Scanning Networks

Module 03

Ethical Hacking and Countermeasures v8

Module 03: Scanning Networks

Exam 312-50

Security News

Saliently Sality Botnet Trapped Scanning IPv4 Address Space

Source: http://www.spamfighter.com

A semi-famous botnet, Sality, used for locating vulnerable **voice-over-IP** (**VoIP**) servers has been controlled toward determining the entire **IPv4 address space** without setting off alerts, claims a new study, published by Paritynews.com, on October 10, 2012.

Sality is a piece of malware with the primary aim of infecting web servers, **dispersing** spam, and **stealing data**. But the latest research has disclosed other purposes, including recognizing susceptible VoIP targets that could be used in toll fraud attacks.

Through a method called "**reverse-byte order scanning**," Sality can be administered toward scanning possibly the whole IPv4 space, devoid of being recognized. That's the only reason the technique uses a very small number of packets that come from various sources.

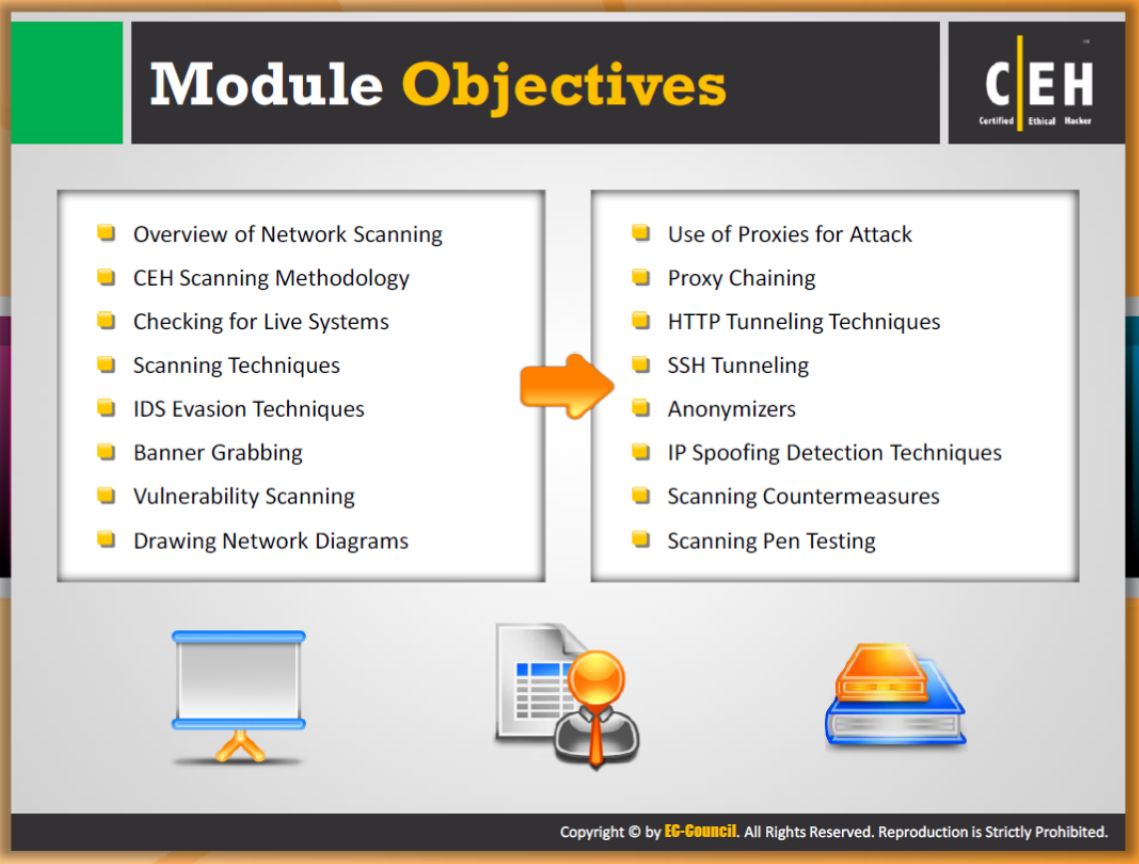
The selection of the target IP addresses develops in **reverse-byte-order increments**. Also, there are many bots contributing in the scan. The conclusion is that a solitary network would obtain scanning packets "**diluted**" over a huge period of time (12 days in this case, from various

sources, **University of California, San Diego (UCSD)**, claimed one of the researchers, Alistair King, as published by Softpedia.com on October 9, 2012).

According to **Alberto Dainotti**, it's not that this stealth-scanning method is exceptional, but it's the first time that such a happening has been both noticed and documented, as reported by Darkreading.com on October 4, 2012. Many other experts hold faith that this manner has been accepted by other botnets. Nevertheless, the team at UCSD is not aware of any data verifying any event like this one.

According to **David Piscitello**, Senior Security Technologist at ICANN, this indeed seems to be the first time that researchers have recognized a botnet that utilizes this scanning method by employing reverse-byte sequential increments of target IP addresses. The **botnet** use classy "**orchestration**" methods to **evade detection**. It can be simply stated that the botnet operator categorized the scans at around **3 million bots** for scanning the full IPv4 address space through a scanning pattern that disperses coverage and partly covers, but is unable to be noticed by present automation, as published by darkreading.com on October 4, 2012.

http://www.spamfighter.com/News-17993-Saliently-Sality-Botnet-Trapped-Scanning-IPv4-Address-Space.htm

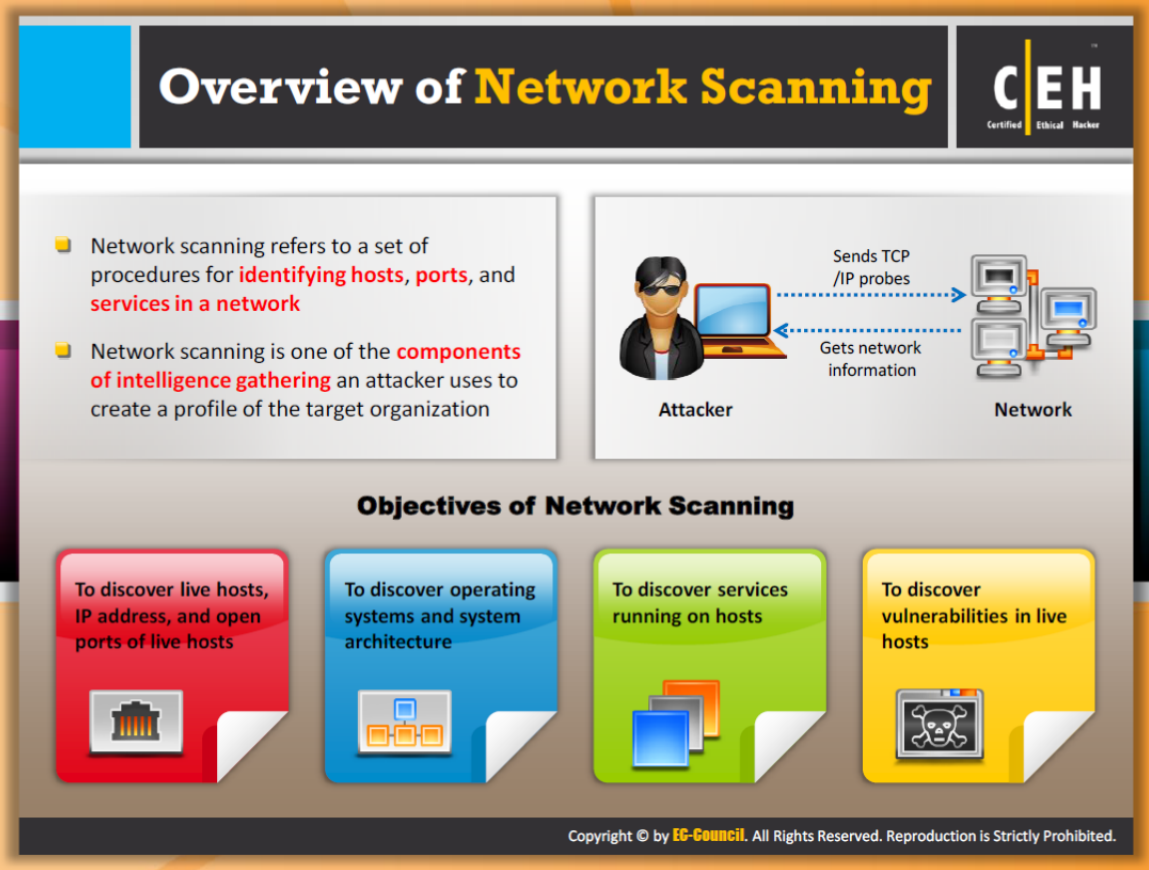


Module Objectives

Once an attacker identifies his/her target system and does the initial reconnaissance, as discussed in the footprinting and reconnaissance module, the attacker concentrates on getting a **mode of entry** into the **target system**. It should be noted that scanning is not limited to intrusion alone. It can be an extended form of reconnaissance where the attacker learns more about his/her target, such as what operating system is used, the services that are being run on the systems, and **configuration lapses** if any can be identified. The **attacker** can then strategize his/her attack, factoring in these aspects.

This module will familiarize you with:

- Overview of Network Scanning
- CEH Scanning Methodology
- Checking for Live Systems
- Scanning Techniques
- IDS Evasion Techniques
- Banner Grabbing
- Vulnerability Scanning
- Drawing Network Diagrams
- Use of Proxies for Attack
- Proxy Chaining
- HTTP Tunneling Techniques
- SSH Tunneling
- Anonymizers
- IP Spoofing Detection Techniques
- Scanning Countermeasures
- Scanning Pen Testing



Overview of Network Scanning

As we already discussed, **footprinting** is the first phase of hacking in which the attacker gains information about a potential target. Footprinting alone is not enough for hacking because here you will gather only the primary information about the target. You can use this primary information in the next phase to gather many more details about the target. The process of **gathering additional details** about the target using highly complex and aggressive reconnaissance techniques is called **scanning**.

The idea is to discover **exploitable communication channels**, to probe as many listeners as possible, and to keep track of the ones that are responsive or useful for hacking. In the scanning phase, you can find various ways of intruding into the target system. You can also discover more about the **target system**, such as what **operating system** is used, what **services** are **running**, and whether or not there are any **configuration lapses** in the target system. Based on the facts that you gather, you can form a strategy to launch an attack.

Types of Scanning

- **Port scanning** - Open ports and services
- **Network scanning** - IP addresses
- **Vulnerability scanning** - Presence of known weaknesses

In a traditional sense, the **access points** that a thief looks for are the doors and windows. These are usually the house's points of vulnerability because of their relatively easy accessibility. When it comes to computer systems and networks, **ports** are the doors and windows of the system that an intruder uses to gain access. The more the ports are open, the more points of vulnerability, and the fewer the ports open, the more secure the system is. This is simply a general rule. In some cases, the level of vulnerability may be high even though few ports are open.

Network scanning is one of the most important phases of intelligence gathering. During the network scanning process, you can gather information about specific IP addresses that can be accessed over the Internet, their targets' operating systems, system architecture, and the services running on each computer. In addition, the attacker also gathers details about the networks and their individual host systems.

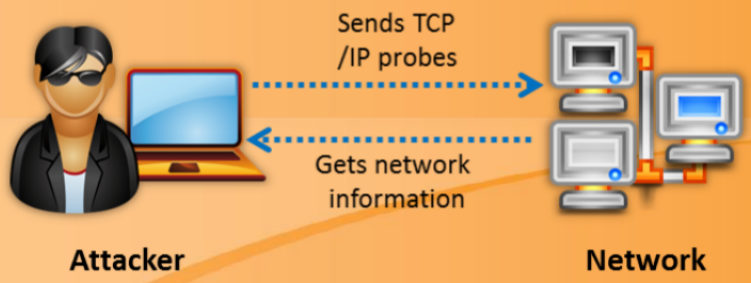


FIGURE 3.1: Network Scanning Diagram

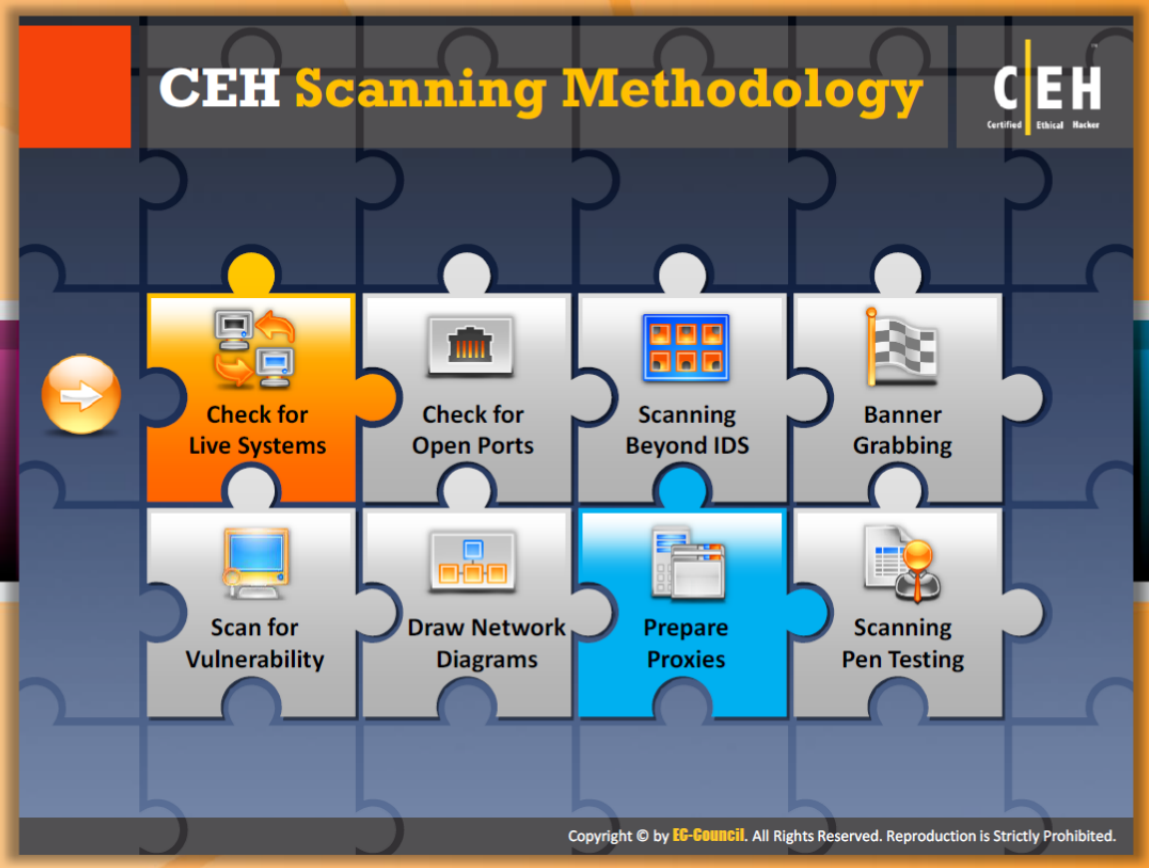
Objectives of Network Scanning

If you have a large amount of information about a **target organization**, there are greater chances for you to learn the **weakness** and **loopholes** of that particular organization, and consequently, for gaining unauthorized access to their network.

Before launching the attack, the attacker observes and analyzes the target network from different perspectives by performing different types of **reconnaissance**. How to perform scanning and what type of information to be achieved during the scanning process entirely depends on the hacker's viewpoint. There may be many objectives for performing scanning, but here we will discuss the most common objectives that are encountered during the hacking phase:

- **Discovering live hosts, IP address, and open ports of live hosts running on the network.**
- **Discovering open ports:** Open ports are the best means to break into a system or network. You can find easy ways to break into the target organization's network by discovering open ports on its network.
- **Discovering operating systems and system architecture of the targeted system:** This is also referred to as fingerprinting. Here the attacker will try to launch the attack based on the operating system's vulnerabilities.

- **Identifying the vulnerabilities and threats:** Vulnerabilities and threats are the security risks present in any system. You can compromise the system or network by exploiting these vulnerabilities and threats.
- **Detecting the associated network service of each port**

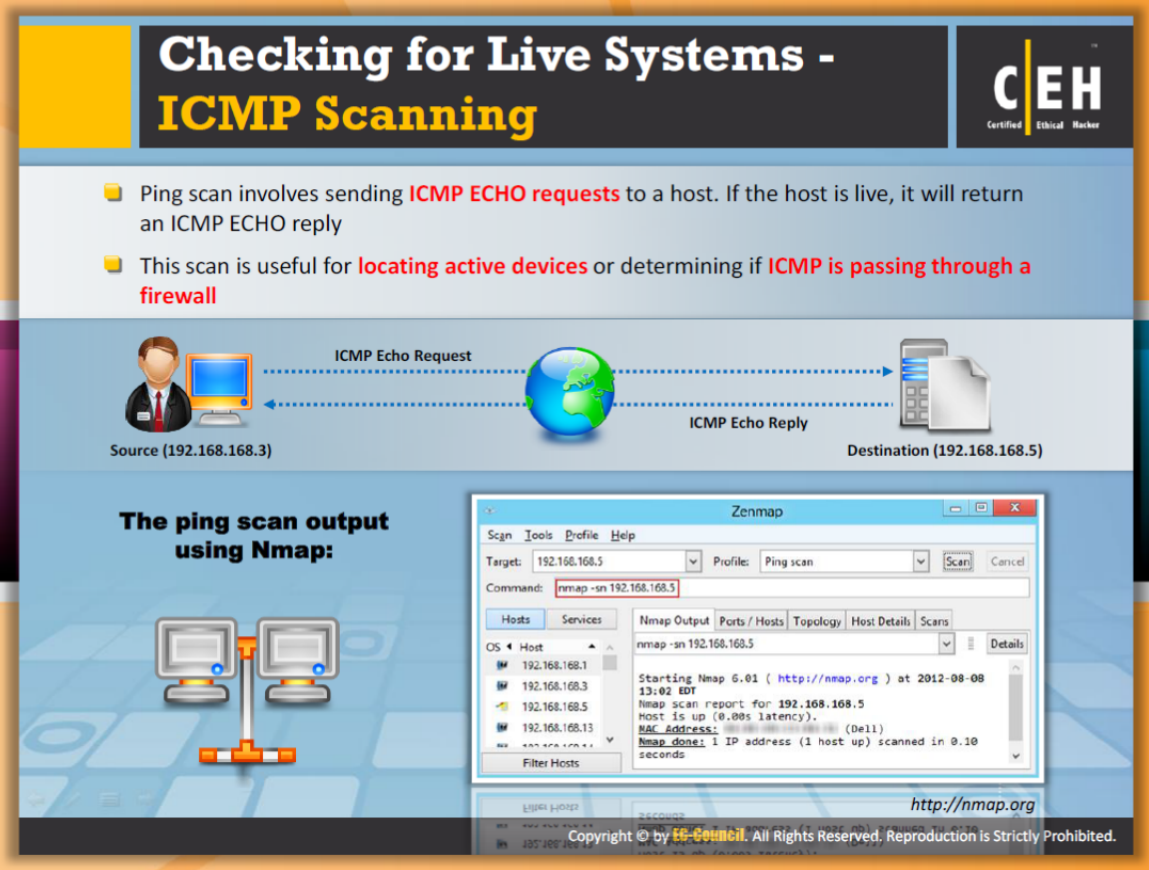


CEH Scanning Methodology

The first step in scanning the network is to check for live systems.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

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.



Checking for Live Systems - ICMP Scanning

ICMP Scanning

All required information about a system can be gathered by **sending ICMP packets** to it. Since ICMP does not have a port abstraction, this cannot be considered a case of port scanning. However, it is useful to determine which hosts in a network are up by pinging them all (the -P option does this; ICMP scanning is now in parallel, so it can be quick). The user can also increase the number of pings in parallel with the -L option. It can also be helpful to tweak the ping timeout value with the -T option.

ICMP Query

The UNIX tool **ICMPquery** or **ICMPush** can be used to request the time on the system (to find out which time zone the system is in) by sending an ICMP type 13 message (TIMESTAMP). The netmask on a particular system can also be determined with ICMP type 17 messages (ADDRESS MARK REQUEST). After finding the netmask of a network card, one can determine all the subnets in use. After gaining information about the subnets, one can target only one particular subnet and avoid hitting the broadcast addresses.

ICMPquery 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.

FIGURE 3.2: ICMP Query Diagram

Ping Scan Output Using Nmap

Source: http://nmap.org

Nmap is a tool that can be used for ping scans, also known as host discovery. Using this tool you can determine the live hosts on a network. It performs ping scans by sending the ICMP ECHO requests to all the hosts on the network. If the host is live, then the host sends an ICMP ECHO reply. This scan is useful for locating active devices or determining if ICMP is passing through a firewall.

The following screenshot shows the sample output of a ping scan using **Zenmap**, the official cross-platform GUI for the Nmap Security Scanner:

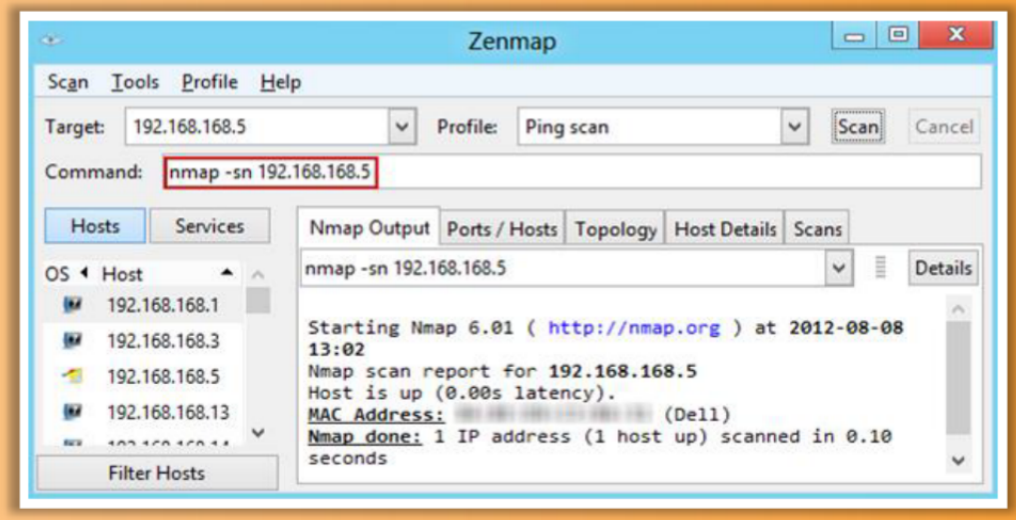
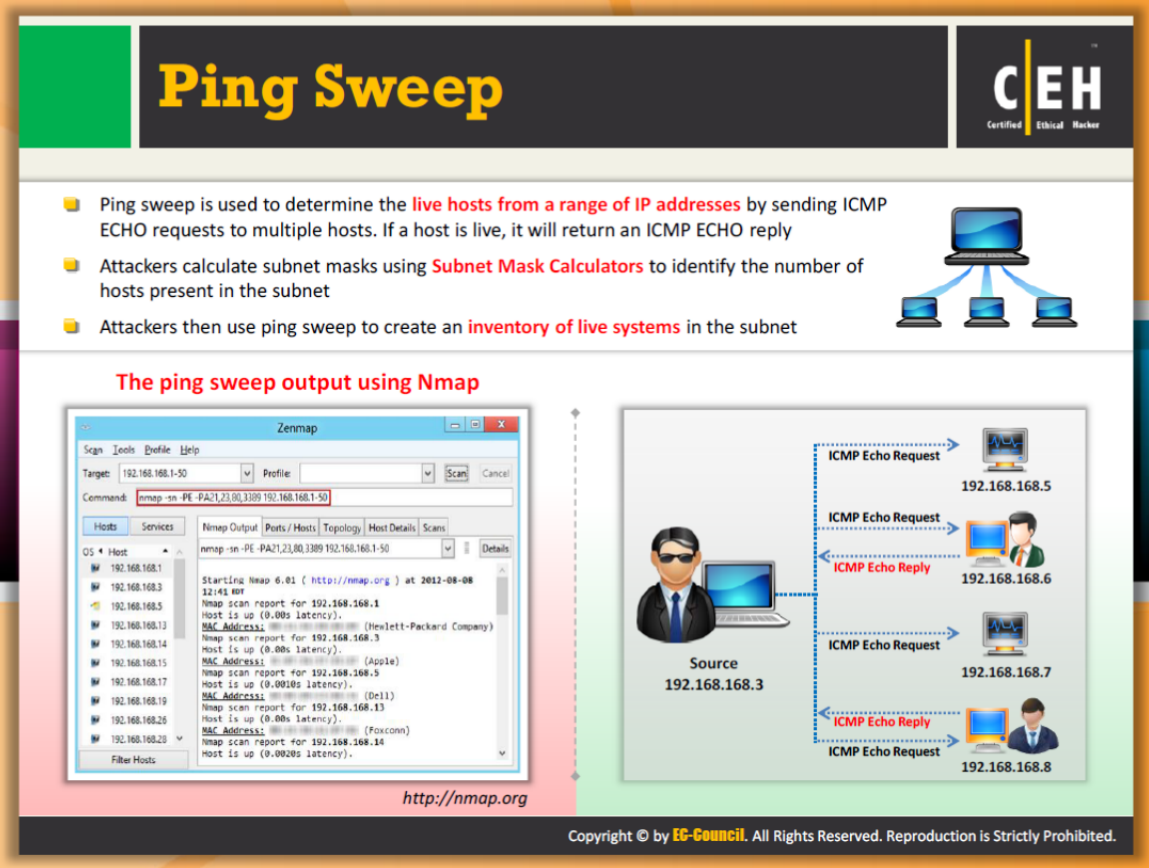


FIGURE 3.3: Zenmap Showing Ping Scan Output



Ping Sweep

A ping sweep (also known as an **ICMP sweep**) is a basic network scanning technique to determine which range of IP addresses map to **live hosts** (computers). While a single ping tells the user whether one specified host computer exists on the network, a ping sweep consists of ICMP ECHO requests sent to multiple hosts.

ICMP ECHO Reply

If a host is active, it returns an **ICMP ECHO** reply. Ping sweeps are among the oldest and slowest methods to scan a network. This utility is distributed across almost all platforms, and acts like a roll call for systems; a system that is live on the network answers the ping query that is sent by another system.

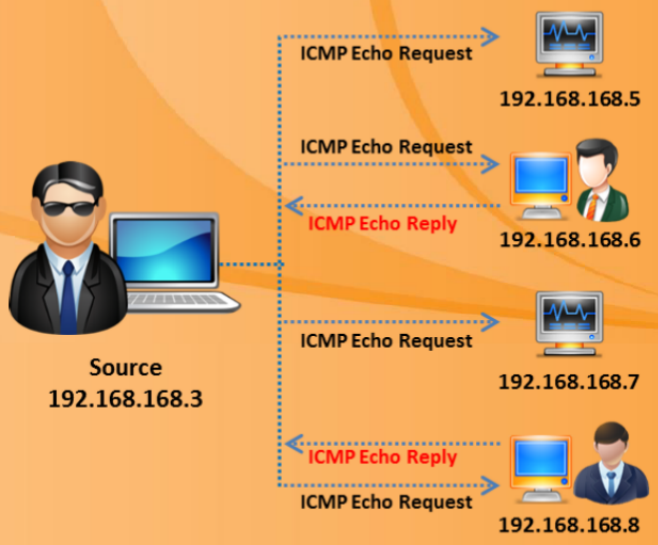


FIGURE 3.4: Ping Sweep Diagram

TCP/IP Packet

To understand ping, you should be able to understand the **TCP/IP packet**. When a system pings, a single packet is sent across the network to a specific IP address. This packet contains **64 bytes**, i.e., **56 data bytes** and **8 bytes** of protocol header information. The sender then waits for a return packet from the target system. A good return packet is expected only when the connections are good and when the targeted system is active. Ping also determines the number of hops that lie between the two computers and the **round-trip time**, i.e., the total time taken by a packet for completing a trip. Ping can also be used for resolving host names. In this case, if the packet bounces back when sent to the IP address, but not when sent to the name, then it is an indication that the system is unable to resolve the name to the specific IP address.

