Summary

- ☐ The objective of scanning is to discover live systems, active/running ports, the operating systems, and the services running on the network
- ☐ Attacker determines the live hosts from a range of IP addresses by sending ICMP ECHO requests to multiple hosts
- ☐ Attackers use various scanning techniques to bypass firewall rules and logging mechanism, and hide themselves as usual under network traffic
- ☐ Banner grabbing or OS fingerprinting is the method applied or used to determine the operating system running on a remote target system
- ☐ Drawing the target's network diagram gives valuable information about the network and its architecture to an attacker
- ☐ Attackers use proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions
- ☐ A chain of proxies can be created to evade a traceback to the attacker

Module Summary

This module ends with an overview discussion of network scanning concepts. In the next module, we will see how attackers, ethical hackers, and pen-testers perform enumeration to collect information about a target before an attack or audit.