Source: http://nmap.org

Using **Nmap Security Scanner** you can perform ping sweep. Ping sweep determines the IP addresses of live hosts. This provides information about the live host IP addresses as well as their MAC address. It allows you to scan multiple hosts at a time and determine active hosts on the network. The following screenshot shows the result of a ping sweep using Zenmap, the official cross-platform GUI for the Nmap Security Scanner:

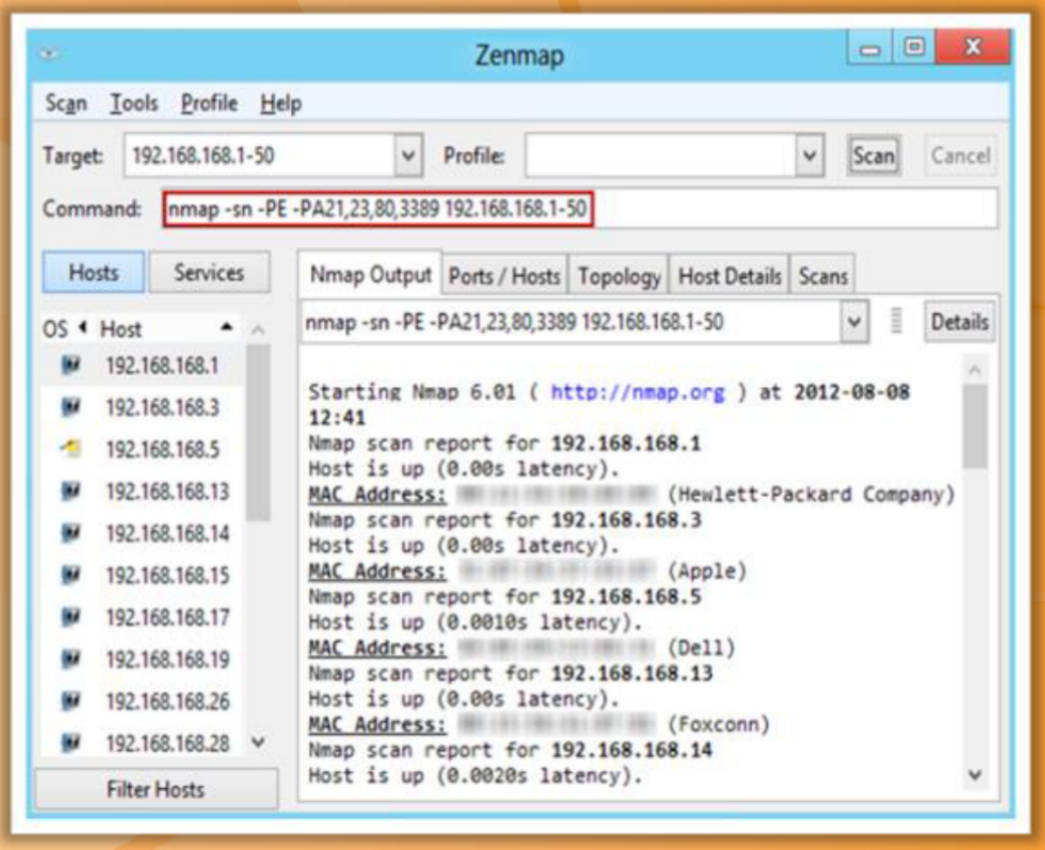
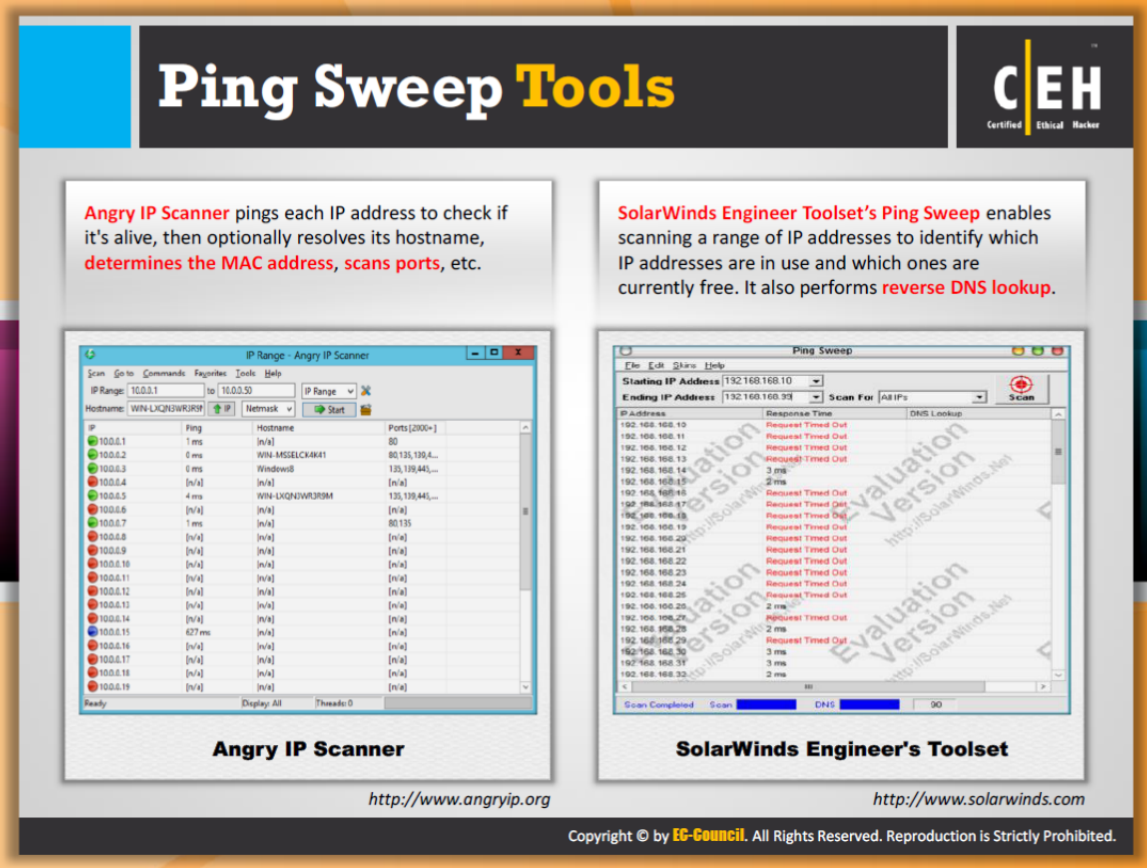


FIGURE 3.5: Zenmap showing ping sweep output



Ping Sweep Tools

Determining live hosts on a target network is the first step in the process of **hacking or breaking** into a network. This can be done using ping sweep tools. There are a number of ping sweep tools readily available in the market using which you can perform ping sweeps easily. These tools allow you to determine the live hosts by sending ICMP ECHO requests to multiple hosts at a time. **Angry IP Scanner** and **Solarwinds Engineer's Toolset** are a few commonly used ping sweep tools.

Angry IP Scanner

Source: http://www.angryip.org

Angry IP Scanner is an **IP scanner tool**. This tool identifies all non-responsive addresses as dead nodes, and resolves hostname details, and checks for open ports. The main feature of this tool is multiple ports scanning, configuring scanning columns. Its main goal is to find the active hosts in the network by scanning all the IP addresses as well as ports. It runs on Linux, Windows, Mac OS X, etc. It can scan IP addresses ranging from **1.1.1.1** to **255.255.255.255**.

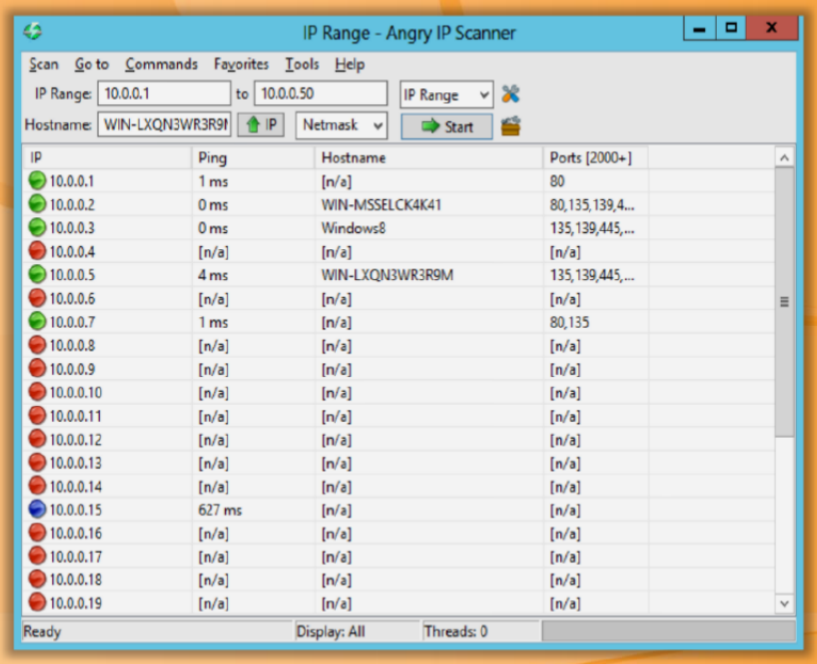


FIGURE 3.6: Angry IP Scanner Screenshot

Solarwinds Engineer's Toolset

Source: http://www.solarwinds.com

The Solarwinds Engineer's Toolset is a collection of **network engineer's tools**. By using this toolset you can scan a range of IP addresses and can identify the IP addresses that are in use currently and the IP addresses that are free. It also performs **reverse DNS lookup**.

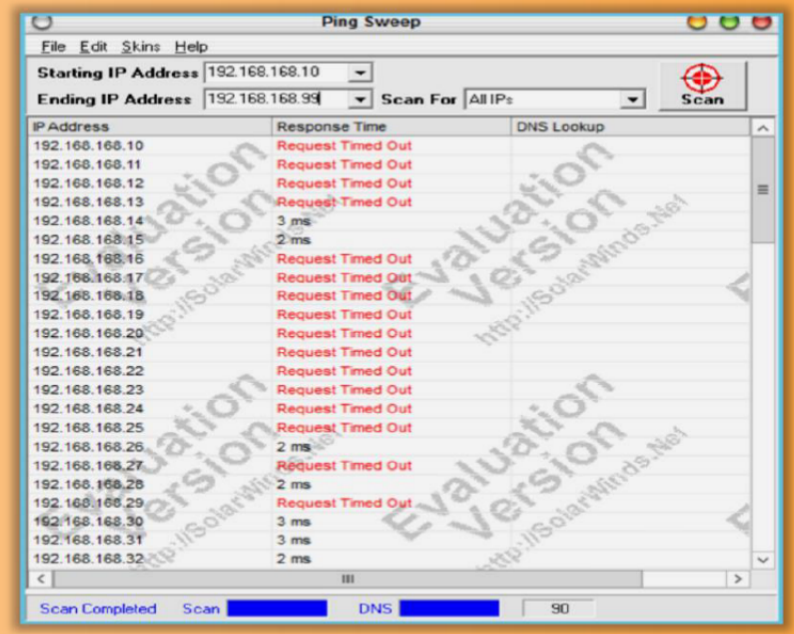
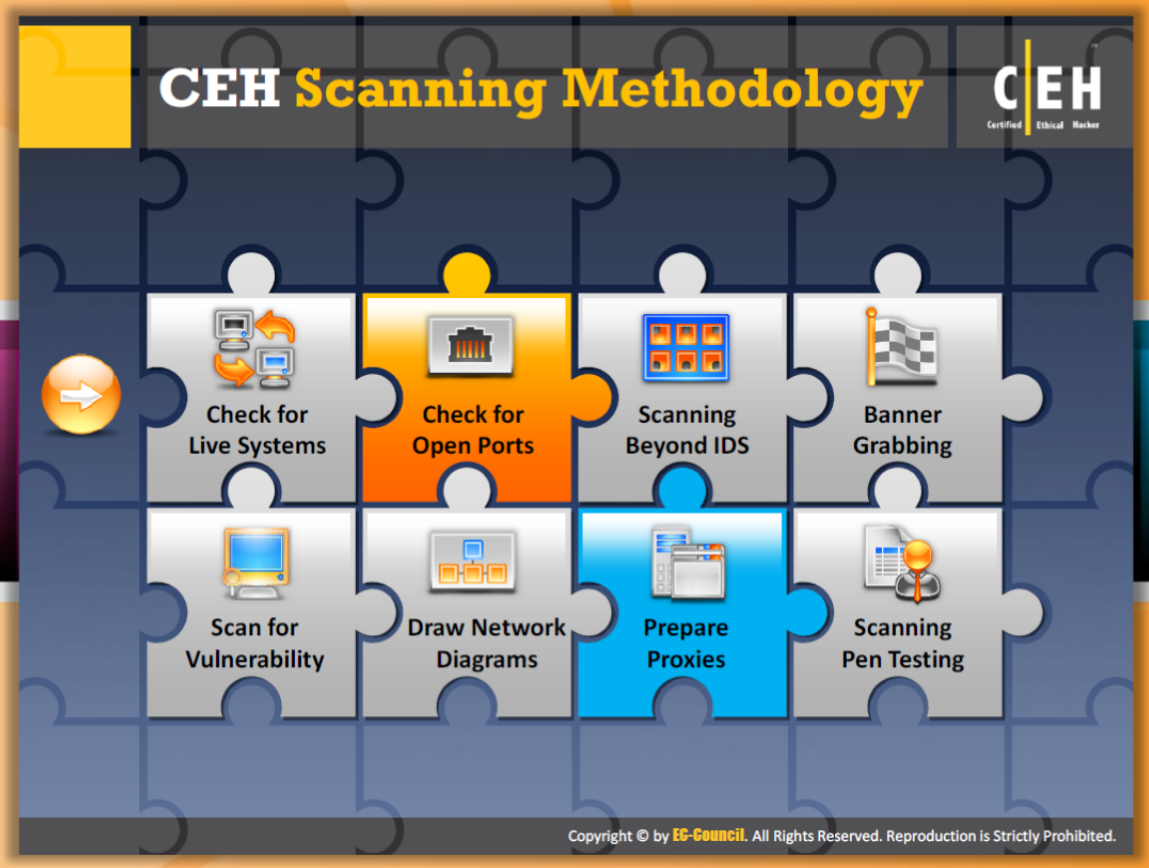


FIGURE 3.7: Solarwinds Engineer's Toolset Screenshot

Ping Sweep Tools (Cont'd)

In addition to Solarwinds Engineer's Toolset and Angry IP Scanner, there are many other tools that feature ping sweep capabilities. For example:

- Colasoft Ping Tool available at http://www.colasoft.com
- Visual Ping Tester – Standarad available at http://www.pingtester.net
- Ping Scanner Pro available at http://www.digilextechnologies.com
- Ultra Ping Pro available at http://ultraping.webs.com
- PingInfoView available at http://www.nirsoft.net
- PacketTrap MSP available at http://www.packettrap.com
- Ping Sweep available at http://www.whatsupgold.com
- Network Ping available at http://www.greenline-soft.com
- Ping Monitor available at http://www.niliand.com
- Pinkie available at http://www.ipuptime.net

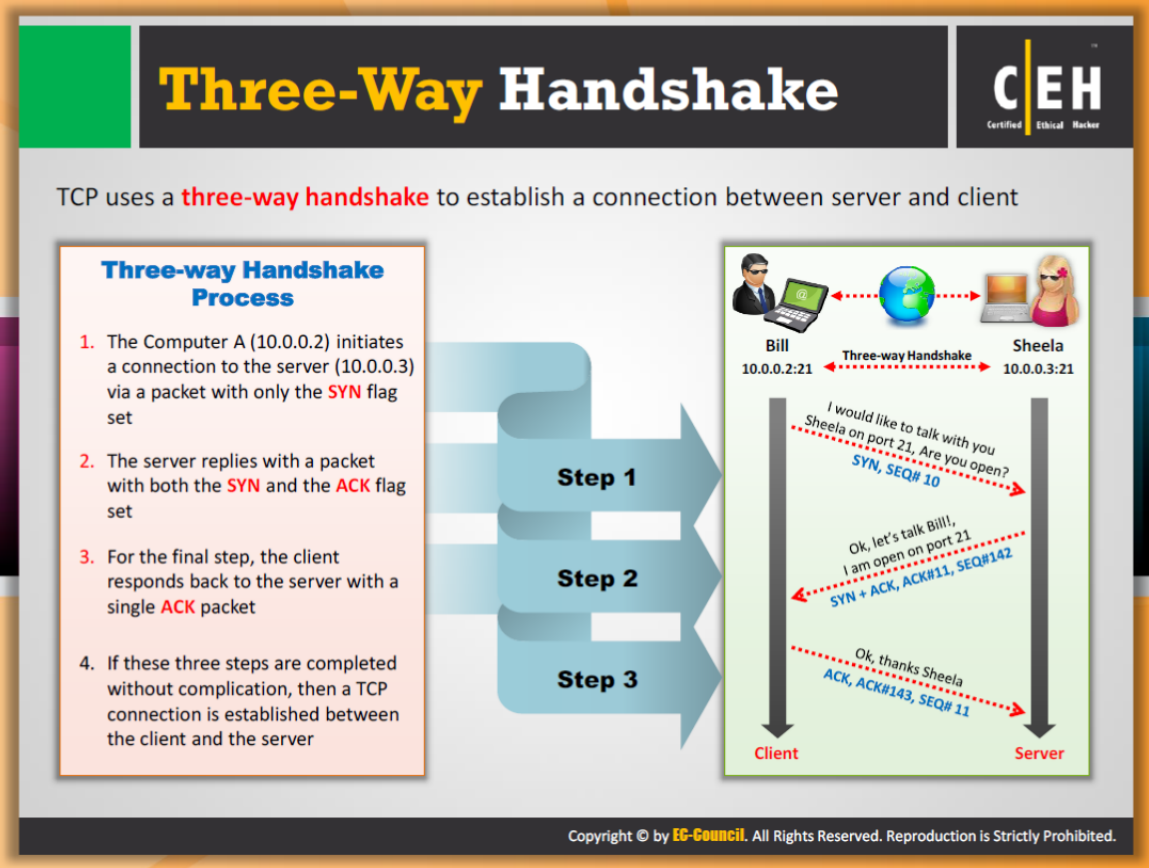


CEH Scanning Methodology

So far we discussed how to check for live systems. Open ports are the doorways for an attacker to launch attacks on systems. Now we will discuss scanning for open ports.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section covers the three-way handshake, scanning IPv6 networks, and various scanning techniques such as FIN scan, SYN scan, and so on.



Three-Way Handshake

TCP is connection-oriented, which implies connection establishment is principal prior to data transfer between applications. This connection is possible through the process of the three-way handshake. The three-way handshake is implemented for establishing the **connection between protocols**.

The three-way handshake process goes as follows:

- To launch a **TCP connection**, the source (10.0.0.2:62000) sends a SYN packet to the destination (10.0.0.3:21).
- The destination, on receiving the SYN packet, i.e., sent by the source, responds by sending a SYN/ACK packet back to the source.
- This ACK packet confirms the arrival of the first SYN packet to the source.
- In conclusion, the source sends an ACK packet for the ACK/SYN packet sent by the destination.
- This triggers an "**OPEN**" connection allowing communication between the source and the destination, until either of them issues a "FIN" packet or a "**RST**" packet to close the connection.

The TCP protocol maintains stateful connections for all connection-oriented protocols across the Internet, and works the same as 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."

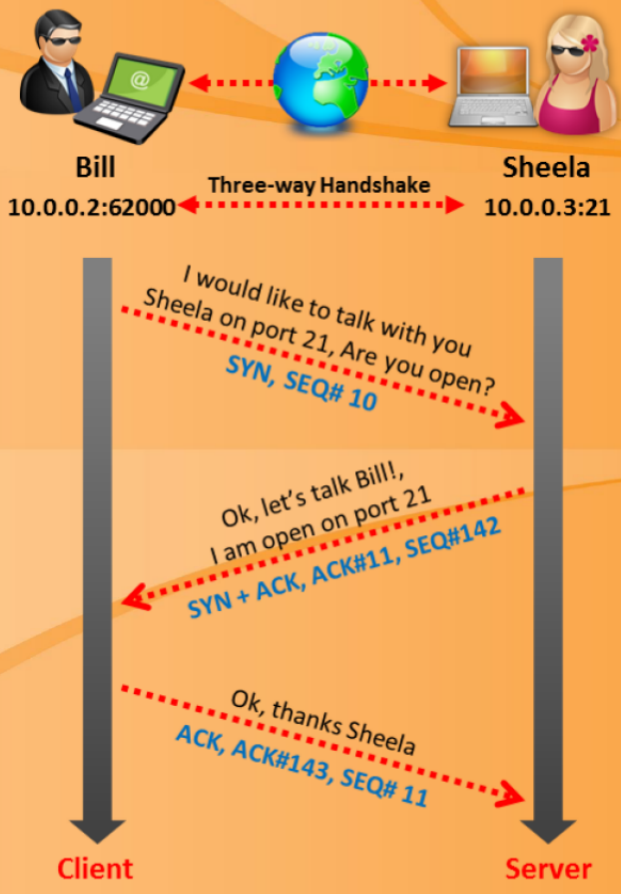


FIGURE 3.8: Three-way Handshake Process

Establishing a TCP Connection

As we previously discussed, a TCP connection is established based on the three-way hand shake method. It is clear from the name of the connection method that the establishment of the connection is accomplished in **three main steps**.

Source: http://support.microsoft.com/kb/172983

The following three frames will explain the establishment of a TCP connection between nodes NTW3 and BDC3:

Frame 1:

In the first step, the client, NTW3, sends a SYN segment (TCP ....S.). This is a request to the server to synchronize the sequence numbers. It specifies its Initial Sequence Number (ISN), which is incremented by 1 and that is sent to the server. To initialize a connection, the client and server must synchronize each other's sequence numbers. There is also an option for the

Maximum Segment Size (MSS) to be set, which is defined by the length (len: 4), this option communicates the maximum segment size the sender wants to receive. The Acknowledgement field (ack: 0) is set to zero because this is the first part of the three-way handshake.

```
1    2.0785 NTW3 --> BDC3 TCP ....S., len: 4, seq: 8221822-8221825, ack: 0,
```

win: 8192, src: 1037 dst: 139 (NBT Session) NTW3 --> BDC3 IP

```
TCP: ....S., len: 4, seq: 8221822-8221825, ack: 0, win: 8192, src: 1037
dst:  139 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 8221822 (0x7D747E)
   TCP: Acknowledgement Number = 0 (0x0)
   TCP: Data Offset = 24 (0x18)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x02 : ....S.

      TCP: ..0..... = No urgent data
      TCP: ...0.... = Acknowledgement field not significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......1. = Synchronize sequence numbers
      TCP: .......0 = No Fin

   TCP: Window = 8192 (0x2000)
   TCP: Checksum = 0xF213
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Options

         TCP: Option Kind (Maximum Segment Size) = 2 (0x2)
         TCP: Option Length = 4 (0x4)
         TCP: Option Value = 1460 (0x5B4)
   TCP: Frame Padding

00000:  02 60 8C 9E 18 8B 02 60 8C 3B 85 C1 08 00 45 00   .`.....`.;....E.
00010:  00 2C 0D 01 40 00 80 06 E1 4B 83 6B 02 D6 83 6B   .,..@....K.k...k
00020:  02 D3 04 0D 00 8B 00 7D 74 7E 00 00 00 00 60 02   .......}t~....`.
00030:  20 00 F2 13 00 00 02 04 05 B4 20 20                .........
```

Frame 2:

In the second step, the server, BDC3, sends an ACK and a SYN on this segment (TCP .A..S.). In this segment the server is acknowledging the request of the client for synchronization. At the same time, the server is also sending its request to the client for synchronization of its sequence numbers. There is one major difference in this segment. The server transmits an acknowledgement number (8221823) to the client. The acknowledgement is just proof to the client that the ACK is specific to the SYN the client initiated. The process of acknowledging the client's request allows the server to increment the client's sequence number by one and uses it as its acknowledgement number.

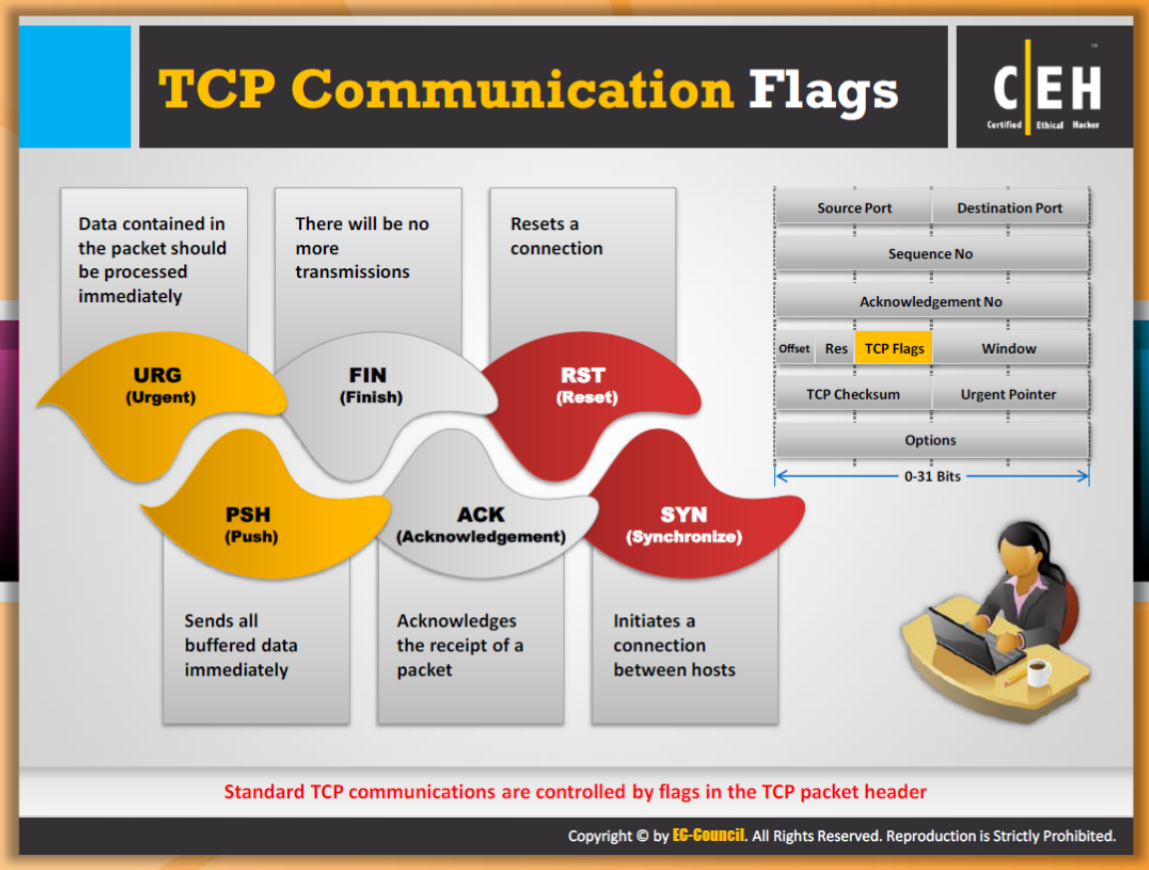
```
2   2.0786 BDC3 --> NTW3  TCP .A..S., len: 4, seq: 1109645-1109648, ack:
8221823, win: 8760, src: 139 (NBT Session)  dst: 1037 BDC3 --> NTW3  IP
TCP: .A..S., len:    4, seq:   1109645-1109648, ack:   8221823, win: 8760,
src:  139 (NBT Session)  dst: 1037
   TCP: Source Port = NETBIOS Session Service
   TCP: Destination Port = 0x040D
   TCP: Sequence Number = 1109645 (0x10EE8D)
   TCP: Acknowledgement Number = 8221823 (0x7D747F)
   TCP: Data Offset = 24 (0x18)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x12 : .A..S.
      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......1. = Synchronize sequence numbers
      TCP: .......0 = No Fin
   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x012D
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Options
         TCP: Option Kind (Maximum Segment Size) = 2 (0x2)
         TCP: Option Length = 4 (0x4)
         TCP: Option Value = 1460 (0x5B4)
   TCP: Frame Padding

00000:  02 60 8C 3B 85 C1 02 60 8C 9E 18 8B 08 00 45 00   .`.;...`......E.
00010:  00 2C 5B 00 40 00 80 06 93 4C 83 6B 02 D3 83 6B   .,[.@....L.k...k
00020:  02 D6 00 8B 04 0D 00 10 EE 8D 00 7D 74 7F 60 12   ...........}t`.
```

```
00030:  22 38 01 2D 00 00 02 04 05 B4 20 20              "8.-......
```

Frame 3:

In the third step, the client sends an ACK on this segment (TCP .A....). In this segment, the client is acknowledging the request from the server for synchronization. The client uses the same algorithm the server implemented in providing an acknowledgement number. The client's acknowledgment of the server's request for synchronization completes the process of establishing a reliable connection, thus the three-way handshake.

```
3   2.787 NTW3 --> BDC3  TCP .A...., len: 0, seq: 8221823-8221823, ack:
1109646, win: 8760, src: 1037  dst:  139 (NBT Session)  NTW3 --> BDC3  IP

TCP: .A...., len:    0, seq:   8221823-8221823, ack:   1109646, win: 8760,
src: 1037  dst:  139 (NBT Session)

   TCP: Source Port = 0x040D
   TCP: Destination Port = NETBIOS Session Service
   TCP: Sequence Number = 8221823 (0x7D747F)
   TCP: Acknowledgement Number = 1109646 (0x10EE8E)
   TCP: Data Offset = 20 (0x14)
   TCP: Reserved = 0 (0x0000)
   TCP: Flags = 0x10 : .A....

      TCP: ..0..... = No urgent data
      TCP: ...1.... = Acknowledgement field significant
      TCP: ....0... = No Push function
      TCP: .....0.. = No Reset
      TCP: ......0. = No Synchronize
      TCP: .......0 = No Fin

   TCP: Window = 8760 (0x2238)
   TCP: Checksum = 0x18EA
   TCP: Urgent Pointer = 0 (0x0)
   TCP: Frame Padding

00000:  02 60 8C 9E 18 8B 02 60 8C 3B 85 C1 08 00 45 00   .`.....`.;....E.
00010:  00 28 0E 01 40 00 80 06 E0 4F 83 6B 02 D6 83 6B   .(..@....O.k...k
00020:  02 D3 04 0D 00 8B 00 7D 74 7F 00 10 EE 8E 50 10   .......}t....P.
00030:  22 38 18 EA 00 00 20 20 20 20 20 20              "8....
```



TCP Communication Flags

Standard TCP communications monitor the **TCP packet header** that holds the flags. These flags govern the connection between hosts, and give instructions to the system. The following are the TCP communication flags:

- Synchronize alias "**SYN**": SYN notifies transmission of a new sequence number
- Acknowledgement alias "**ACK**": ACK confirms receipt of transmission, and identifies next expected sequence number
- Push alias "**PSH**": System accepting requests and forwarding buffered data
- Urgent alias "**URG**": Instructs data contained in packets to be processed as soon as possible
- Finish alias "**FIN**": Announces no more transmissions will be sent to remote system
- Reset alias "**RST**": Resets a connection

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

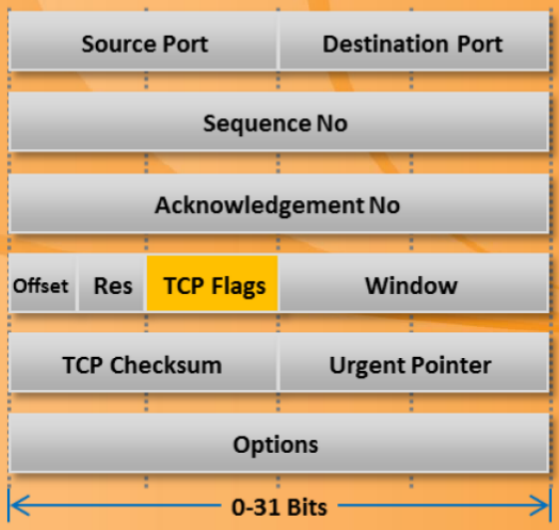
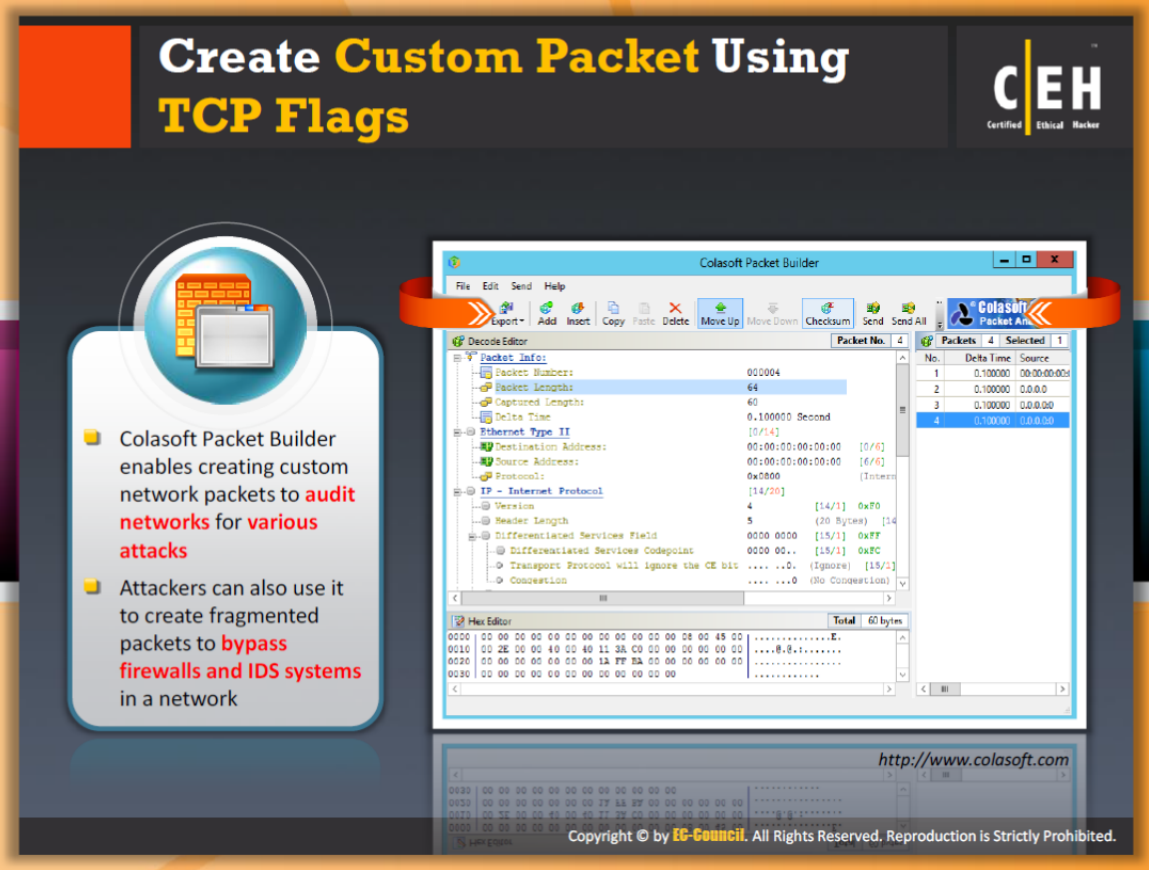


FIGURE 3.9: TCP Communication Flags



Create Custom Packets using TCP Flags

Source: http://www.colasoft.com

Colasoft Packet Builder is a tool that allows you to create **custom network packets** and also allows you to check the network against various attacks. It allows you to 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** also supports **saving** packets to packet files and **sending packets** to the network.

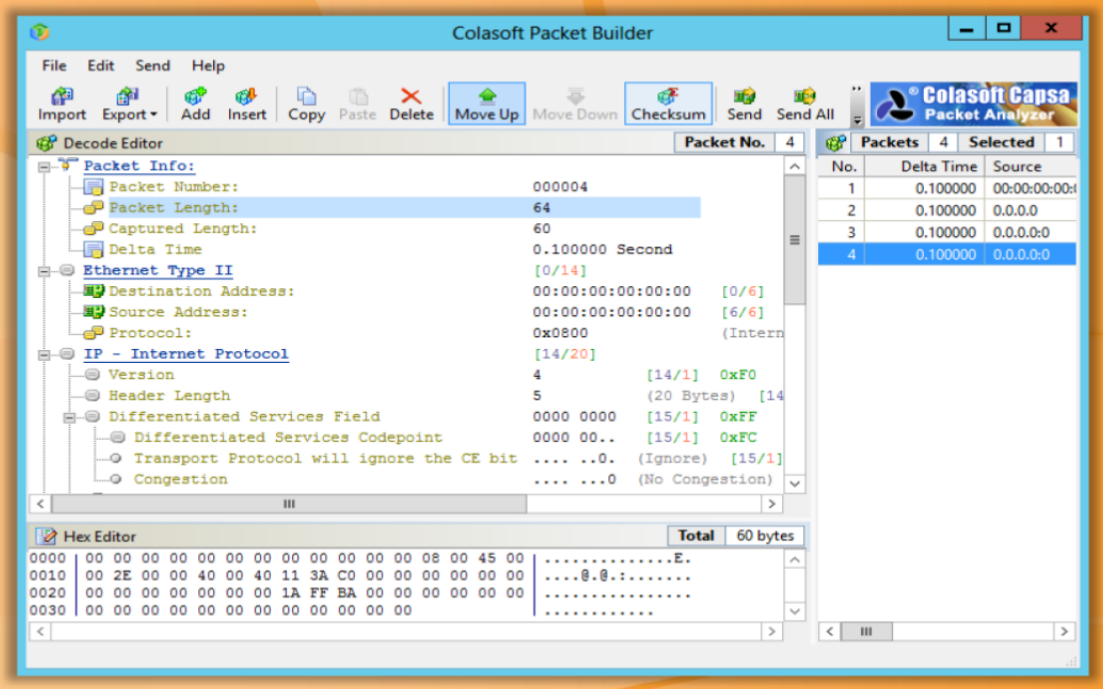
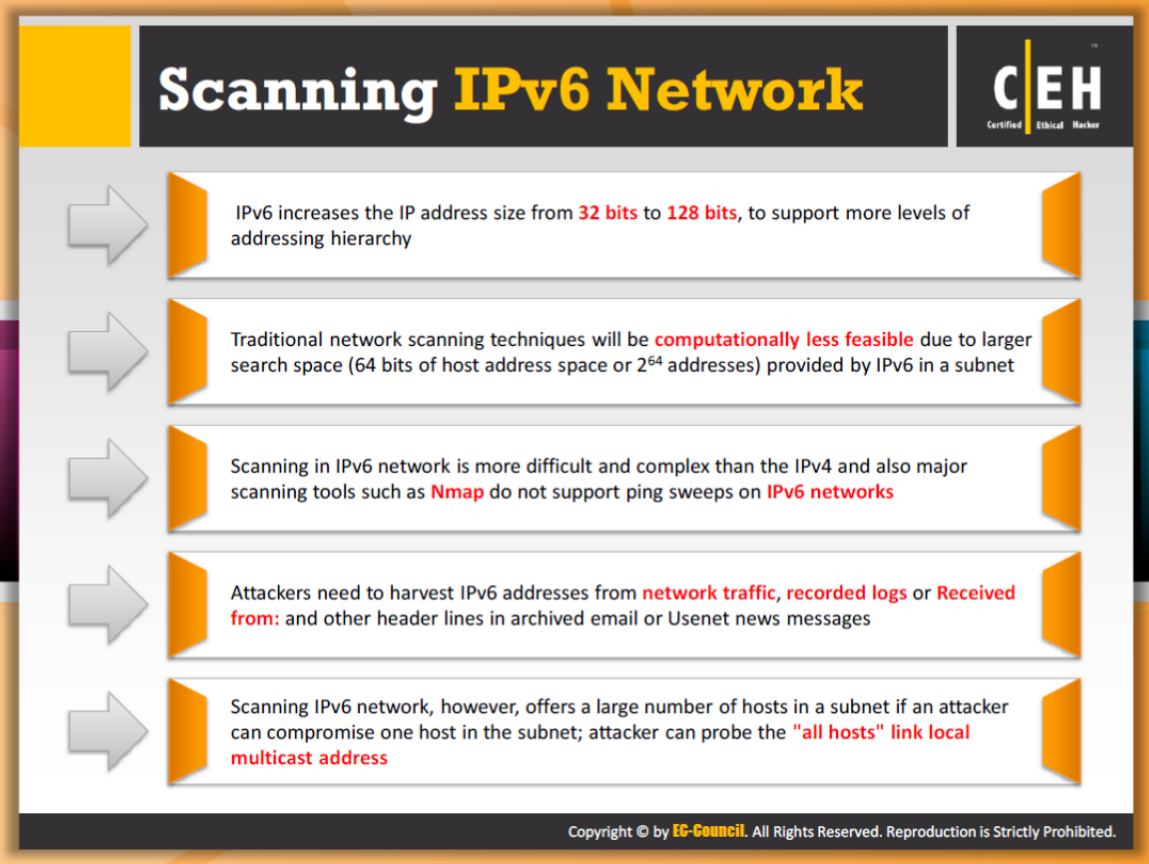
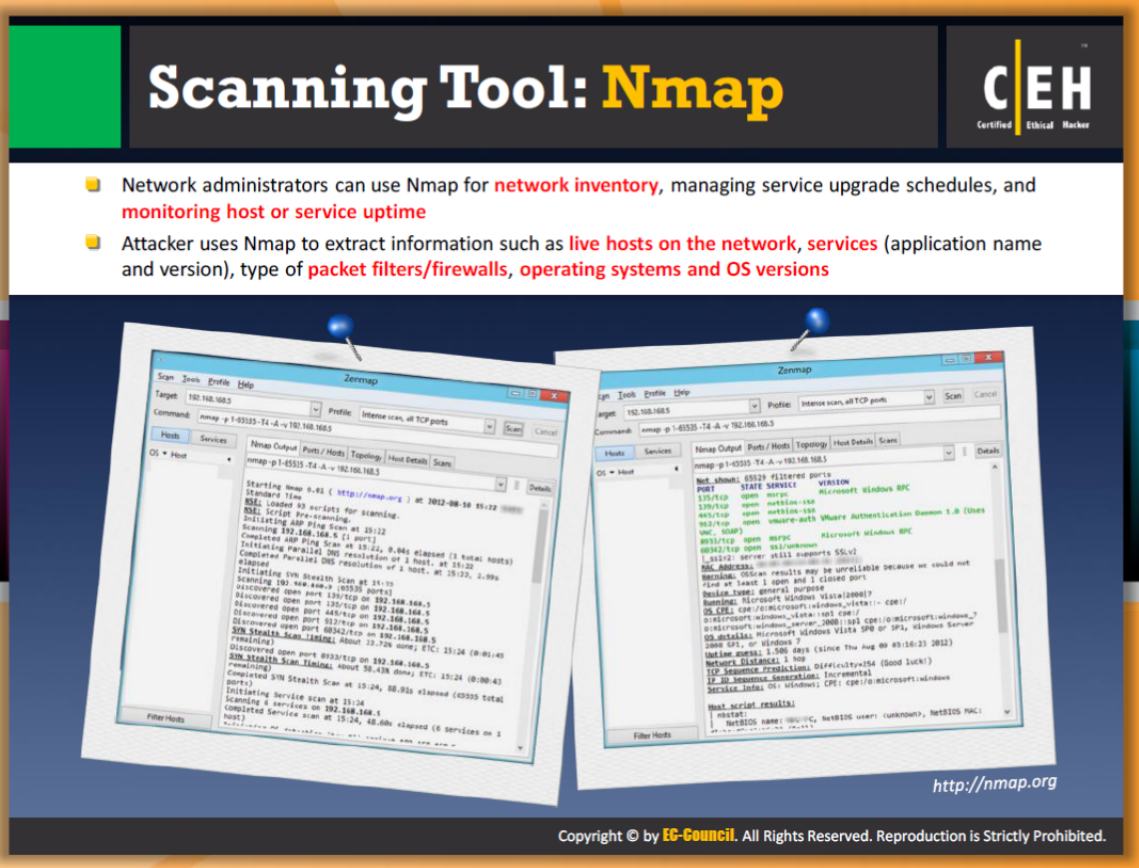


FIGURE 3.10: Colasoft Packet Builder Screenshot



Scanning IPv6 Network

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 will be computationally less feasible due to **larger search space** (64 bits of host address space or 264 addresses) provided by IPv6 in a subnet. Scanning an IPv6 network is more difficult and complex than IPv4 and also major scanning tools such as Nmap **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 IPv6 network, however, offers **a large number of hosts in a subnet**; if an attacker can compromise one host in the subnet he can probe the "all hosts" link local multicast address.



Scanning Tool: Nmap

Source: http://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 particular 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**.

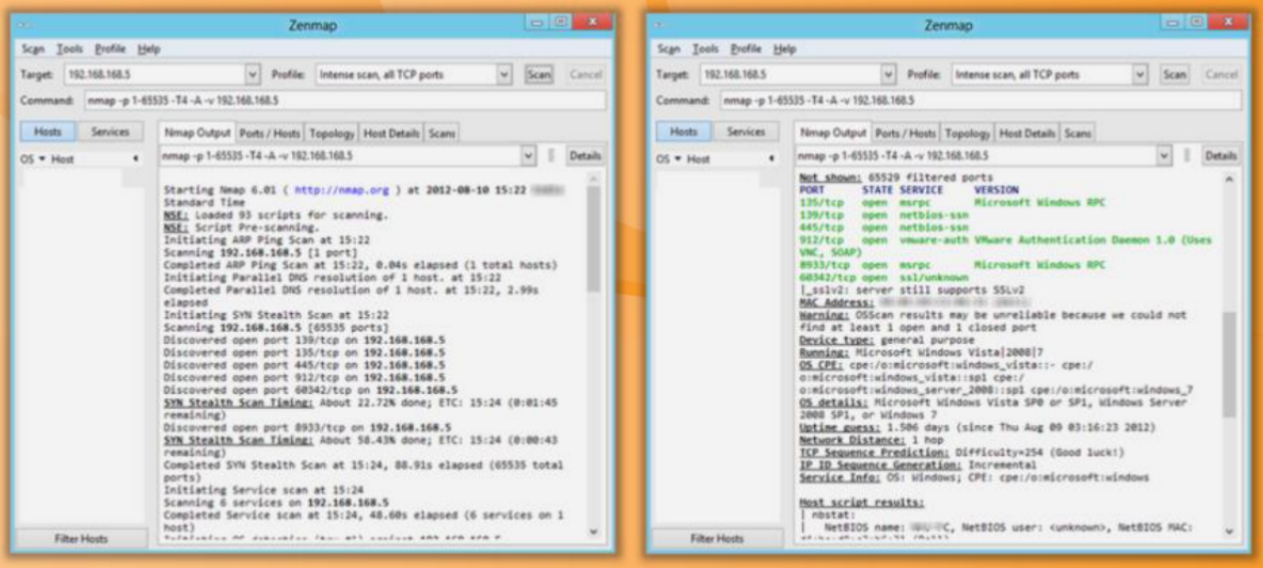


FIGURE 3.11: Zenmap Screenshots

Hping2/Hping3

Source: http://www.hping.org

HPing2/HPing3 is a **command-line-oriented TCP/IP** packet assembler/analyzer that sends ICMP echo requests and supports TCP, UDP, ICMP, and raw-IP protocols. It has **Traceroute mode**, and enables you to send files between covert channels. It has the ability to send custom TCP/IP packets and display target replies like a ping program does with **ICMP replies**. It handles fragmentation, arbitrary packets' body and size, and can be used in order 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.

Features:

The following are some of the features of HPing2/HPing3:

- Determines whether the host is up even when the host blocks ICMP packets
- **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
- 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.

This protocol is used by operating system, router, switch, internet-protocol-based devices via the **ping command** to **Echo request** and **Echo response** as a connectivity tester between different hosts.

The following screenshot shows ICMP scanning using the Hping3 tool:

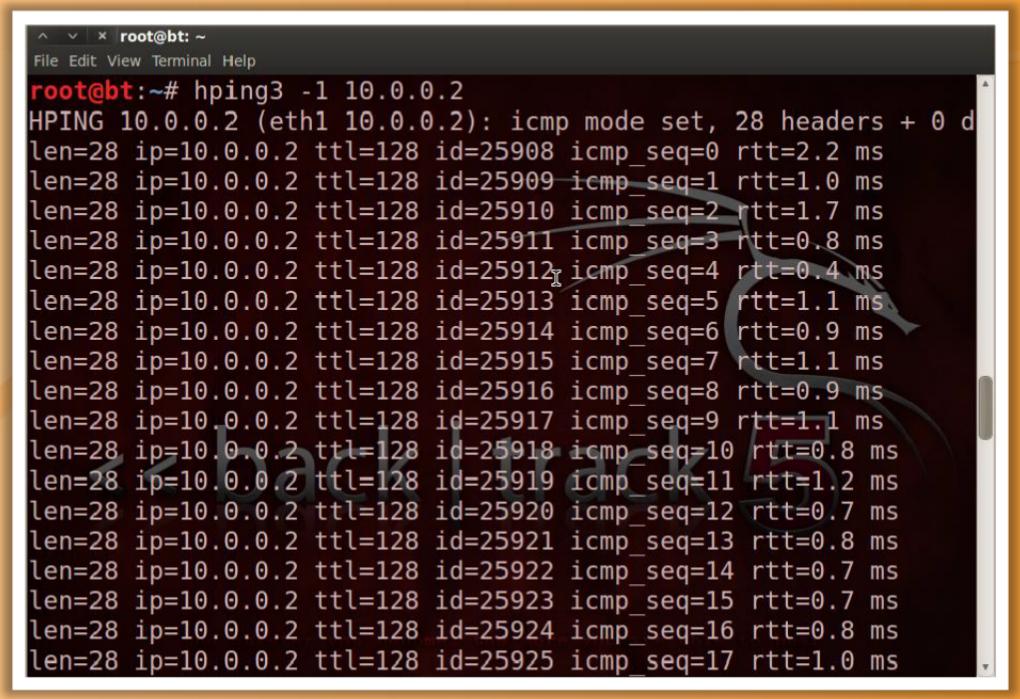


FIGURE 3.12: Hping3 tool showing ICMO scanning output

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 will allow you to establish connection (packets with the ACK bit set), whereas a sophisticated stateful firewall will not allow you to establish a connection.

The following screenshot shows ACK scanning on port 80 using the Hping3 tool:

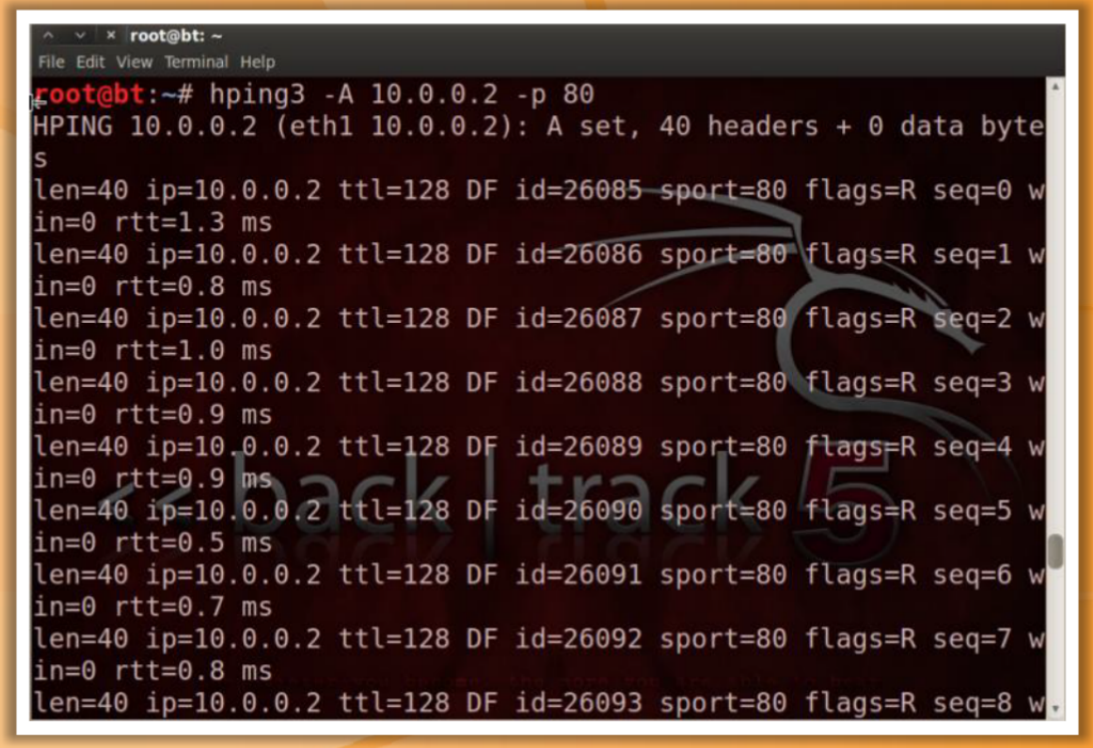
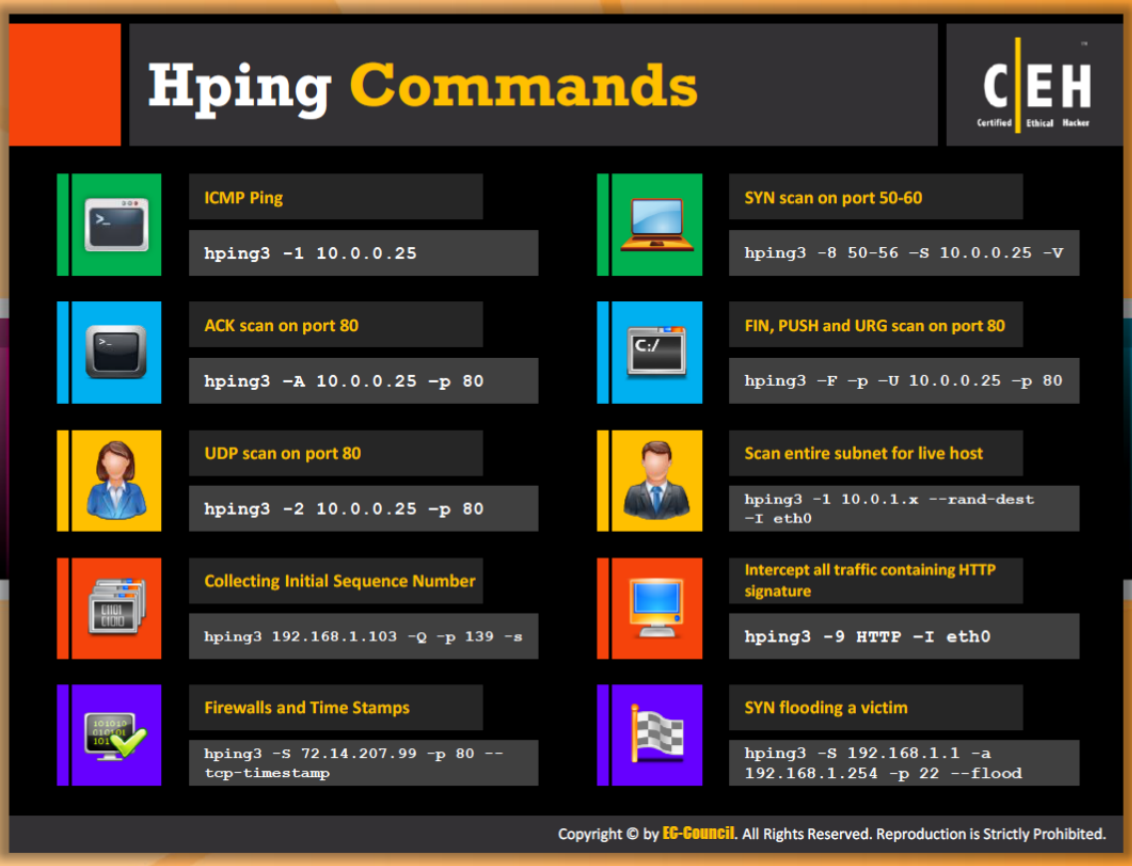


FIGURE 3.13: Hping3 tool showing ACK scanning output

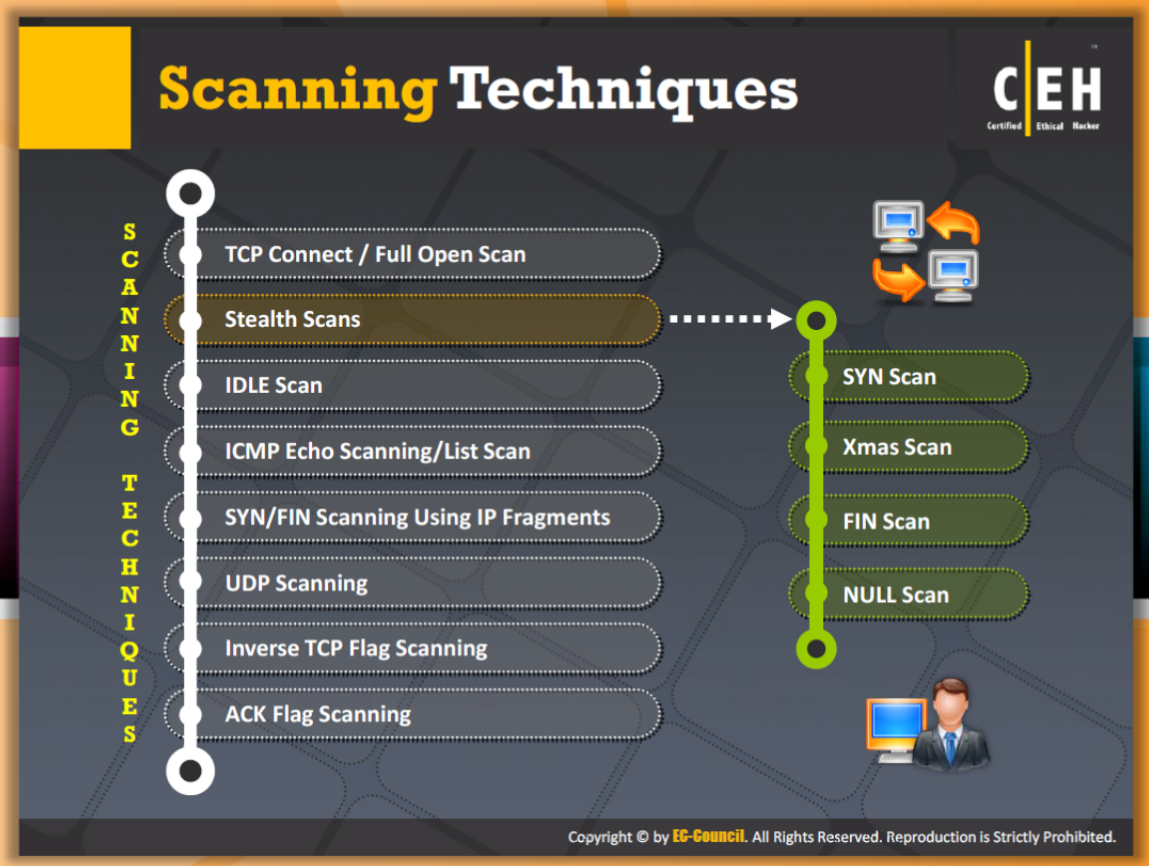


Hping Commands

The following table lists various scanning methods and 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 time stamps	`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 Commands Table



Scanning Techniques

Scanning is the process of **gathering information** about the systems that are alive and responding on the network.

The **port scanning techniques** are designed to identify the open ports on a targeted server or host. This is often used by administrators to verify security policies of their networks and by attackers to identify running services on a host with the intent of compromising it.

Different types of scanning techniques employed include:

- TCP Connect / Full Open Scan
- Stealth Scans: SYN Scan (Half-open Scan); XMAS Scan, FIN Scan, NULL Scan
- IDLE Scan
- ICMP Echo Scanning/List Scan
- SYN/FIN Scanning Using IP Fragments
- UDP Scanning
- Inverse TCP Flag Scanning
- ACK Flag 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
nicname	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

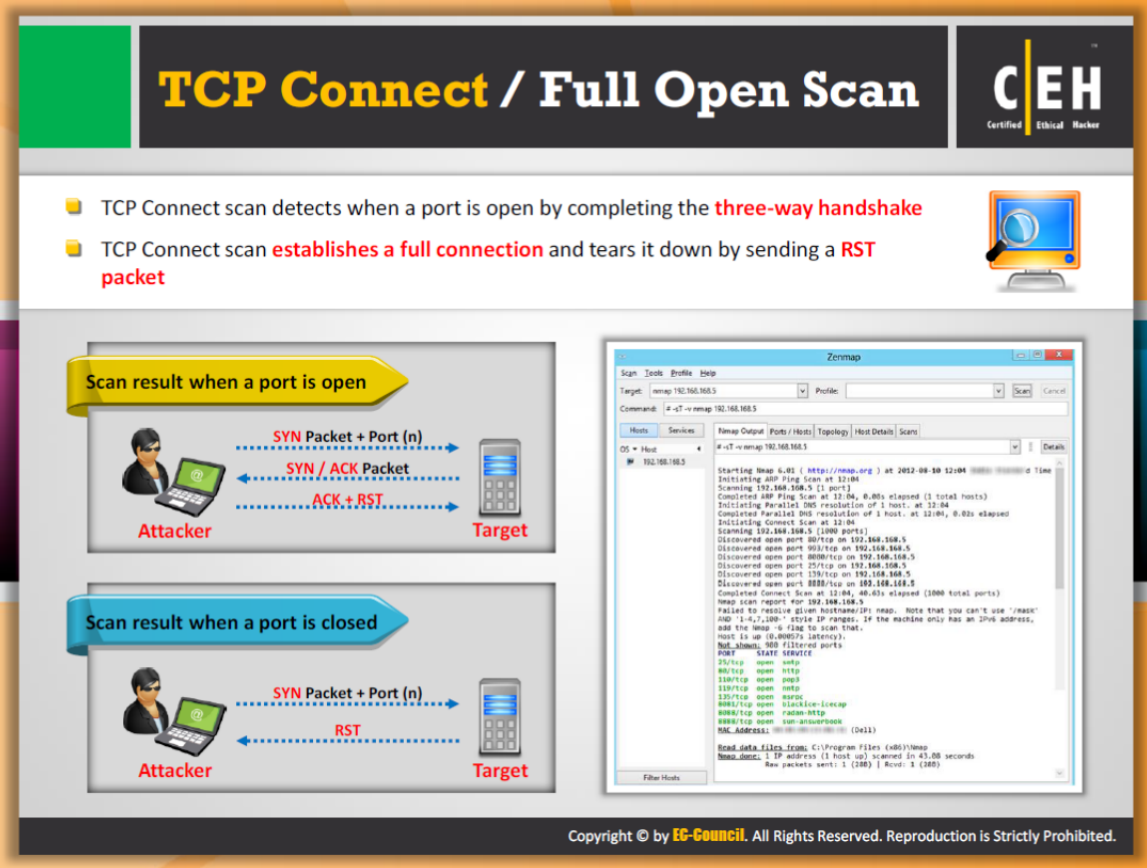
imap	143/udp	Internet Message Access Protocol
sql-net	150/tcp	SQL-NET
sql-net	150/udp	SQL-NET
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
snmp	161/tcp	
snmp	161/udp	
snmp-trap	162/tcp	
snmp-trap	162/udp	
cmip-man	163/tcp	CMIP/TCP Manager
cmip-man	163/udp	CMIP
cmip-agent	164/tcp	CMIP/TCP Agent
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	
ipx	213/udp	
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
54erberos54-ds	445/tcp	
54erberos54-ds	445/udp	
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
56erberos-adm	749/tcp	Kerberos Administration
56erberos-adm	749/udp	Kerberos Administration
kerberos	750/tcp	kdc Kerberos authentication—tcp
kerberos	750/udp	Kerberos
56erberos_master	751/udp	Kerberos authentication
56erberos_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
Name	Port/Protocol	Description
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	
afs	7000-7009/udp	

TABLE 3.2: Reserved Ports Table



TCP Connect / Full Open Scan

Source: http://www.insecure.org

TCP Connect / Full Open Scan is one of the most **reliable** forms of **TCP scanning**. The TCP connect() system call provided by an OS is used to open a connection to every interesting port on the machine. If the port is listening, connect() will succeed; otherwise, the port isn't reachable.

TCP Three-way Handshake

In the TCP three-way handshake, the client sends a **SYN flag**, which is acknowledged by a **SYN+ACK** flag by the server which, in turn, is acknowledged by the client with an ACK flag to complete the connection. You can establish a connection from both ends, and terminate from both ends individually.

Vanilla Scanning

In vanilla scanning, once the handshake is completed, the client ends the connection. If the connection is not established, then the scanned machine will be **DoS'd**, which allows you to make a new socket to be **created**/**called**. This confirms you with an open port to be scanned for a running service. The process will continue until the maximum port threshold is reached.

If the port is closed the server responds with an RST+ACK flag (RST stands for "**Reset the connection**"), whereas the client responds with a RST flag and here ends the connection. This is created by a TCP connect () system call and will be identified instantaneously if the port is opened or closed.

Making separate connects() 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.

Disadvantages

The drawback of this type of scan is **easily detectable** and **filterable**. The logs in the target system will disclose the connection.

The Output

Initiating Connect () Scan against (172.17.1.23)

Adding open port 19/tcp

Adding open port 21/tcp

Adding open port 13/tcp

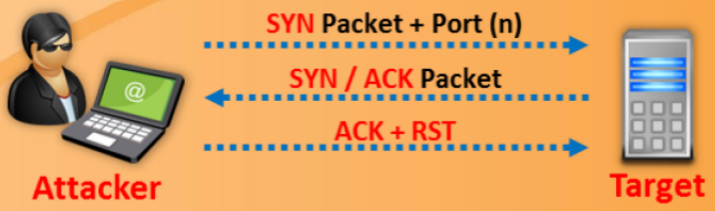


FIGURE 3.14: Scan results when a port is open

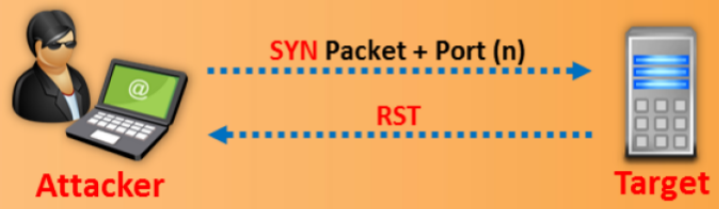


FIGURE 3.15: Scan results when a port is closed

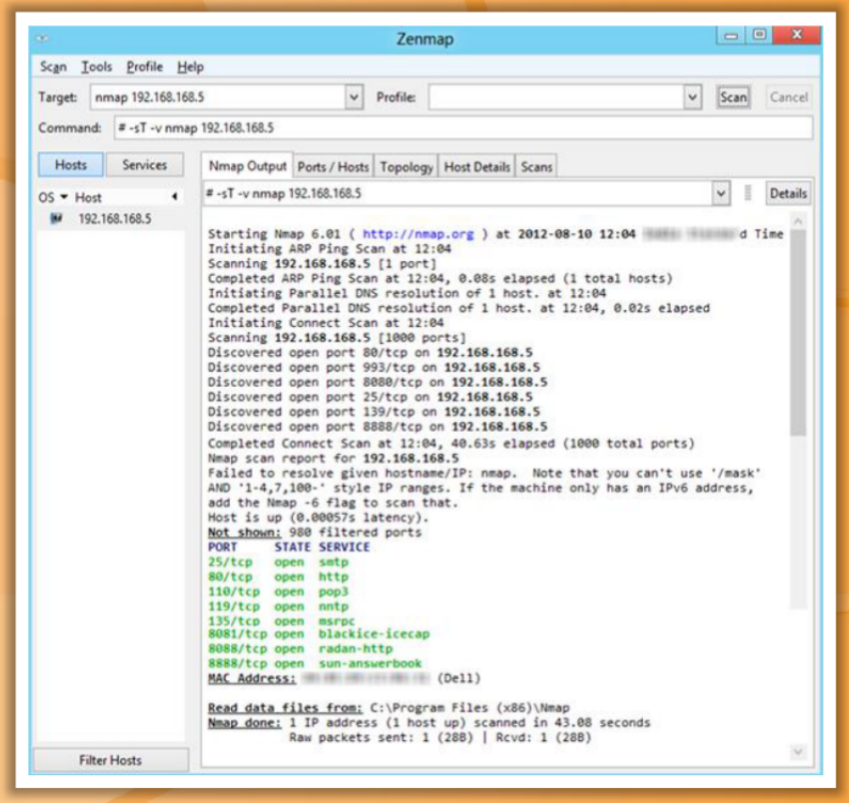
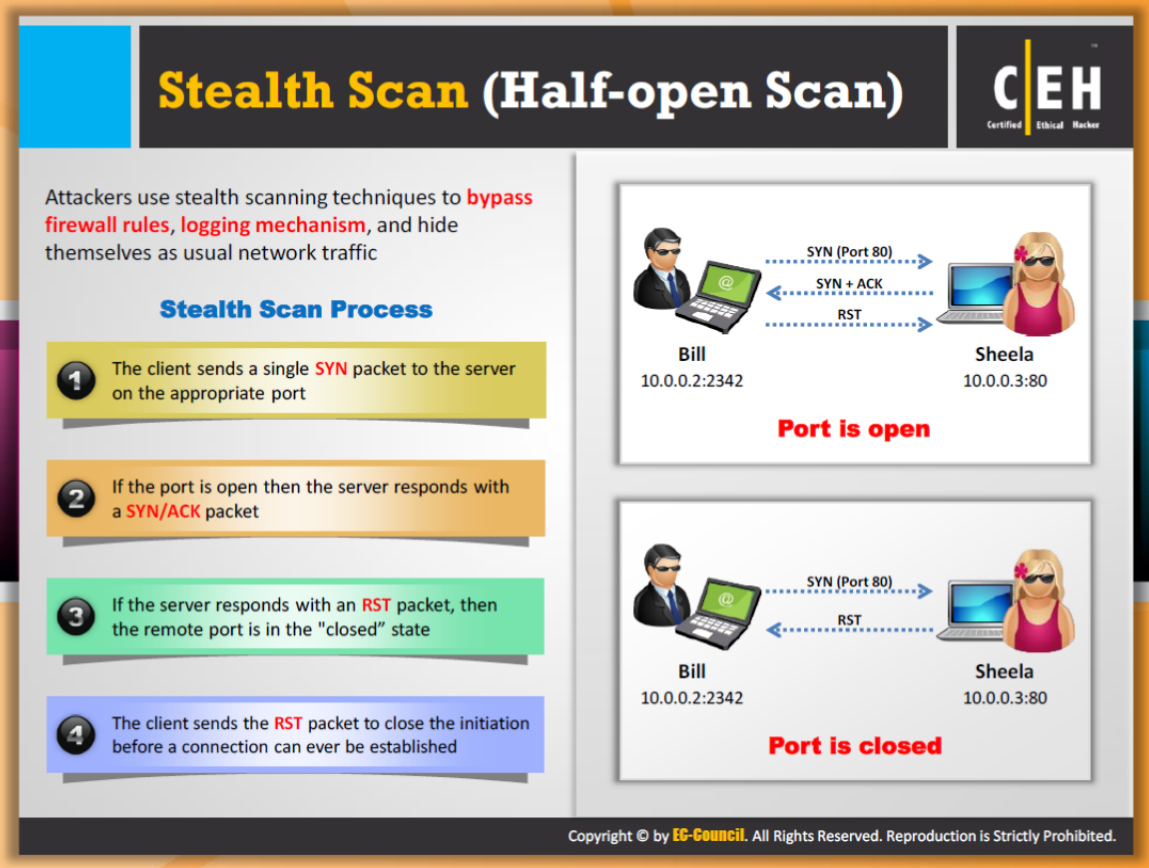


FIGURE 3.16: Zenmap Screenshot



Stealth Scan (Half-Open Scan)

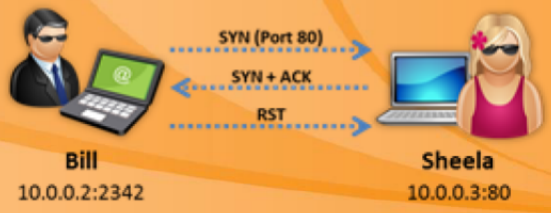
Stealth scan sends a **single frame** to a **TCP port** without any TCP handshaking or additional packet transfers. This is a scan type that sends a single frame with the expectation of a single response. The half-open scan partially opens a connection, but stops halfway through. This is also known as a SYN scan because it only sends the **SYN packet**. This stops the service from ever being notified of the incoming connection. **TCP SYN** scans or half-open scanning is a stealth method of port scanning.

The three-way handshake methodology is also implemented by the stealth scan. The difference is that in the last stage, remote ports are identified by examining the packets entering the interface and terminating the connection before a new initialization was triggered.

The process preludes the following:

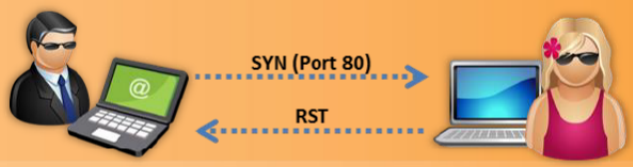
- To start initialization, the client forwards a single "**SYN**" packet to the destination server on the corresponding port.
- The server actually initiates the stealth scanning process, depending on the response sent.
- If the server forwards a "**SYN/ACK**" response packet, then the port is supposed to be in an "**OPEN**" state.

If the response is forwarded with an "RST" packet, then the port is supposed to be in a "CLOSED" state.



Port is open

FIGURE 3.16: Stealth Scan when Port is Open



Port is closed

FIGURE 3.17: Stealth Scan when Port is Closed

Zenmap Tool

Zenmap is the official **graphical user interface** (**GUI**) for the **Nmap** Security Scanner. Using this tool you can save the frequently used scans as profiles to make them easy to run recurrently. It contains a command creator that allows you to interact and create Nmap command lines. You can save the Scan results and view them in the future and they can be compared with another scan report to locate differences. The results of the recent scans can be stored in a **searchable database**.

The advantages of Zenmap are as follows:

- Interactive and graphical results viewing
- Comparison
- Convenience
- Repeatability
- Discoverability

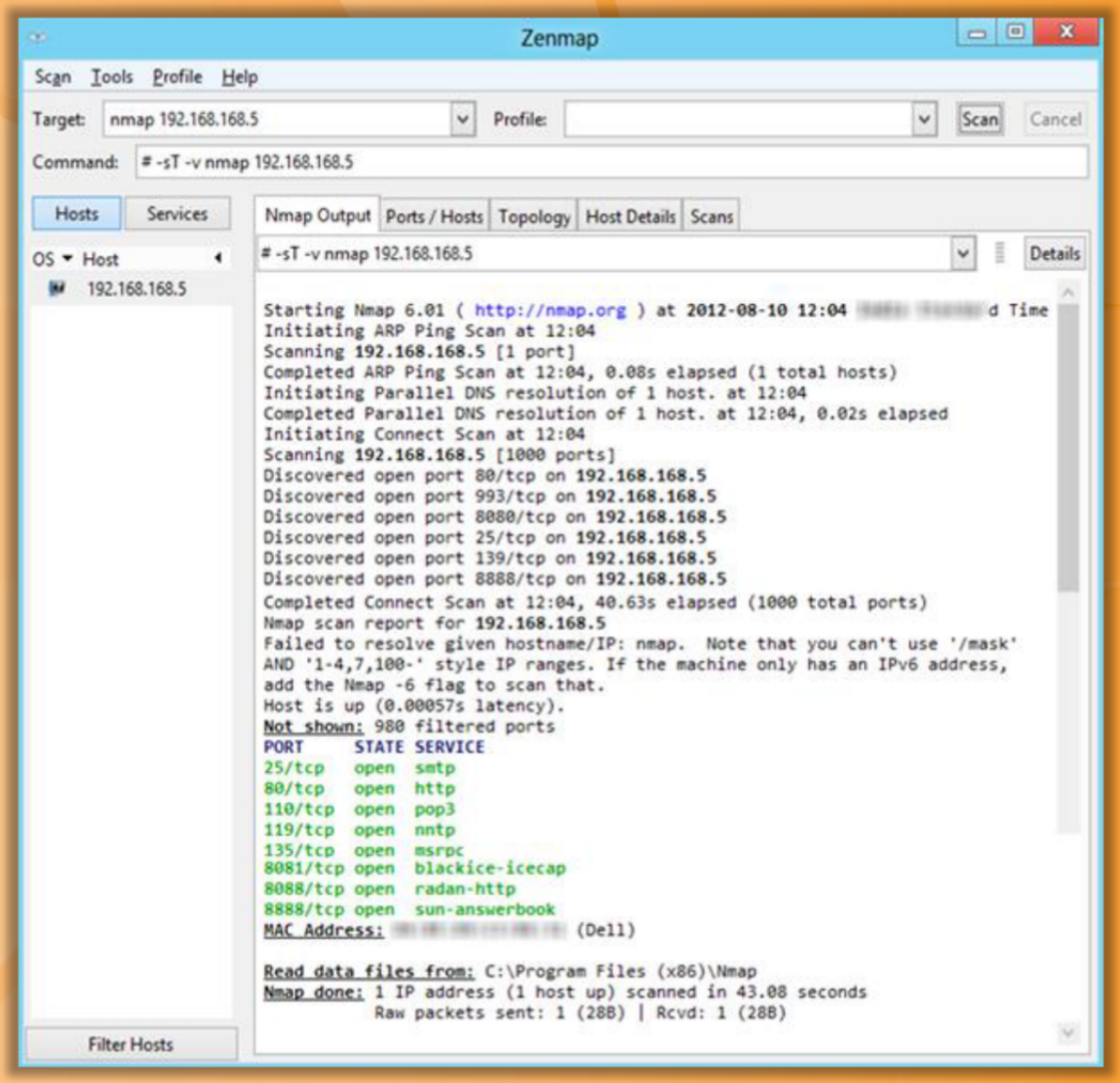
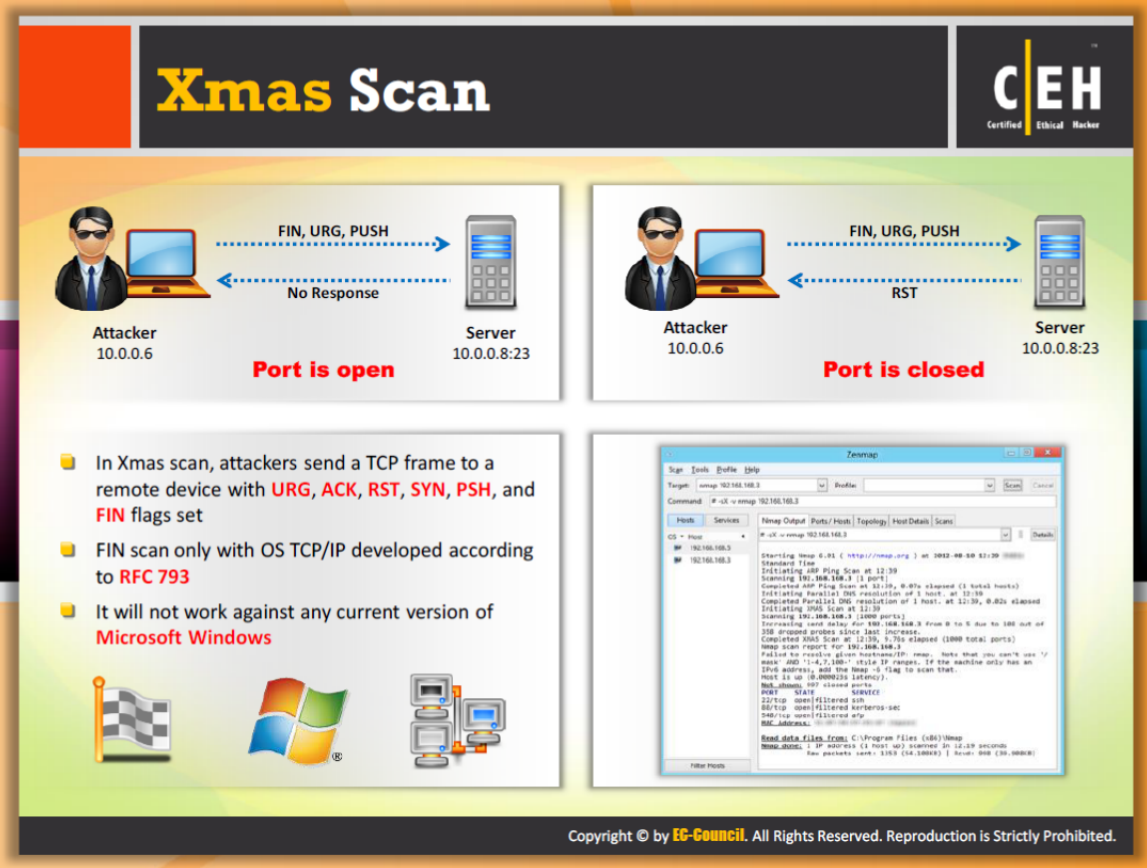


FIGURE 3.18: Zenmap Showing Scanning Results



Xmas Scan

Xmas Scan is a **port scan technique** with ACK, RST, SYN, URG, PSH, and FIN flags set to send a TCP frame to a remote device. If the target port is closed, then you will receive a remote system reply with a RST. You can use this port scan 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. This scan only works when systems are compliant with RFC 793.

BSD Networking Code

This method is based on BSD networking code; you can use this only for **UNIX hosts** and it does not support Windows NT. If this scan is directed at any Microsoft system, it shows all the ports on the host are opened.

Transmitting Packets

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

Advantage:

It avoids the IDS and TCP three-way handshake.

Disadvantage:

It works on the UNIX platform only.

FIGURE 3.19: Xmas Scan when Port is Open

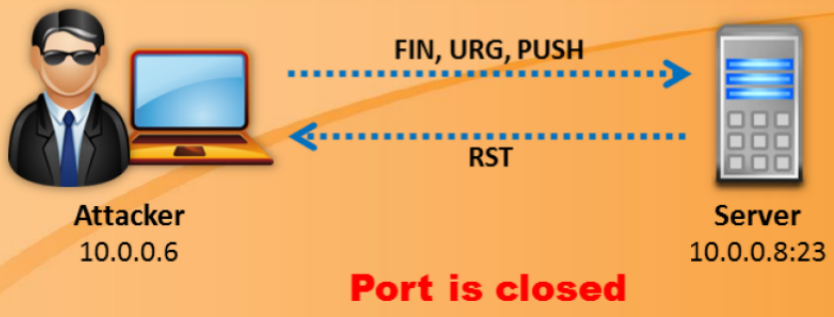


FIGURE 3.20: Xmas Scan when Port is Closed

Zenmap is the official graphical user interface (GUI) for the **Nmap Security Scanner**. Using this tool you can **save** the frequently used **scans** as **profiles** to make them easy to run recurrently.

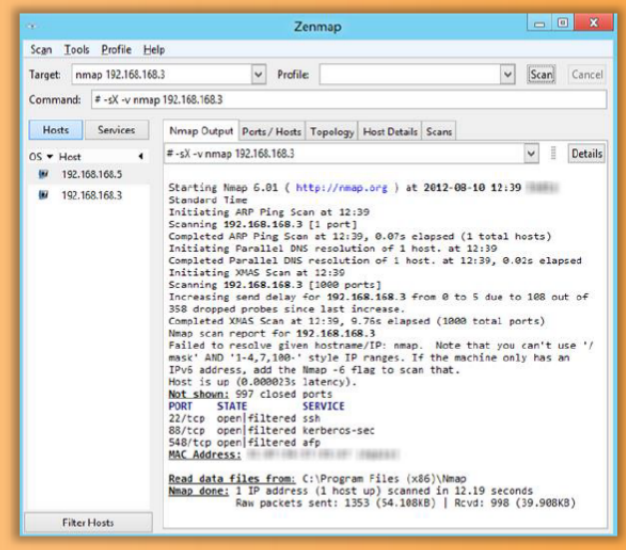
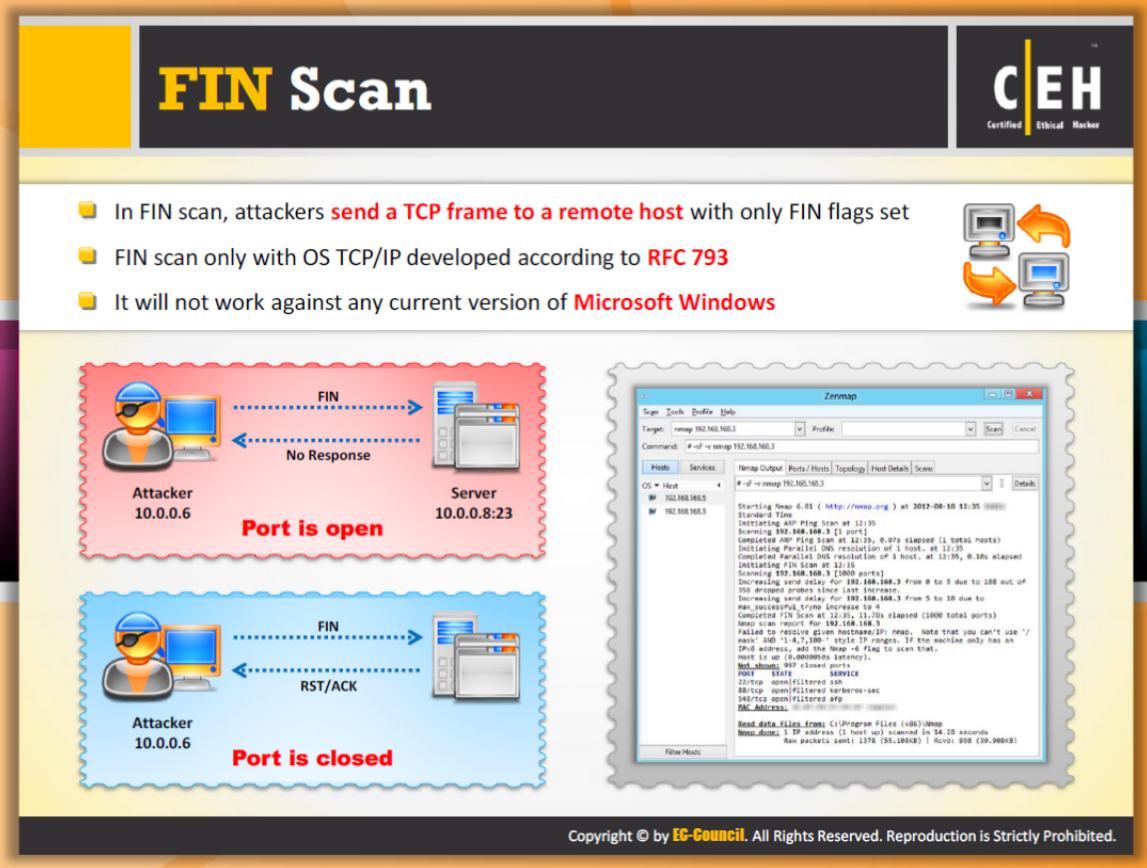


FIGURE 3.21: Zenmap Showing Xmas Scan Result



FIN Scan

FIN Scan is a type of port scan. The client sends a **FIN packet** to the target port, and if the service is not running or if the port is closed it replies to you with the probe packet with an RST.

FIGURE 3.22: FIN Scan when Port is Open

FIGURE 3.23: FIN Scan when Port is Closed

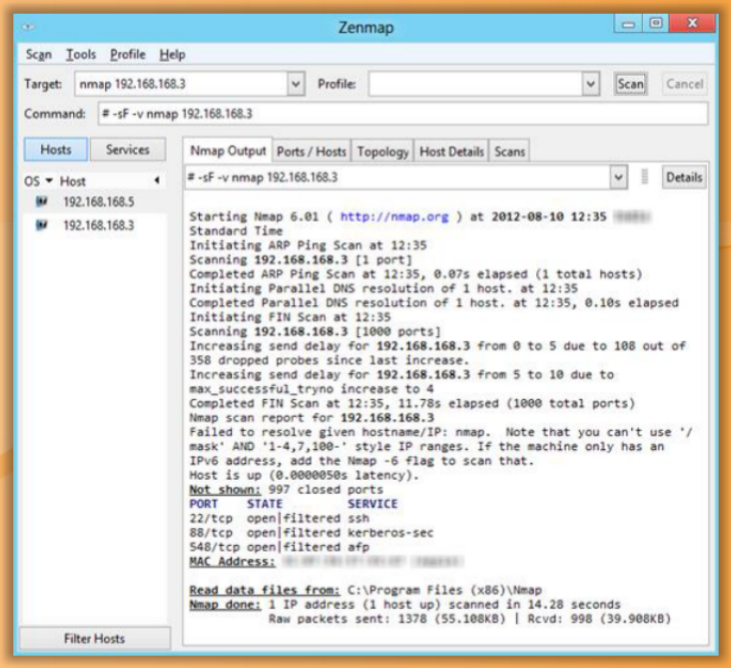
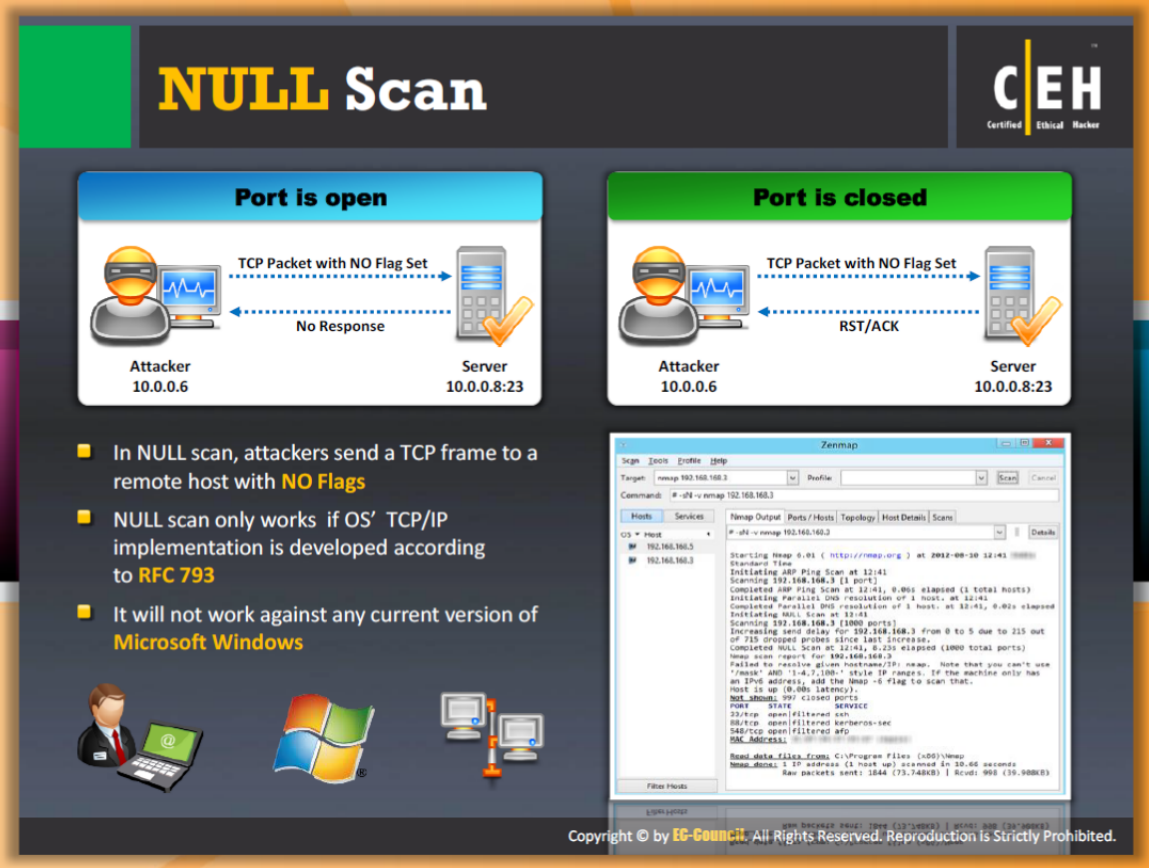


FIGURE 3.24: Zenmap showing FIN Scan Result



NULL Scan

NULL scans send TCP packets with **all flags turned off**. It is assumed that closed ports will return a TCP RST. Packets received by open ports are discarded as invalid.

It sets all flags of TCP headers, such as ACK, FIN, RST, SYN, URG and PSH, to NULL or unassigned. When any packets arrive at the server, BSD networking code informs the kernel to drop the incoming packet if a port is open, or returns an RST flag if a port is closed. This scan uses flags in the reverse fashion as the Xmas scan, but gives the same output as FIN and Xmas tree scans.

Many network codes of major operating systems can behave differently in terms of responding to the packet, e.g., Microsoft versus UNIX. This method does not work for Microsoft operating systems.

Command line option for null scanning with NMAP is "-sN"

Advantage:

It avoids IDS and TCP three-way handshake.

Disadvantage:

It works only for UNIX.

FIGURE 3.25: NULL Scan when Port is Open

FIGURE 3.26: NULL Scan when Port is Closed

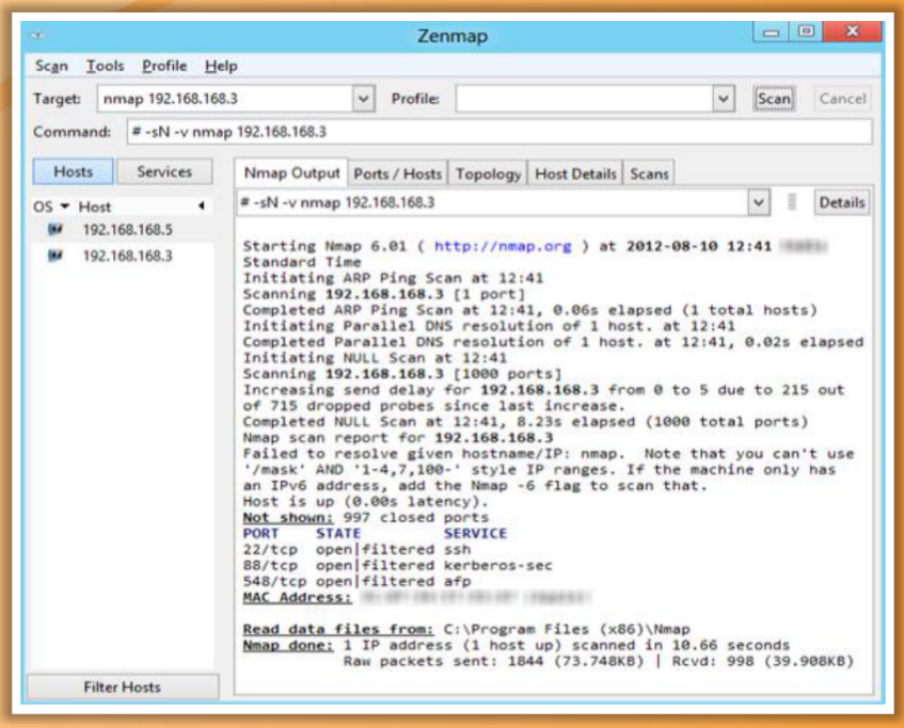
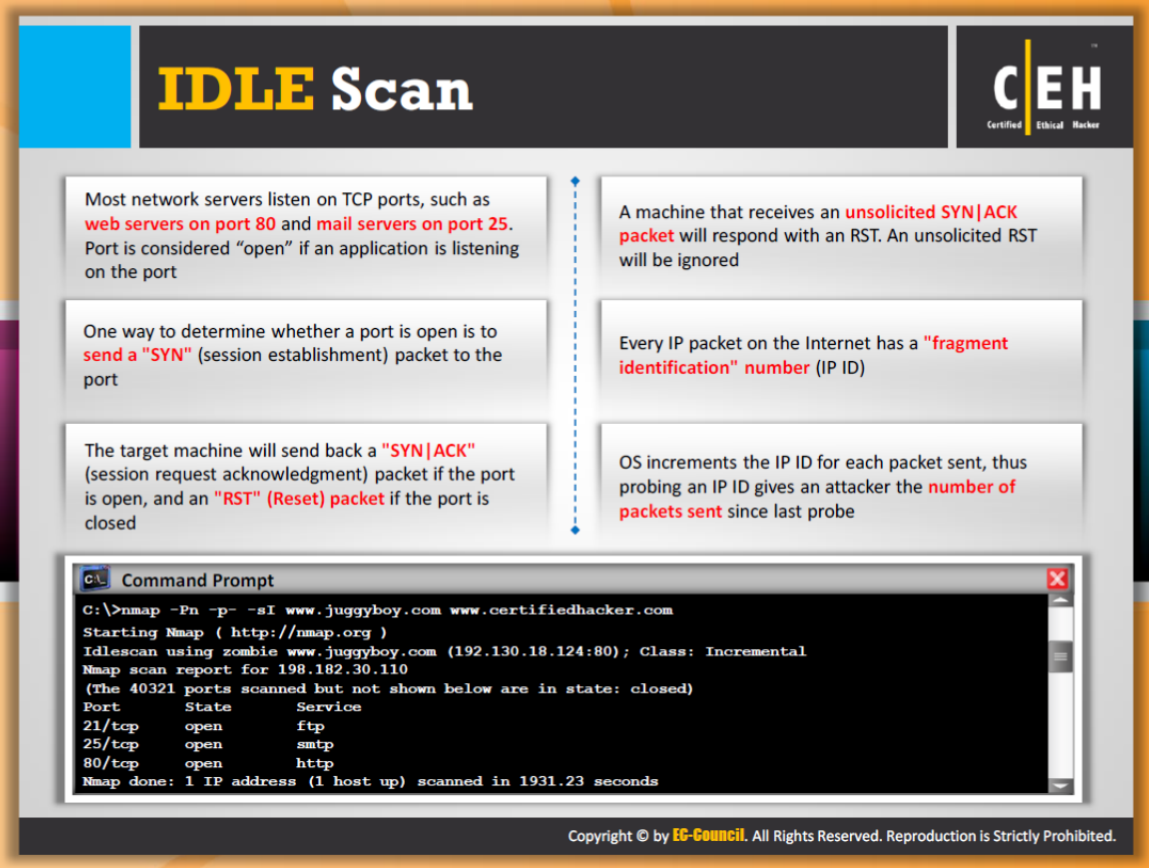


FIGURE 3.27: Zenmap showing NULL Scan Result



IDLE Scan

The idle 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 and offers complete blind scanning of a remote host. This is accomplished by impersonating another computer. No packet is sent from your own IP address; instead, another host is used, often called a "**zombie**," to scan the remote host and determine the open ports. This is done by expecting 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 show up.

Understanding TCP/IP

Source: http://nmap.org

Idle scanning is a sophisticated port scanning method. You do not need to be a TCP/IP expert to understand it. You need to understand the following basic facts:

- Most of the network servers listen on TCP ports, such as web servers on port 80 and mail servers on port 25. A port is considered "open" if an application is listening on the port; otherwise it is closed.

- To determine whether a port is open, send a session establishment “SYN” packet to the port. The target machine responds with a session request acknowledgment "SYN|ACK" packet if the port is open and a Reset “RST” packet if the port is closed.
- A machine that receives an unsolicited SYN|ACK packet responds with an RST. An unsolicited RST is ignored.
- Every IP packet on the Internet has a "fragment identification" number. Many operating systems simply increment this number for every packet they send. So probing for this number can tell an attacker how many packets have been sent since the last probe.

From these facts, it is possible to scan a target network while forging your identity so that it looks like an innocent "zombie" machine did the scanning.

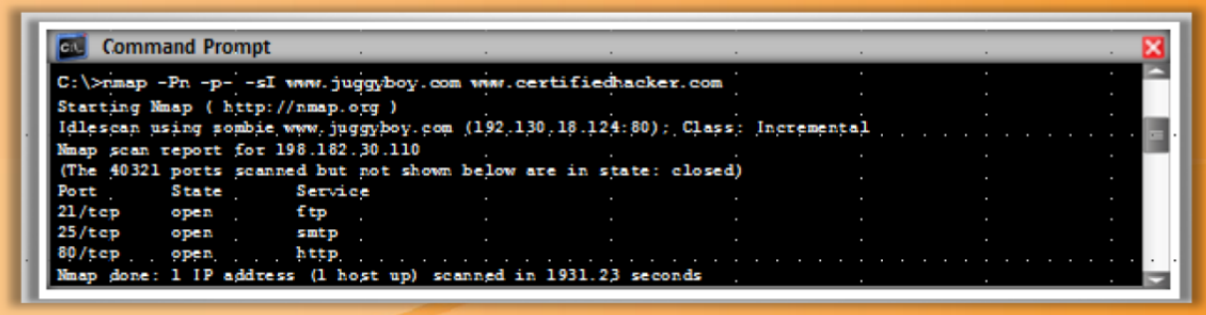
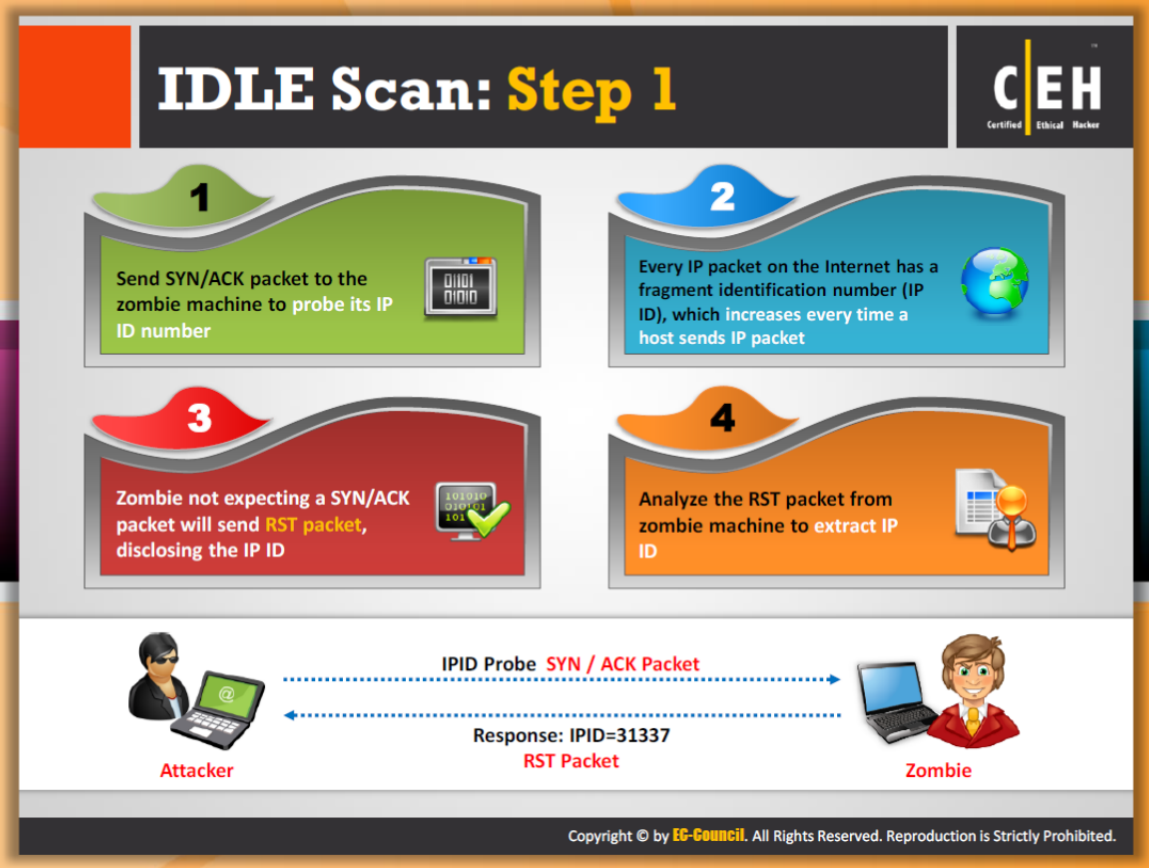


FIGURE 3.28: Nmap Showing Idle Scan Result



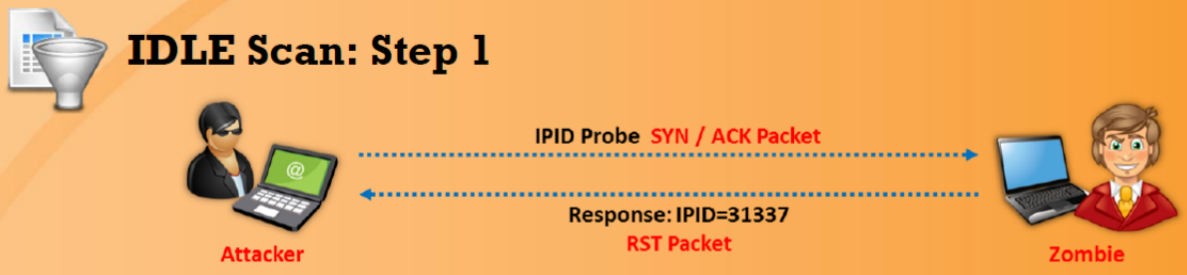
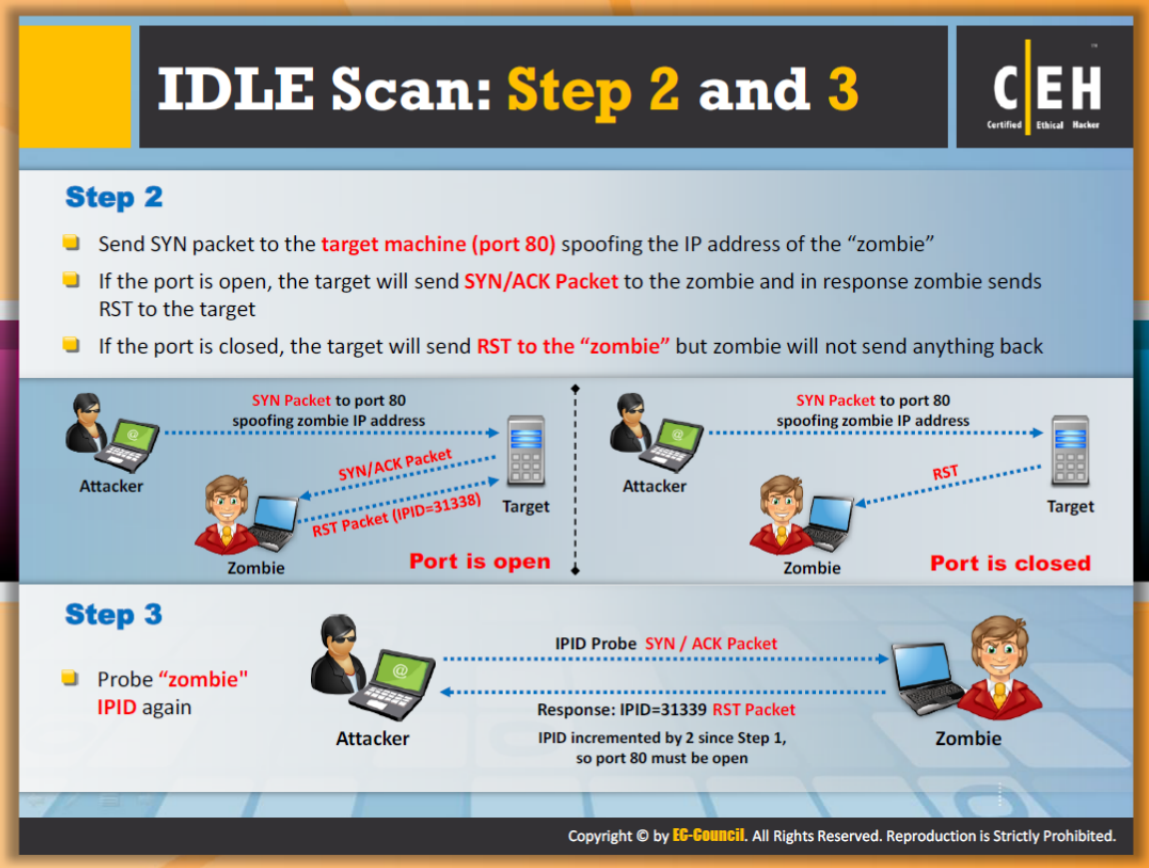


FIGURE 3.29: IPID Probe Request and Response

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

In the first step, you can send a session establishment "SYN" packet or **IPID** probe to determine whether a port is open or closed. If the port is open, the "zombie" responds with a session request acknowledgment "SYN|ACK" packet containing the IPID of the remote host machine. If the port is closed, it sends a reset "RST" packet. Every IP packet on the Internet has a "fragment identification" number, which is incremented by one for every packet transmission. In the above diagram, the zombie responds with **IPID=31337**.



IDLE Scan: Step 2 and 3

Idle Scan: Step 2.1 (Open Port)

Send a SYN packet to the target machine (port 80) spoofing the IP address of the "zombie." If the port is open, the target will send the **SYN/ACK packet** to the zombie and in response the zombie sends the RST to the target.

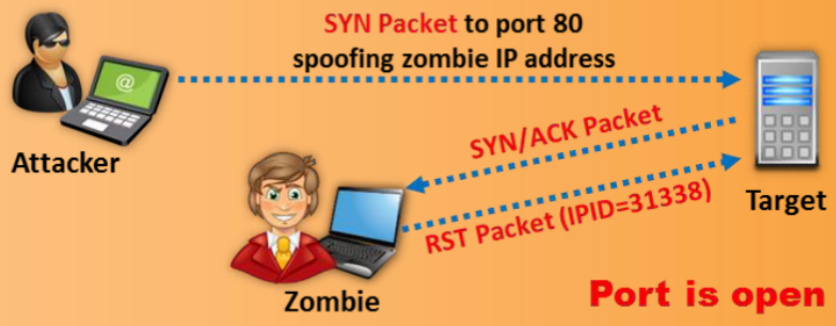


FIGURE 3.30: Target Response to Spoofed SYN Request when Port is Open

Idle Scan: Step 2.2 (Closed Port)

The target will send the RST to the "zombie" if the port is closed, but the zombie will

not send anything back.

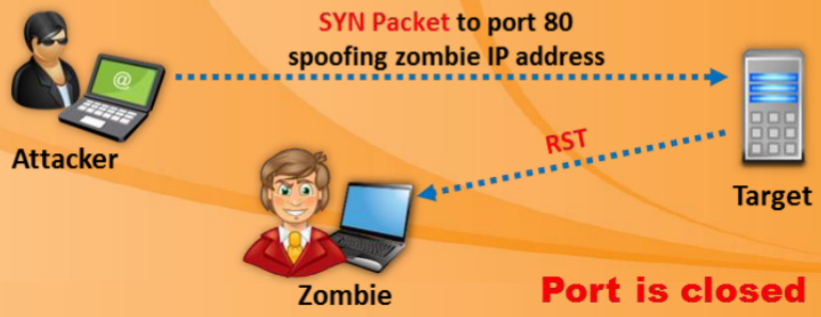


FIGURE 3.31: Target Response to Spoofed SYN Request when Port is Closed

Idle Scan: Step 3

Probe the “zombie" IPID again.

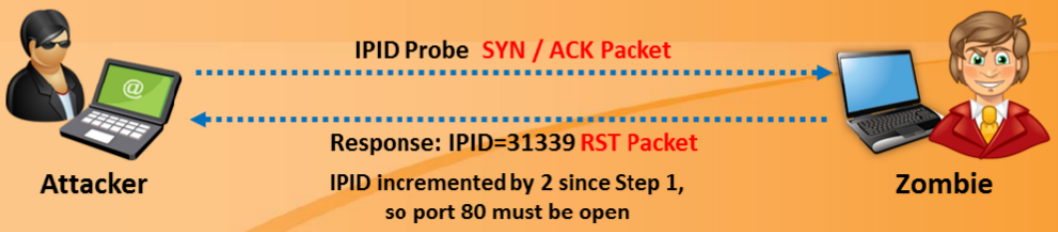
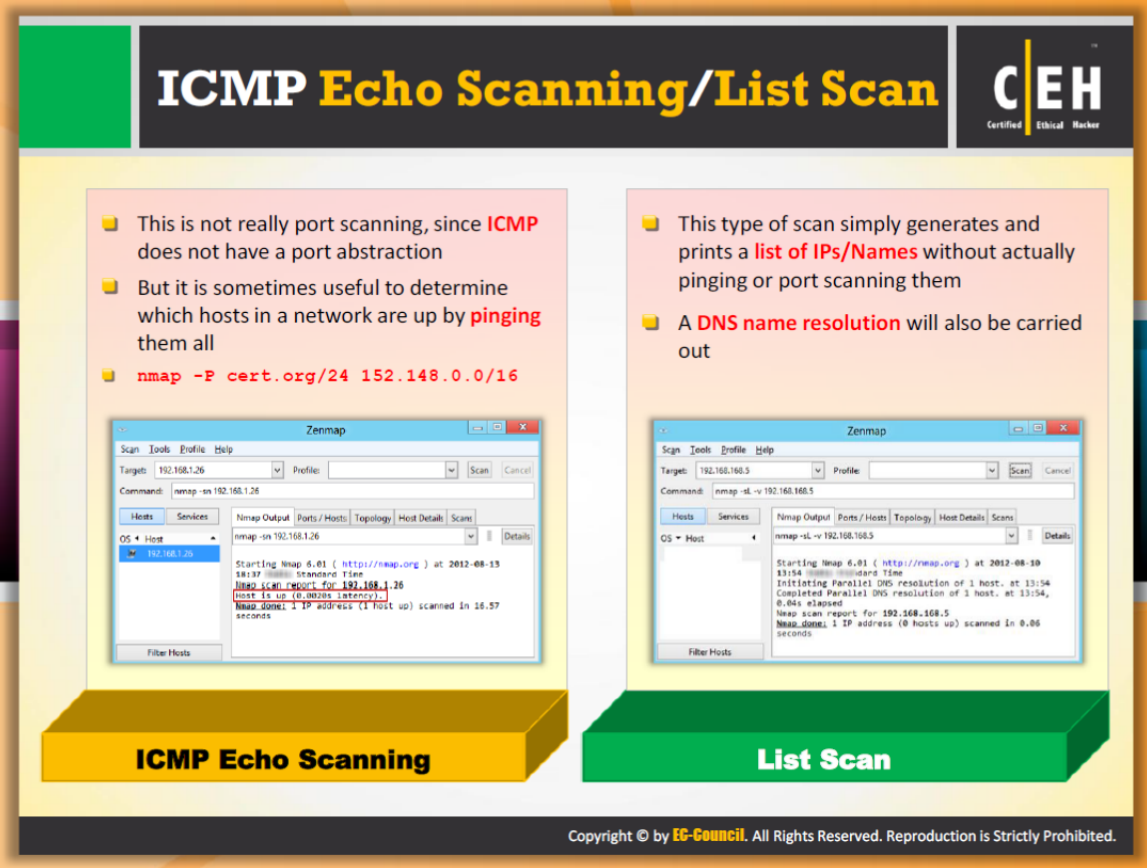


FIGURE 3.32: IPID Probe Request and Response



ICMP Echo Scanning/List Scan

ICMP echo scanning is used to **discover live machines by pinging all the machines** in the target network. Attackers send ICMP probes to the broadcast or network address which is relayed to all the host addresses in the subnet. The live systems will send ICMP echo reply message to the source of ICMP echo probe.

ICMP echo scanning is used in UNIX/Linux and BSD-based machines as the TCP/IP stack implementations in these operating system responds to the ICMP echo requests to the broadcast addresses. This technique cannot be used in Windows based networks as the TCP/IP stack implementation in windows machines is configured, by default, not to reply ICMP probes directed to the broadcast address.

ICMP echo scanning is not referred to as **port scanning** since it does not have a port abstraction. ICMP echo scanning is useful to determine which hosts in a network are active by pinging them all. The active hosts in the network is displayed in **Zenmap** as "Host is up **(0.020s latency)**." You can observe that in the screenshot:

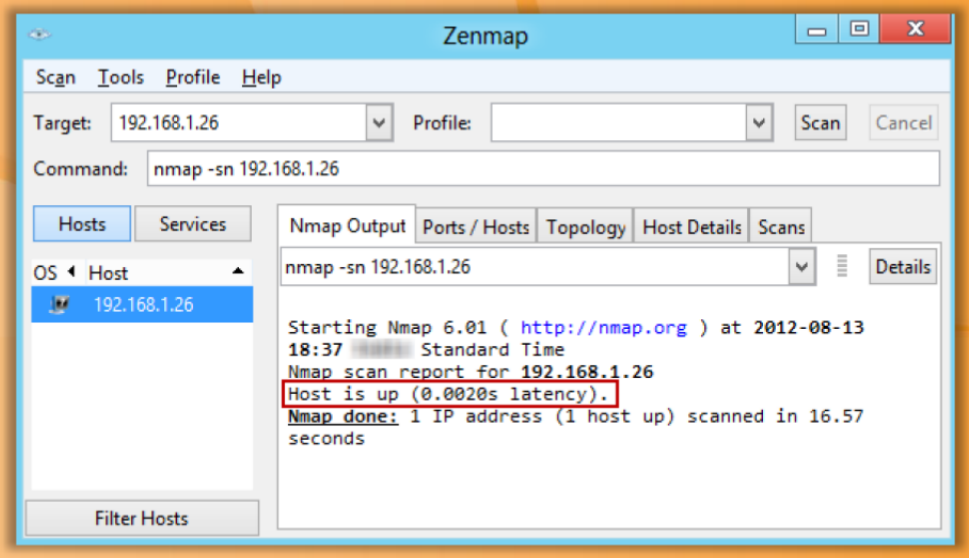


FIGURE 3.33: Zenmap showing ICMP Echo Scanning Result

In a list scan, discovery of the active host in the network is done indirectly. A list scan simply generates and prints a list of IPs/Names without actually pinging the host names or port scanning them. As a result, the list scan output of all the IP addresses will be shown as "not scanned," i.e., (0 hosts up). By default, a **reverse DNS** resolution is still being carried out on the host by Nmap for learning their names.

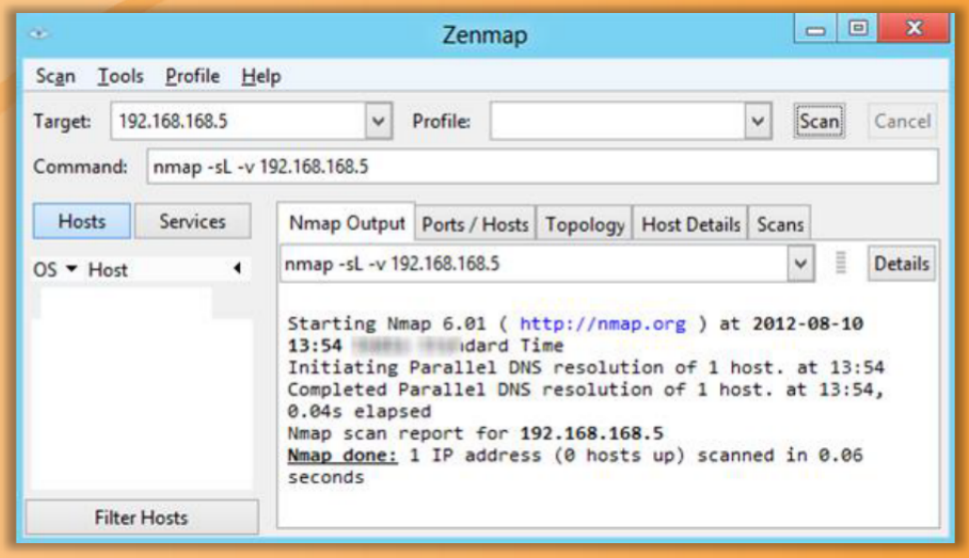
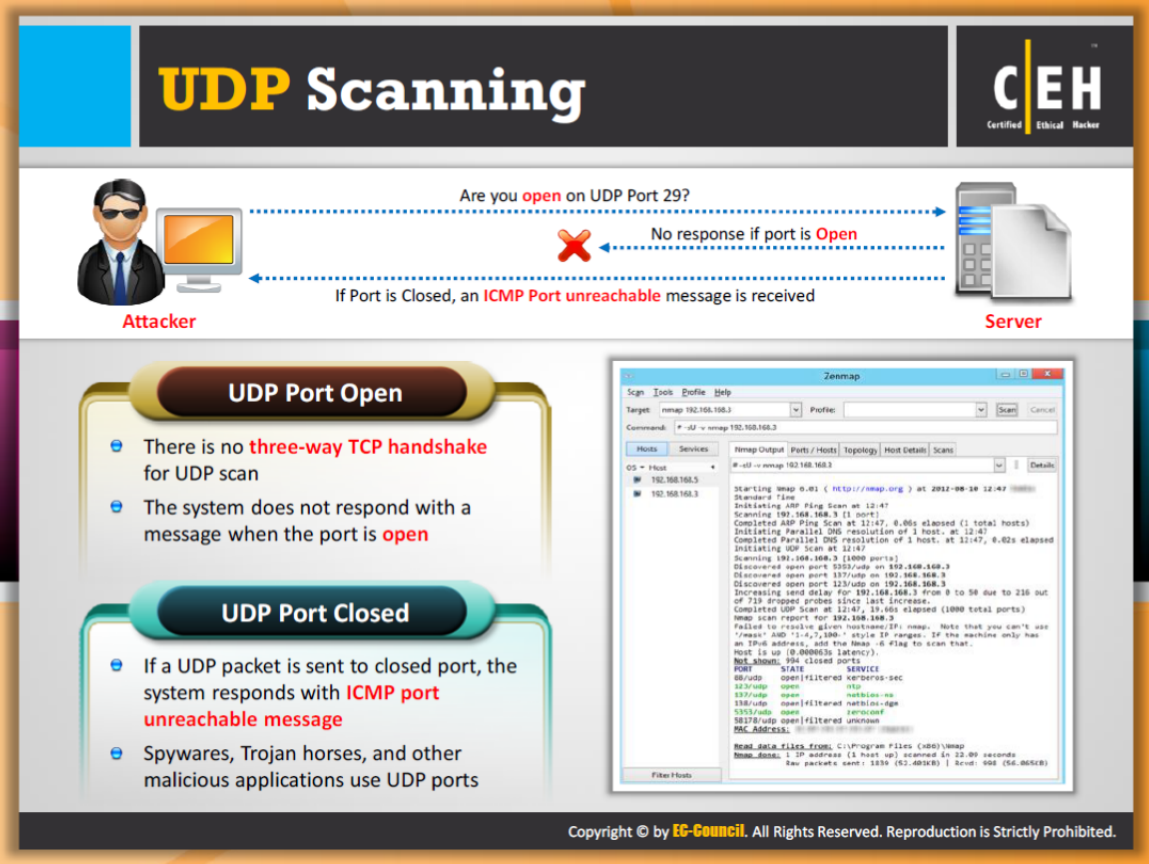


FIGURE 3.34: Zenmap showing List Scanning Result

Advantage:

- A list scan can perform a good sanity check.
- The incorrectly defined IP addresses on the command line or in an option file are detected by the list scan. The detected errors should be repairedprior to running any "active" scan.



UDP Scanning

UDP Raw ICMP Port Unreachable Scanning

UDP port scanners use the **UDP protocol** instead of TCP, and can be more difficult than TCP scanning. You can send a packet, but you cannot determine that the host is alive or dead or filtered. However, there is one ICMP that you can use to determine whether ports are open or closed. 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, then it's closed, while the ports that didn't answer are either open or filtered by 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: http://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 find out if a port is NOT open. Neither UDP packets nor the ICMP errors are guaranteed to arrive, so UDP scanners of this sort must also implement the retransmission of packets that appear lost. UDP scanners interpret lost traffic as open ports.

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

UDP RECVFROM () and WRITE () Scanning

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

Example

For example, a second write () call to a closed port will usually fail. A lot of scanners, such as Netcat and Pluvial pscan.c do recvfrom () on non-blocking UDP sockets, 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). Root users can also use the -I (lamer UDP scan) options to force this.

Advantage:

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

Disadvantage:

The UDP scan provides port information only. If additional version 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, so 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 that can be used to identify these invasions achieved by the attacker on open UDP ports caused by spyware applications, Trojan horses, and other malicious software.

FIGURE 3.35: UDP Scanning

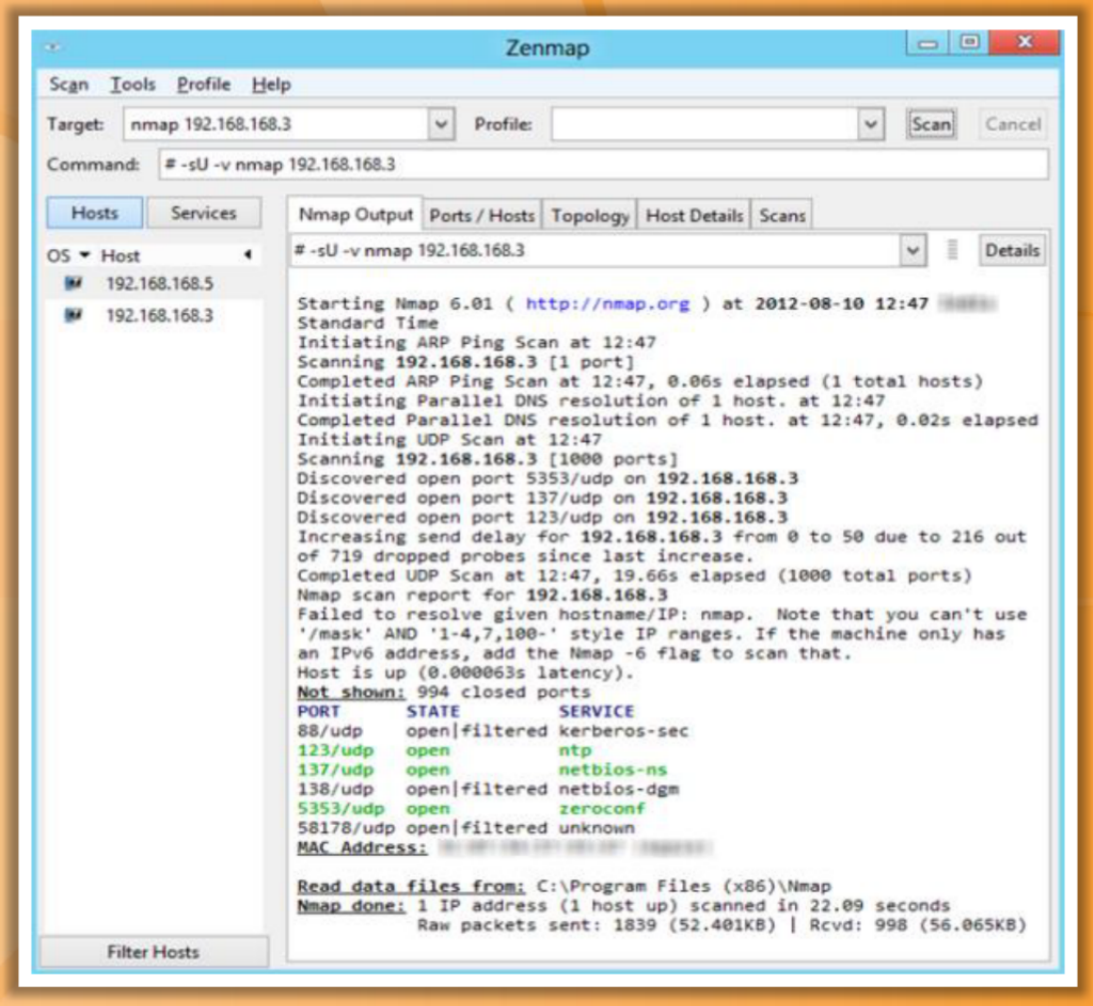
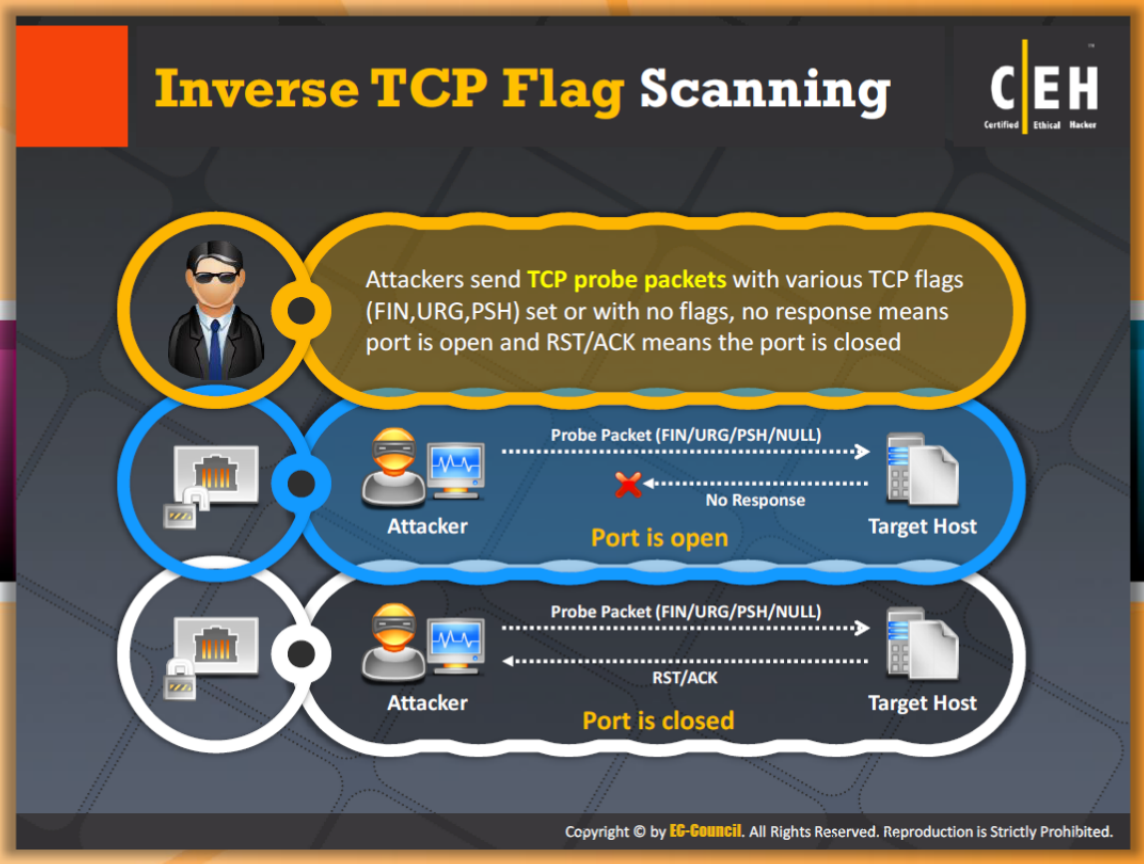


FIGURE 3.36: Zenmap showing UDP Scanning Result



Inverse TCP Flag Scanning

Attackers send the **TCP probe packets** by enabling various TCP flag (FIN, URG, PSH) or with no flags. When the port is open, the attacker doesn't get any response from the host, whereas when the port is closed, he or she receives the **RST/ACK** from the target host.

The SYN packets that are sent to the sensitive ports of the targeted hosts are detected by using security mechanisms such as **firewalls** and **IDS**. 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.

Probing a target using a half-open SYN flag is known as an inverted technique. It is called this because the closed ports can only send the response back. According to RFC 793, An RST/ACK packet must be sent for connection reset, when the port is closed on host side. 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 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 the closed ports on the targeted host will send an RST/ACK response. Since the RFC 793 standard is completely ignored in the operating system such as Windows, you cannot see the RST/ACK response when connected to the closed port on the target host. This technique is effective when used with UNIX-based operating systems.

Advantages

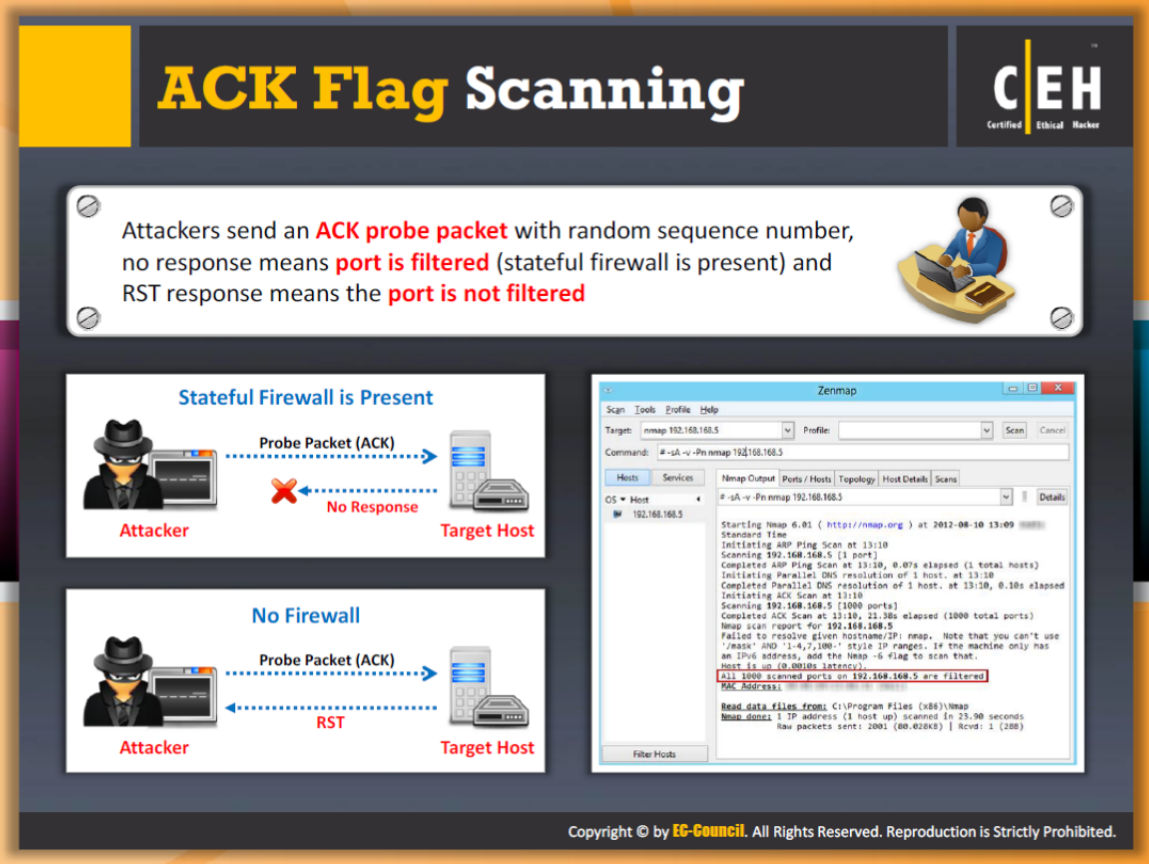
- Avoids many IDS and logging systems, highly stealthy

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)

FIGURE 3.37: Inverse TCP Flag Scanning when Port is Open

FIGURE 3.38: Inverse TCP Flag Scanning when Port is Closed



ACK Flag Scanning

A stealthy technique is used for **identifying open TCP ports**. In this technique a TCP packet with ACK flag ON is sent to the remote host and then the header information of the RST packets sent by remote host are analyzed. Using this technique one can exploit the potential vulnerabilities of BSD derived TCP/IP stack. This technique gives good results when used with certain operating systems and platforms.

ACK scanning can be performed in two ways:

- TTL field ananlysis
- WINDOW field analysis

Using TTL value one can determine the number of systems the TCP packet traverses. You can send an ACK probe packet with random sequence number: no response means port is filtered (state full firewall is present) and RST response means the port is not filtered.

```
nmap -sA -P0 10.10.0.25
Starting nmap 5.21 (http://nmap.org) at 2010-05-16 12:15 EST
All 529 scanned ports on 10.10.0.25 are: filtered
```

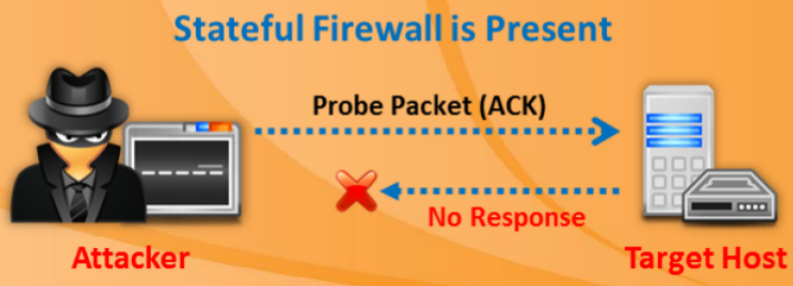


FIGURE 3.39: ACK Flag Scanning when Stateful Firewall is Present

FIGURE 3.40: ACK Flag Scanning when No Firewall is Present

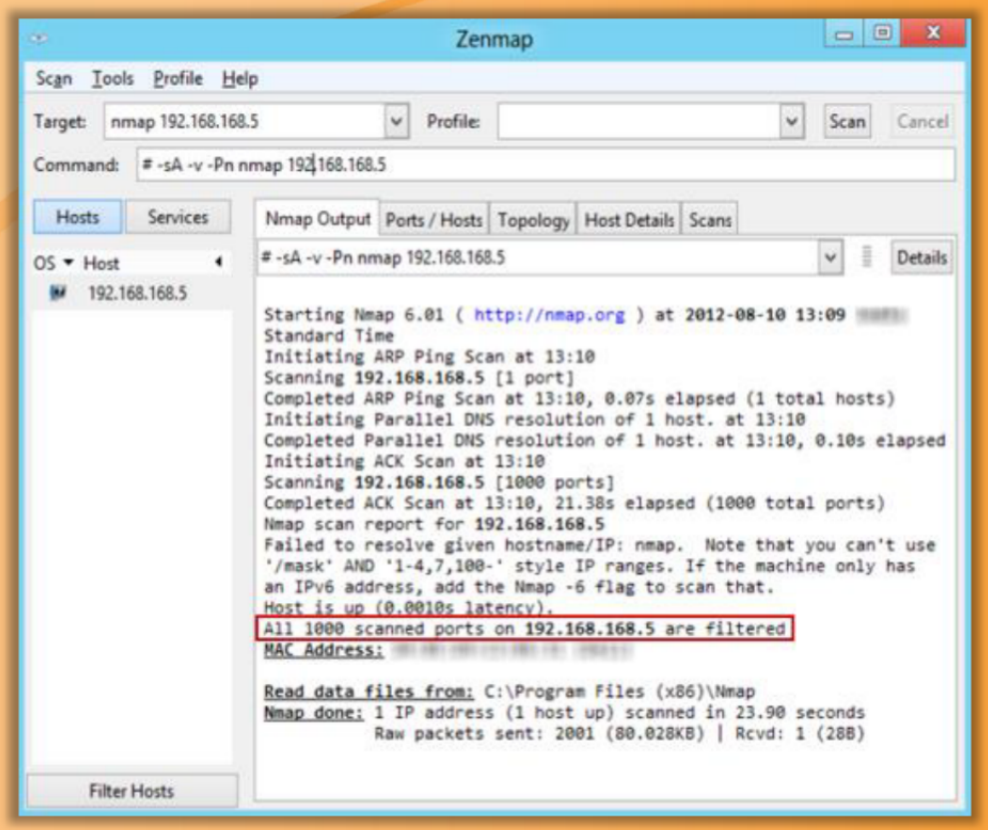
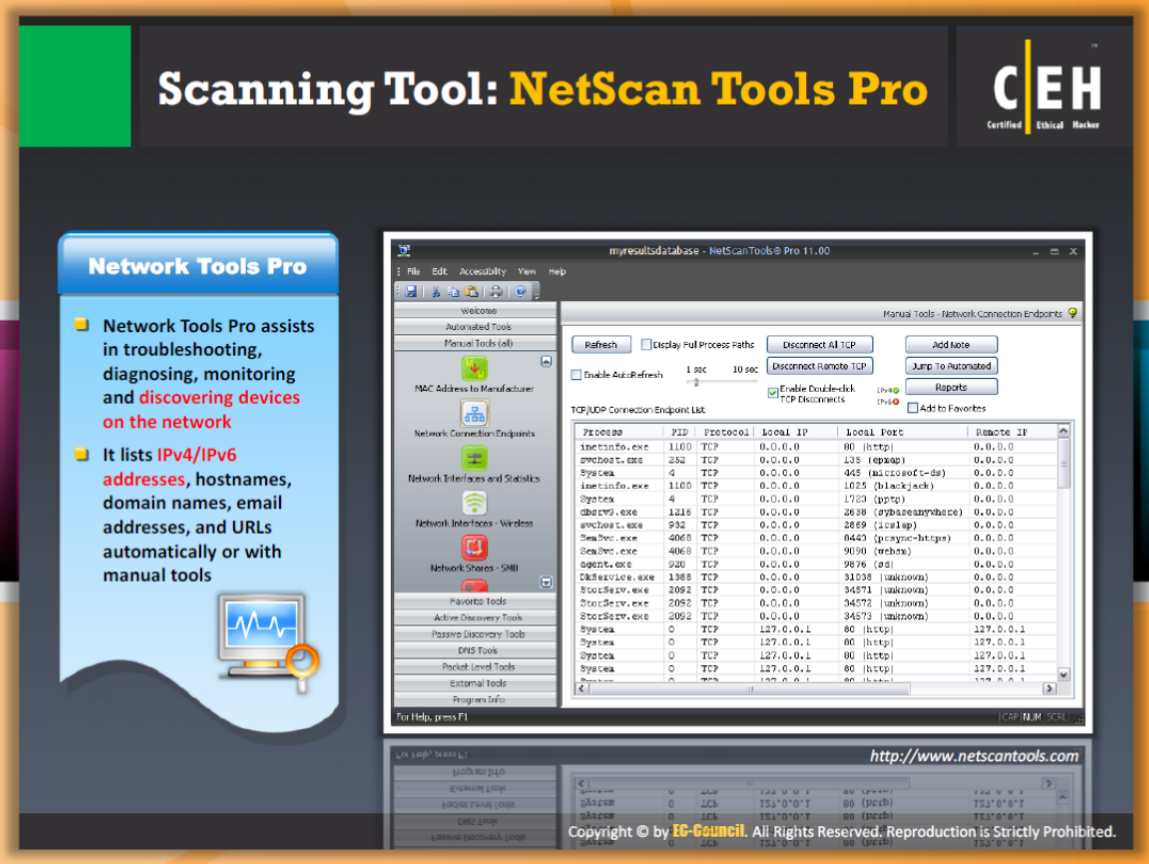


FIGURE 3.41: Zenmap showing ACK Flag Scanning Result



Scanning Tool: NetScan Tools Pro

Source: http://www.netscantools.com

NetScan Tools Pro is an **investigation tool**. It allows you to troubleshoot, monitor, discover, and detect devices on your network. You can **gather information** about the local LAN as well as Internet users, IP addresses, ports, etc. using this tool. You can find vulnerabilities and exposed ports in your system. It is the combination of many **network tools** and **utilities**. The tools are categorized by functions such as active, passive, DNS, and local computer.

Active Discovery and Diagnostic Tools: Used for testing and locating devices that are connected to your network.

Passive Discovery Tools: Monitors the activities of the devices that are connected to your network and also gathers information from third parties.

DNS Tools: Used to detect problems with DNS.

Local Computer and General Information Tools: Provides 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

- Produces the result reports in your web browser clearly

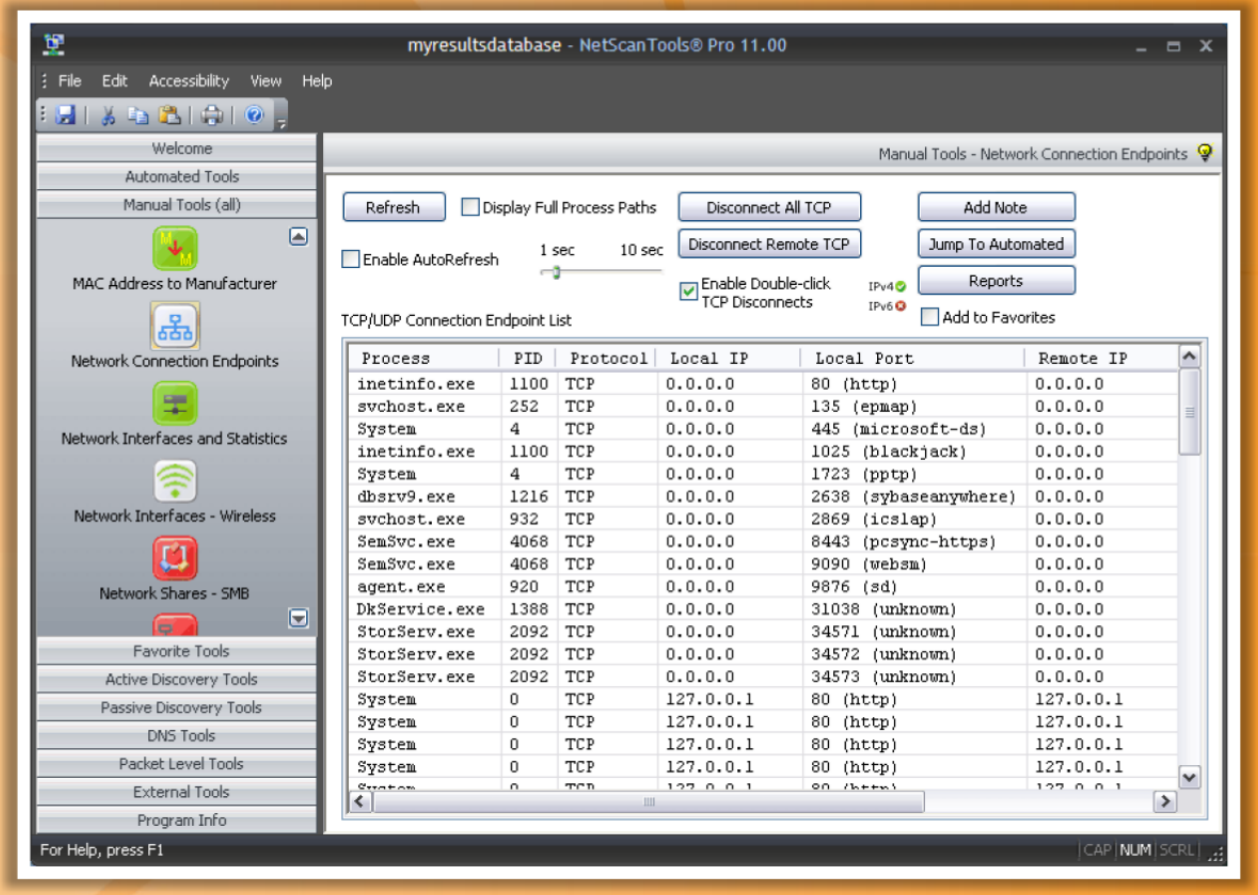
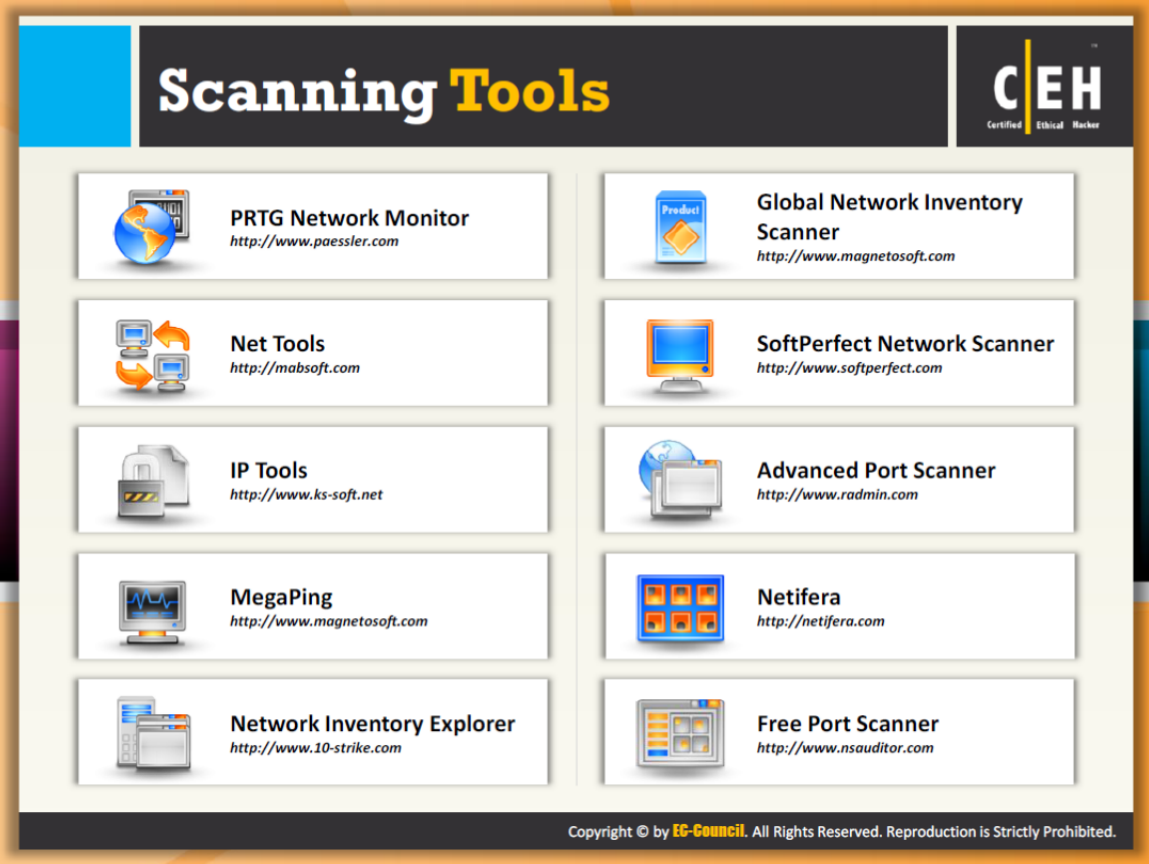


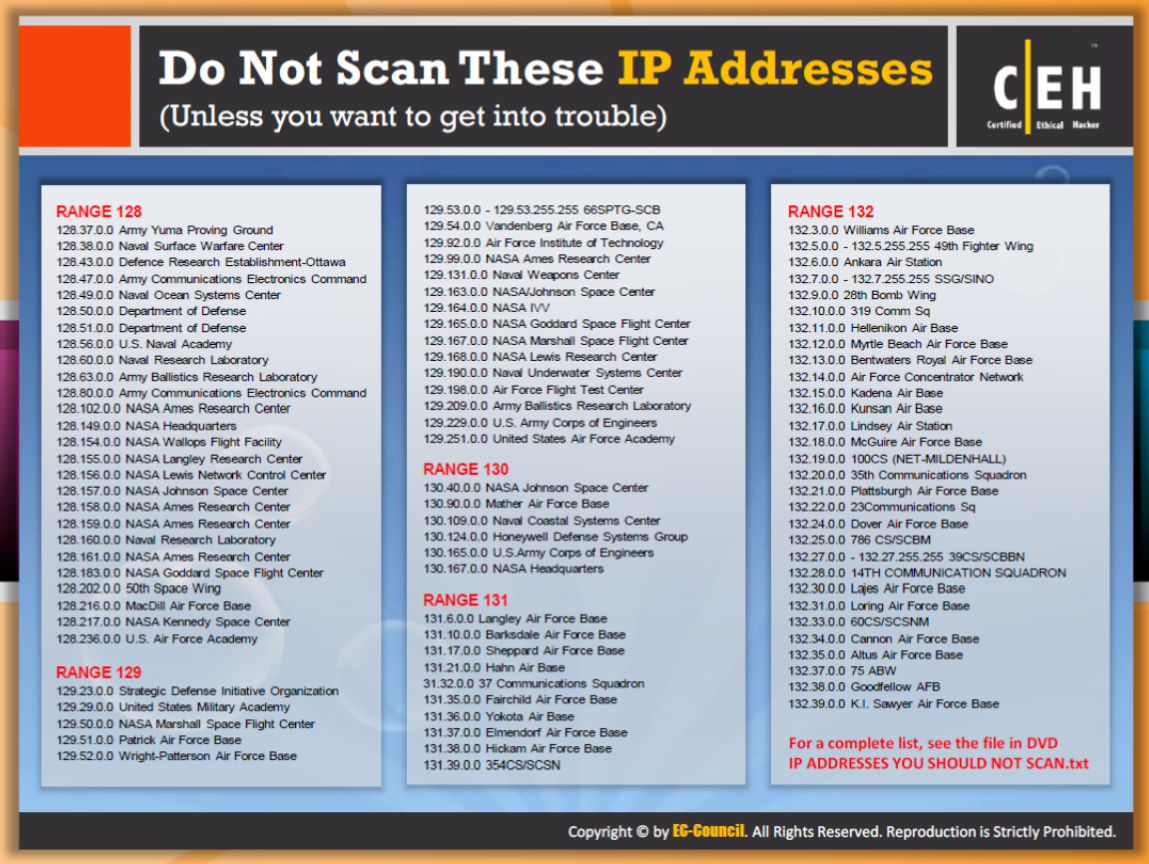
FIGURE 3.42: NetScan Tools Pro Screenshot



Scanning Tools

Scanning tools ping computers, scan for listening **TCP/UDP ports**, and display the type of resources shared on the network (including system and hidden). The attacker may attempt to launch attacks against your network or network resources based on the information gathered with the help of scanning tools. A few of the scanning tools that can detect active ports on the systems are listed as follows:

- PRTG Network Monitor available at http://www.paessler.com
- Net Tools available at http://mabsoft.com
- IP Tools available at http://www.ks-soft.net
- MegaPing available at http://www.magnetosoft.com
- Network Inventory Explorer available at http://www.10-strike.com
- Global Network Inventory Scanner available at http://www.magnetosoft.com
- SoftPerfect Network Scanner available at http://www.softperfect.com
- Advanced Port Scanner available at http://www.radmin.com
- Netifera available at http://netifera.com
- Free Port Scanner available at http://www.nsauditor.com



Do Not Scan These IP Addresses (Unless you want to get into trouble)

The IP addresses listed in the following table are associated with the critical information resource centers of the US. Scanning these IP addresses will be considered an attempt to break the US's information security. Therefore, do not scan these IP addresses unless you want to get into trouble.

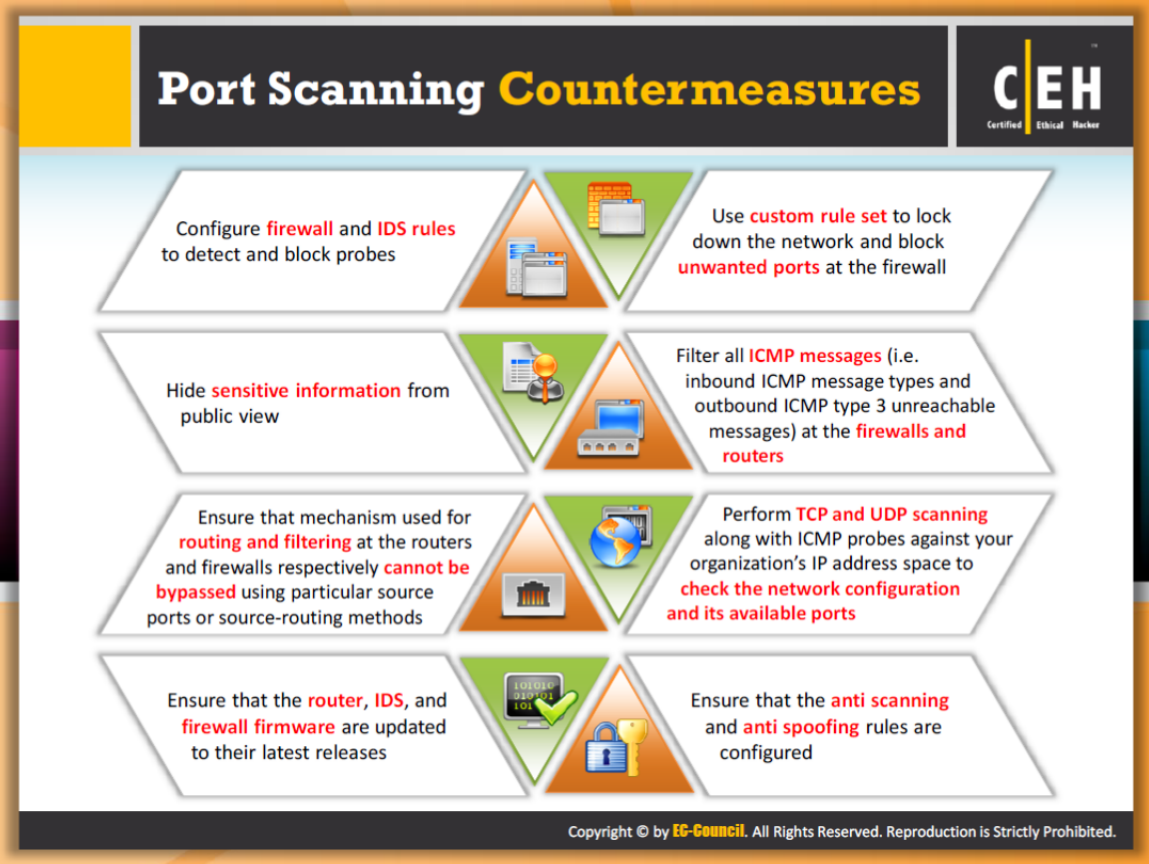
RANGE 6	**129.51.0.0 Patrick Air Force Base**
6.* – Army Information Systems Center	129.52.0.0 Wright-Patterson Air Force Base
RANGE 7	129.53.0.0 - 129.53.255.255 66SPTG-SCB
7.*.*.* Defense Information Systems Agency, VA	129.54.0.0 Vandenberg Air Force Base, CA
RANGE 11	129.92.0.0 Air Force Institute of Technology

11.*.*.* DoD Intel Information Systems, Defense Intelligence Agency, Washington DC	**129.99.0.0 NASA Ames Research Center**
RANGE 21	129.131.0.0 Naval Weapons Center
21. – US Defense Information Systems Agency	129.163.0.0 NASA/Johnson Space Center
RANGE 22	129.164.0.0 NASA IVV
22.* – Defense Information Systems Agency	129.165.0.0 NASA Goddard Space Flight Center
RANGE 24	129.167.0.0 NASA Marshall Space Flight Center
24.198.*.*	129.168.0.0 NASA Lewis Research Center
RANGE 25	129.190.0.0 Naval Underwater Systems Center
25.*.*.* Royal Signals and Radar Establishment, UK	129.198.0.0 Air Force Flight Test Center
RANGE 26	129.209.0.0 Army Ballistics Research Laboratory
26.* – Defense Information Systems Agency	129.229.0.0 U.S. Army Corps of Engineers
RANGE 29	129.251.0.0 United States Air Force Academy
29.* – Defense Information Systems Agency	**RANGE 130**
RANGE 30	130.40.0.0 NASA Johnson Space Center
30.* – Defense Information Systems Agency	130.90.0.0 Mather Air Force Base
RANGE 49	130.109.0.0 Naval Coastal Systems Center
49.* – Joint Tactical Command	130.124.0.0 Honeywell Defense Systems Group
RANGE 50	130.165.0.0 U.S.Army Corps of Engineers
50.* – Joint Tactical Command	130.167.0.0 NASA Headquarters
RANGE 55	**RANGE 131**
55.* – Army National Guard Bureau	131.6.0.0 Langley Air Force Base
RANGE 55	131.10.0.0 Barksdale Air Force Base

55.* – Army National Guard Bureau	**131.17.0.0 Sheppard Air Force Base**
55.* – Army National Guard Bureau	131.17.0.0 Sheppard Air Force Base
RANGE 62	131.21.0.0 Hahn Air Base
62.0.0.1 – 62.30.255.255 Do not scan!	31.32.0.0 37 Communications Squadron
RANGE 64	131.35.0.0 Fairchild Air Force Base
64.70.*.* Do not scan	131.36.0.0 Yokota Air Base
64.224.* Do not Scan	131.37.0.0 Elmendorf Air Force Base
64.225.* Do not scan	131.38.0.0 Hickam Air Force Base
64.226.* Do not scan	131.39.0.0 354CS/SCSN
RANGE 128	**RANGE 132**
128.37.0.0 Army Yuma Proving Ground	132.3.0.0 Williams Air Force Base
128.38.0.0 Naval Surface Warfare Center	132.5.0.0 - 132.5.255.255 49th Fighter Wing
128.43.0.0 Defence Research Establishment-Ottawa	132.6.0.0 Ankara Air Station
128.47.0.0 Army Communications Electronics Command	132.7.0.0 - 132.7.255.255 SSG/SINO
128.49.0.0 Naval Ocean Systems Center	132.9.0.0 28th Bomb Wing
128.50.0.0 Department of Defense	132.10.0.0 319 Comm Sq
128.51.0.0 Department of Defense	132.11.0.0 Hellenikon Air Base
128.56.0.0 U.S. Naval Academy	132.12.0.0 Myrtle Beach Air Force Base
128.60.0.0 Naval Research Laboratory	132.13.0.0 Bentwaters Royal Air Force Base
128.63.0.0 Army Ballistics Research Laboratory	132.14.0.0 Air Force Concentrator Network
128.80.0.0 Army Communications Electronics Command	132.15.0.0 Kadena Air Base
128.102.0.0 NASA Ames Research Center	132.16.0.0 Kunsan Air Base
128.149.0.0 NASA Headquarters	132.17.0.0 Lindsey Air Station
128.154.0.0 NASA Wallops Flight Facility	132.18.0.0 McGuire Air Force Base
128.155.0.0 NASA Langley Research Center	132.19.0.0 100CS (NET-MILDENHALL)

128.156.0.0 NASA Lewis Network Control Center	**132.20.0.0 35th Communications Squadron**
128.157.0.0 NASA Johnson Space Center	132.21.0.0 Plattsburgh Air Force Base
128.157.0.0 NASA Johnson Space Center	132.21.0.0 Plattsburgh Air Force Base
128.158.0.0 NASA Ames Research Center	132.22.0.0 23Communications Sq
128.159.0.0 NASA Ames Research Center	132.24.0.0 Dover Air Force Base
128.160.0.0 Naval Research Laboratory	132.25.0.0 786 CS/SCBM
128.161.0.0 NASA Ames Research Center	132.27.0.0 - 132.27.255.255 39CS/SCBBN
128.183.0.0 NASA Goddard Space Flight Center	132.28.0.0 14TH COMMUNICATION SQUADRON
128.202.0.0 50th Space Wing	132.30.0.0 Lajes Air Force Base
128.216.0.0 MacDill Air Force Base	132.31.0.0 Loring Air Force Base
128.217.0.0 NASA Kennedy Space Center	132.33.0.0 60CS/SCSNM
128.236.0.0 U.S. Air Force Academy	132.34.0.0 Cannon Air Force Base
RANGE 129	132.35.0.0 Altus Air Force Base
129.23.0.0 Strategic Defense Initiative Organization	132.37.0.0 75 ABW
129.29.0.0 United States Military Academy	132.38.0.0 Goodfellow AFB
129.50.0.0 NASA Marshall Space Flight Center	132.39.0.0 K.I. Sawyer Air Force Base

TABLE 3.3: Do Not Scan These IP Addresses Table

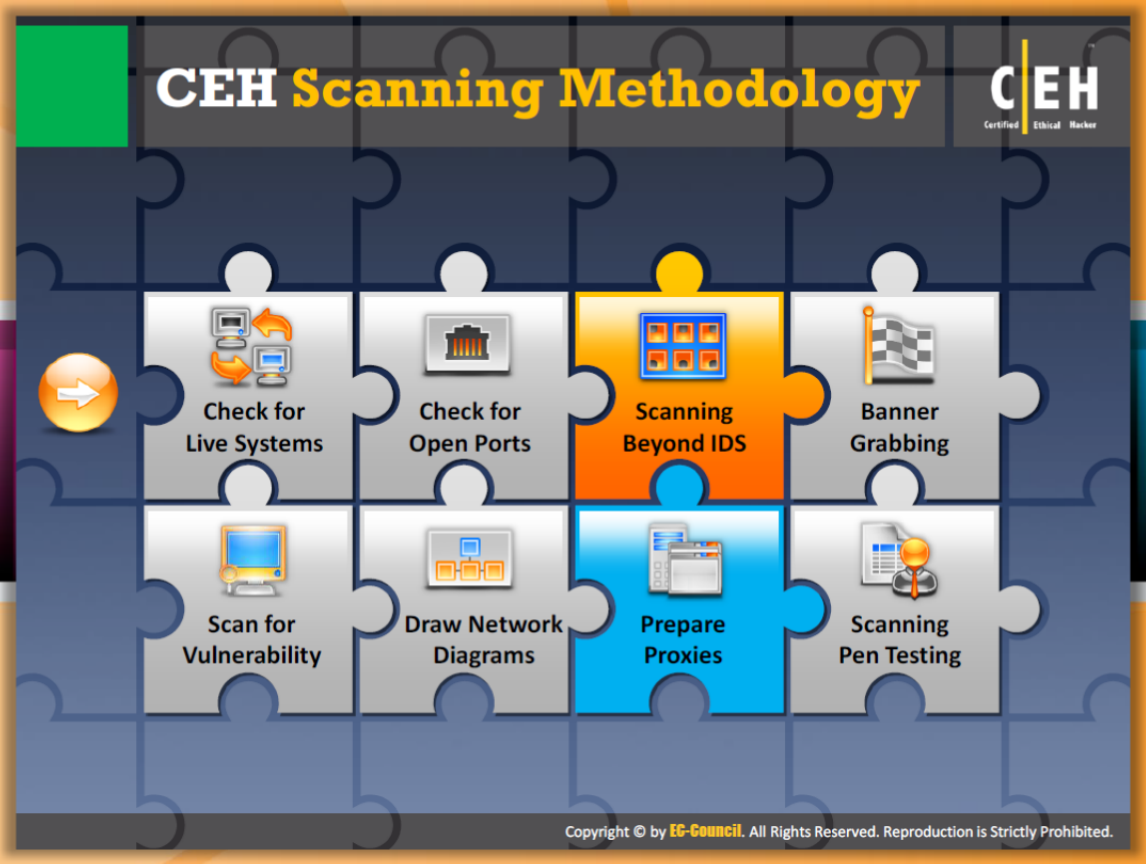


Port Scanning Countermeasures

As discussed previously, port scanning provides a lot of useful information such as IP addresses, host names, open ports, etc. to the attacker. Open ports especially provide an easy means for the attacker to **break into the security**. But there is nothing to worry about, as you can secure your system or network against port scanning by applying the following countermeasures:

- The firewall should be good enough to **detect probes** an attacker sends to scan the network. So the firewall should carry out stateful inspection if it has a specific rule set. Some firewalls do a better job than others in detecting stealth scans. Many firewalls have specific options to detect **SYN scans**, while others completely ignore FIN scans.
- Network intrusion detection systems should detect the OS detection method used by tools such as Nmap, etc. Snort (http://.snort.org) is an intrusion detection and prevention technology that can be of great help, mainly because signatures are frequently available from public authors.
- Only necessary ports should be kept open; the rest of the ports should be filtered as the intruder will try to enter through any open port. This can be accomplished with the custom rule set. Filter inbound ICMP message types and all outbound ICMP type 3 unreachable messages at border routers and firewalls.

- Ensure that routing and filtering mechanisms cannot be bypassed using specific source ports or source-routing techniques.
- Test your own IP address space using TCP and UDP port scans as well as ICMP Probes to determine the network configuration and accessible ports.
- If a commercial firewall is in use, then ensure that the firewall is patched with the latest updates, antispoofing rules have been correctly defined, and fastmode services are not used in Check Point Firewall-1 environments.

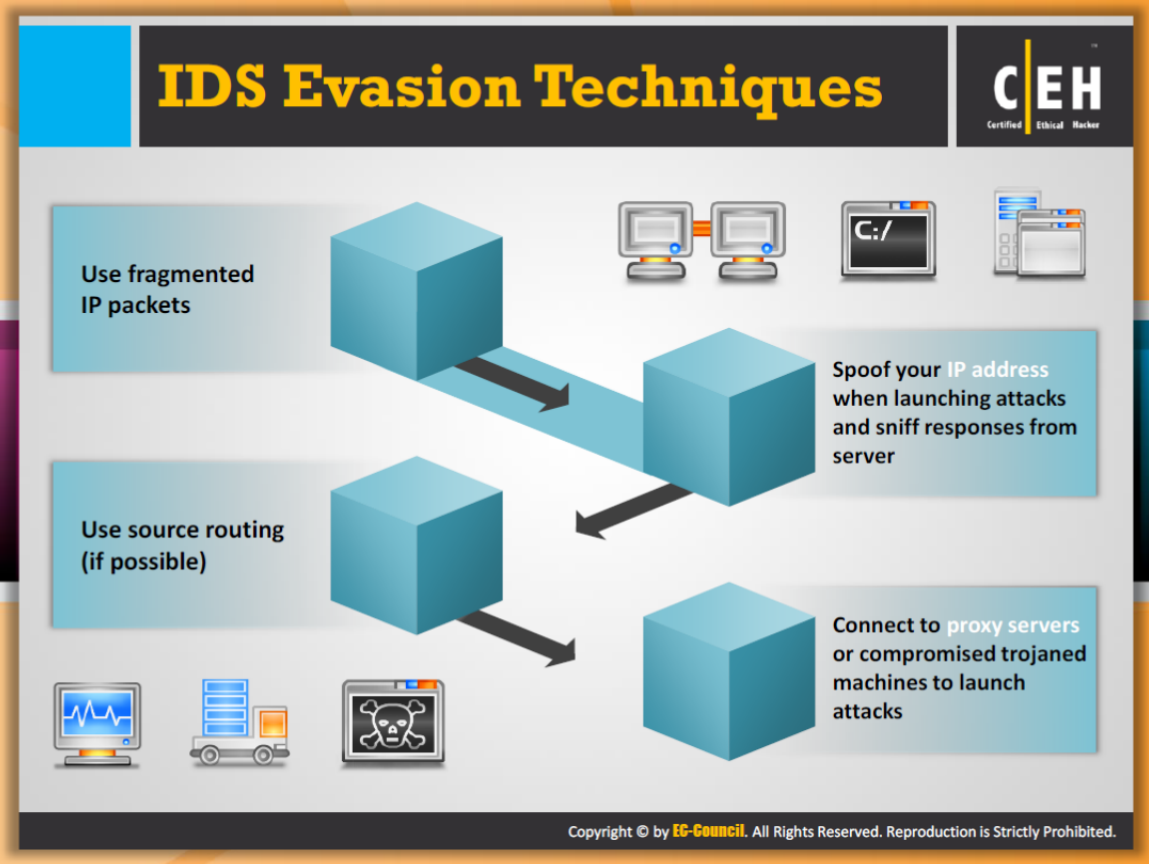


CEH Scanning Methodology

So far we have discussed how to check for live systems and open ports, the two common network vulnerabilities. An IDS is the security mechanism intended to prevent an attacker from entering a secure network. But, even the IDS has some limitations in offering security. Attackers are trying to launch attacks by exploiting limitations of IDS.

Check for Live Systems	**Scan for Vulnerability**
Check for Open Ports	**Draw Network Diagrams**
Scanning Beyond IDS	**Prepare Proxies**
Banner Grabbing	**Scanning Pen Testing**

This section highlights **IDS evasion techniques** and **SYN/FIN scanning**.



IDS Evasion Techniques

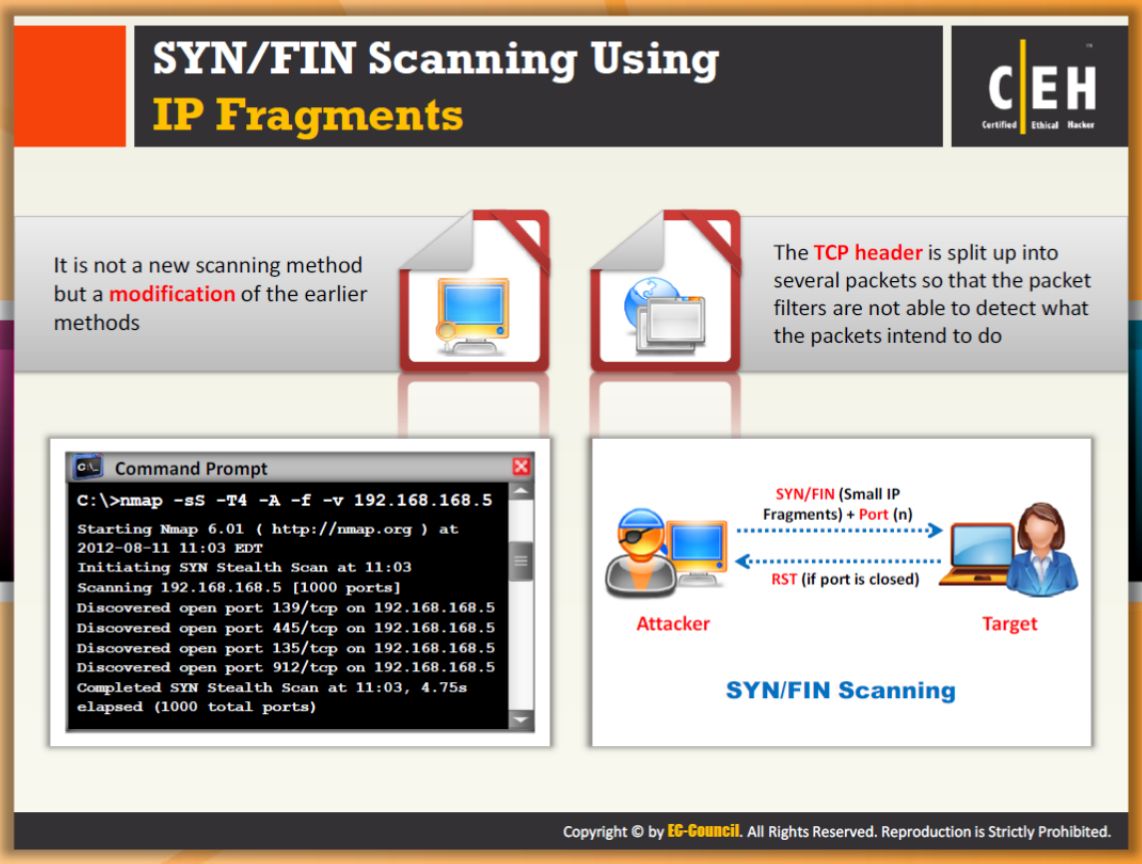
Most of the IDS evasion techniques rely on the use fragmented probe packets that reassemble once they reach the target host. IDS evasion can also occur with the use of **spoofed fake hosts launching** network scanning probes.

Use fragmented IP packets

Attackers use different fragmentation methods to evade the IDS. These attacks are similar to session splicing. With the help of `fragroute`, all the probe packets flowing from your host or network can be fragmented. It can also be done with the help of a port scanner with fragmentation feature such as Nmap. This is accomplished because most IDS sensors fail to process large volumes of **fragmented packets**, as this involves greater CPU consumption and memory at the network sensor level.

Use source routing (if possible)

Source routing is a technique whereby the sender of a packet can specify the route that a packet should take through the network. It is assumed that the source of the packet knows about the layout of the network and can specify the best path for the packet.



SYN/FIN Scanning Using IP Fragments

SYN/FIN scanning using **IP fragments** is a modification of the earlier methods of scanning; the **probe packets** are further fragmented. This method came into existence to avoid the **false positive** from other scans, due to a packet filtering device present on the target machine. You have to split the **TCP header** into **several packets** instead of just sending a probe packet for avoiding the packet filters. Every TCP header should include the source and destination port for the first packet during any transmission: **(8 octet, 64 bit)**, and the initialized flags in the next, which allow the remote host to reassemble the packet upon receipt through an Internet protocol module that recognizes the fragmented data packets with the help of field equivalent values of protocol, source, destination, and identification.

Fragmented Packets

The TCP header, after splitting into small fragments, is transmitted over the network. But, at times you may observe unpredictable results such as fragmentation of the data in the IP header after the reassembly of IP on the server side. Some hosts may not be capable of parsing and reassembling the fragmented packets, and thus may cause crashes, reboots, or even network device monitoring dumps.

Firewalls

Some firewalls may have rule sets that block IP fragmentation queues in the kernel (like the CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel), although this is not widely implemented due to the adverse effect on performance. Since several intrusions detection systems employ signature-based methods to indicate scanning attempts based on IP and/or the TCP headers, fragmentation is often able to evade this type of packet filtering and detection. There is a probability of network problems on the target network.

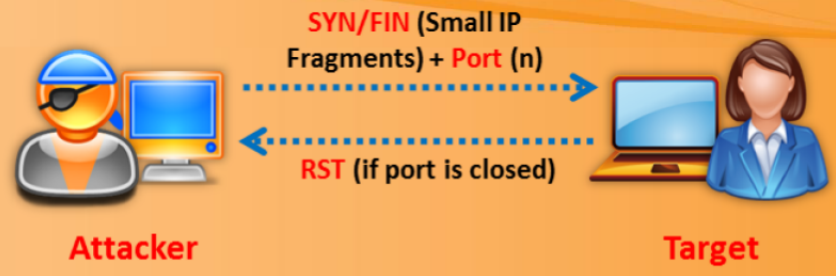


FIGURE 3.43: SYN/FIN Scanning

Nmap command prompts for SYN/FIN scan.

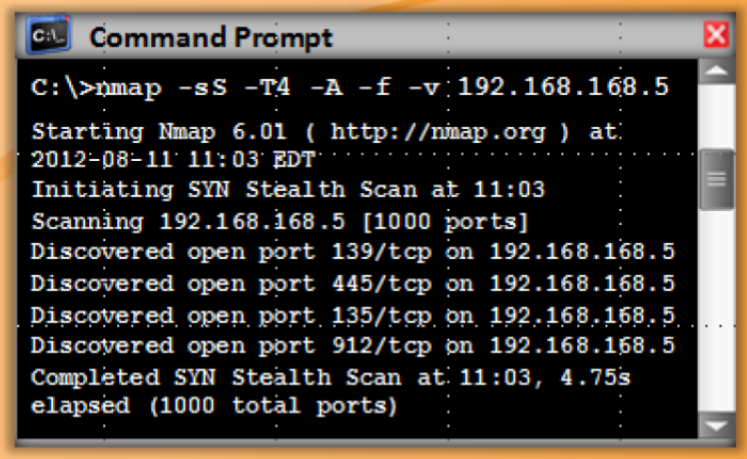
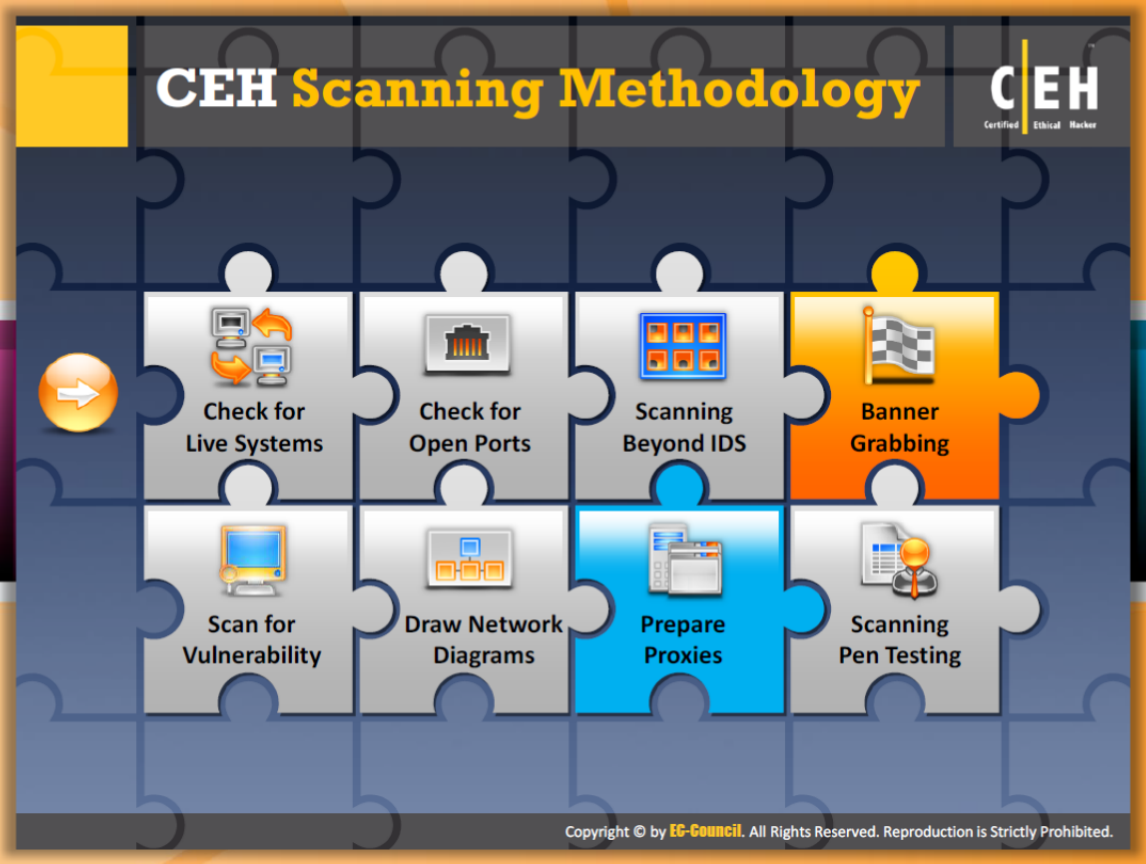


FIGURE 3.44: Nmap showing SYN/FIN Scanning Result

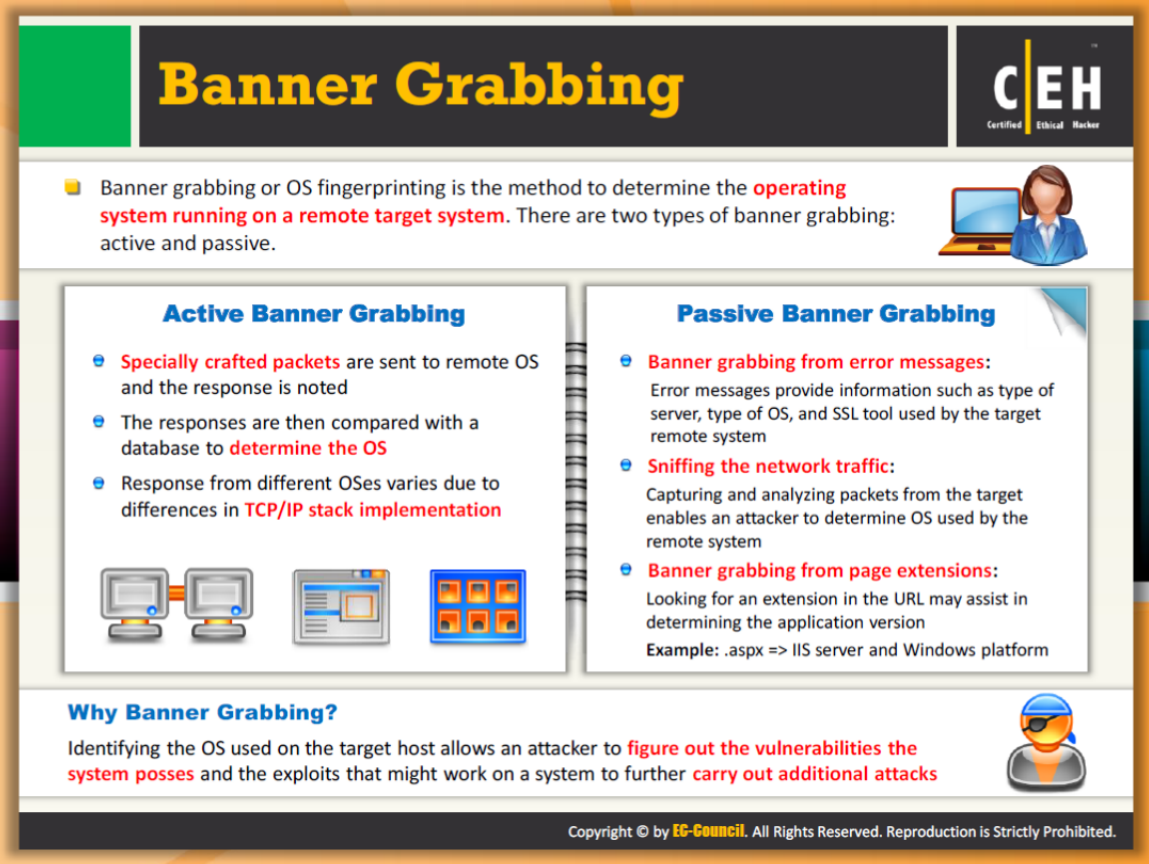


CEH Scanning Methodology

So far we have discussed how to check for live systems, open ports, and scan beyond IDS. All of these are the doorways for an attacker to break into a network. Another important tool of an attacker is banner grabbing, which we will discuss next.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section highlights banner grabbing, the need to perform banner grabbing, various ways of banner grabbing, and the tools that help in banner grabbing.



Banner Grabbing

Banner grabbing or OS fingerprinting is a method to determine the **operating system running on a remote target system**. Banner grabbing is important for hacking as it provides you with a greater probability of success in hacking. This is because most of the vulnerabilities are OS specific. Therefore, if you know the OS running on the target system, you can hack the system by exploiting the vulnerabilities specific to that operating system.

Banner grabbing can be carried out in two ways: either by spotting the banner while trying to connect to a service such as FTP or downloading the **binary file/bin/ls** to check the architecture with which it was built.

Banner grabbing is performed using the fingerprinting technique. A more advanced fingerprinting technique depends on stack querying, which transfers the packets to the network host and evaluates packets based on the reply. The first stack querying method was designed considering the TCP mode of communication, in which the response of the connection requests is evaluated. The next method was known as ISN (Initial Sequence Number) analysis. This identifies the differences in the random number generators found in the TCP stack. A new method, using the ICMP protocol, is known as **ICMP response analysis**. It consists of sending the ICMP messages to the remote host and evaluating the reply. The latest ICMP messaging is

known as temporal response analysis. Like others, this method uses the TCP protocol. Temporal response analysis looks at the **retransmission timeout (RTO)** responses from a remote host.

There are two types of banner grabbing techniques available; one is active and the other is passive.

Active Banner Grabbing

Active banner grabbing is based on the principle that an operating system's IP stack has a unique way of responding to specially crafted TCP packets. This arises because of different interpretations that vendors apply while implementing the TCP/IP stack on the particular OS. In active banner grabbing, a variety of malformed packets are sent to the remote host, and the responses are compared to a database.

For instance, in Nmap, the **OS fingerprint** or **banner grabbing** is done through eight tests. The eight tests are named T1, T2, T3, T4, T5, T6, T7, and PU (port unreachable). Each of these tests is illustrated as follows, as described in www.packetwatch.net:

T1: In this test, a TCP packet with the SYN and ECN-Echo flags enabled is sent to an open TCP port.

T2: It involves sending a TCP packet with no flags enabled to an open TCP port. This type of packet is known as a NULL packet.

T3: It involves sending a TCP packet with the URG, PSH, SYN, and FIN flags enabled to an open TCP port.

T4: It involves sending a TCP packet with the ACK flag enabled to an open TCP port.

T5: It involves sending a TCP packet with the SYN flag enabled to a closed TCP port.

T6: It involves sending a TCP packet with the ACK flag enabled to a closed TCP port.

T7: It involves sending a TCP packet with the URG, PSH, and FIN flags enabled to a closed TCP port.

PU (Port Unreachable): It involves sending a UDP packet to a closed UDP port. The objective is to extract an "**ICMP port unreachable**" message from the target machine.

The last test that Nmap performs is named TSeq for TCP Sequencability 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. The test is performed by sending six TCP packets with the SYN flag enabled to an open TCP port.

The objective is to find patterns in the initial sequence of numbers that the TCP implementations choose while responding to a connection request. These can be categorized into many 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 where the ISN is incremented by a fixed amount for each time period.

Source: www.insecure.org, "Most operating systems increment a system-wide IPID value for each packet they send. Others, such as **OpenBSD**, use a random IPID and some systems (like Linux) use an IPID of 0 in many cases where the 'Don't Fragment' bit is not set. Windows does not put the IPID in network byte order, so it increments by **256** for each packet. Another number that can be sequenced for OS detection purposes is the TCP timestamp option values. Some systems do not support the feature; others increment the value at frequencies of 2HZ, 100HZ, or 1000HZ and still others return 0."

Passive Banner Grabbing

Source: http://honeynet.org

Like active banner grabbing, passive banner grabbing is also based 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.

The four areas that are typically noted to determine the operating system are:

- TTL - What the operating system sets the Time To Live on the outbound packet
- Window Size - hat the operating system sets the Window size
- DF - Does the operating system set the Don't Fragment bit
- OS - Does the operating system set the Type of Service, and if so, at what

Passive fingerprinting has to be neither fully accurate nor be limited to these four signatures. However, by looking at several signatures and combining information, accuracy can be improved. The following is the analysis of a sniffed packet dissected by Lance Spitzner in his paper on passive fingerprinting (http://www.honeynet.org/papers/finger/).

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

Based on 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

Database Signatures

This information is then compared to a database of signatures. Considering the TTL used by the remote host, it is determined from the sniffer trace that the TTL is set at 45. This indicates that it went through **19 hops** to get to the target, so the original TTL must have been set at 64.

Based on this TTL, it appears that the packet was sent from a **Linux or FreeBSD** box (however, more system signatures need to be added to the database). This TTL is confirmed by doing a traceroute to the remote host. If the trace needs to be done stealthily, the traceroute time-to-live (**default 30 hops**) can be set to be one or two hops less than the remote host (-m option). Setting traceroute in this manner reveals the path information (including the upstream provider) without actually touching the remote host.

Window Sizes

The next step is to **compare window sizes**. The window size is another effective tool that determines specifically what window size is used and how often the size is changed. In the previous signature, it is set at **0x7D78**, a default window size is 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 are constantly changing. The window size is more accurate if measured after the initial three-way handshake (due to TCP slow start).

Session Based

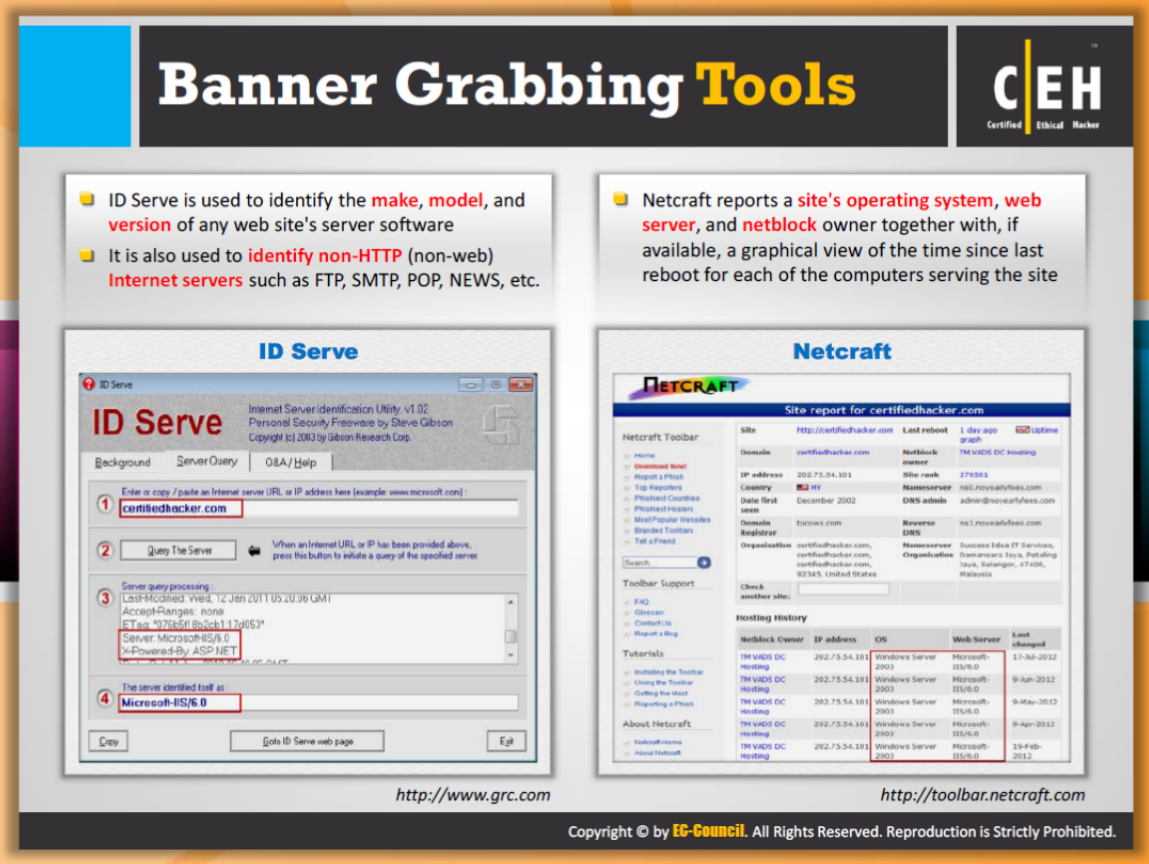
Most systems use the DF bit set, so this is of limited value. However, this does make it easier to identify the few systems that do not use the **DF flag** (such as **SCO or OpenBSD**). TOS is also of limited value, since it seems to be more session-based than **operating-system-based**. In other words, it is not so much the operating system that determines the TOS, but the protocol used. Therefore, based on this information, specifically TTL and window size, one can compare the results to the database of signatures and, with a degree of confidence, determine the OS (in this case, Linux kernel 2.2.x).

Just as with active fingerprinting, passive fingerprinting has some limitations. First, applications that build their own packets (such as Nmap, hunt, nemesis, etc.) will not use the same signatures as the operating system. Second, it is relatively simple for a remote host to adjust the TTL, window size, DF, or TOS setting on packets.

Passive fingerprinting can be used for several other purposes. Crackers can use “stealthy” fingerprinting. For example, to determine the operating system of a potential target, such as a web server, one need only 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. Also, passive fingerprinting may be used to identify remote proxy firewalls. Since proxy firewalls rebuild connections for clients, it may be possible to ID proxy firewalls based on the signatures that have been discussed. Organizations can use passive fingerprinting to identify rogue systems on their network. These would be systems that are not authorized on the network.

Why Banner Grabbing?

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



Banner Grabbing Tools

Banner grabbing can be done even with the help of tools. Many tools are available in the market. These tools make banner grabbing an easy task. The following are examples of banner grabbing tools:

ID Serve

Source: http://www.grc.com

ID Serve is used to identify the make, model, and version of any website's server software; it is also used to identify non-HTTP (non-web) Internet servers such as FTP, SMTP, POP, NEWS, etc.

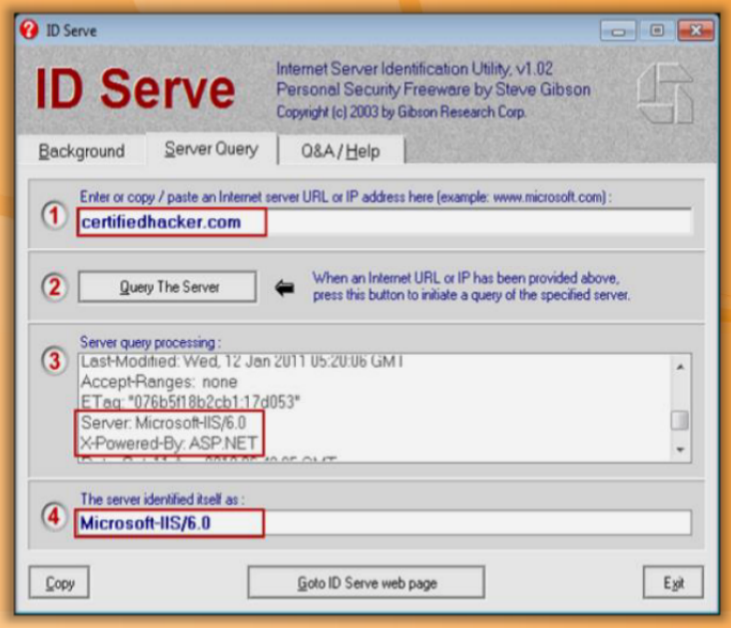


FIGURE 3.45: ID Serve Screenshot

Netcraft

Source: http://toolbar.netcraft.com

Netcraft reports a site's operating system, web server, and netblock owner together with, if available, a graphical view of the time since last reboot for each of the computers serving the site.

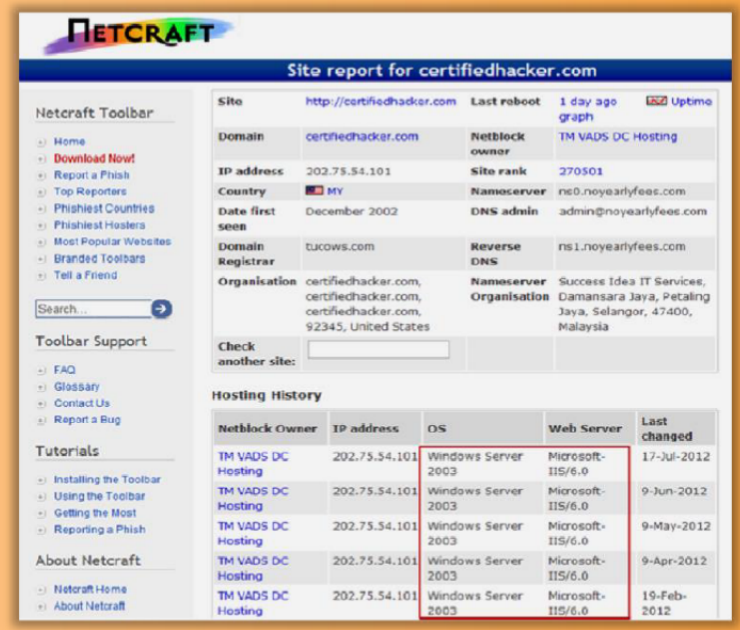
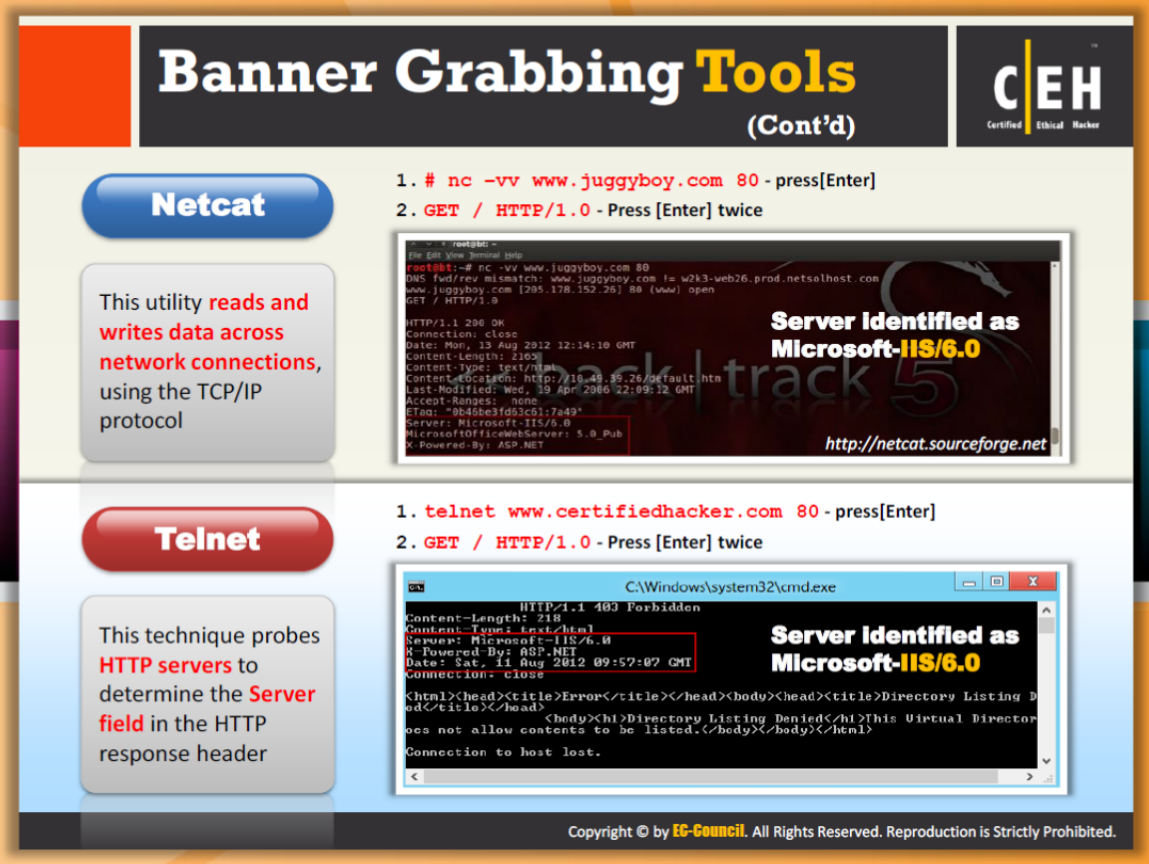


FIGURE 3.46: Netcraft Screenshot



Banner Grabbing Tools (Cont'd)

Netcat

Source: http://netcat.sourceforge.net

Netcat is a networking utility that **reads** and **writes data** across network connections, with the help of the TCP/IP protocol. It is designed to be a reliable "**back-end**" tool that can be used directly or easily driven by other programs and scripts. It provides access to the following key features:

- Outbound and inbound connections, TCP or UDP, to or from any ports.
- Featured tunneling mode, which also allows special tunneling such as UDP to TCP, with the possibility of specifying all network parameters (source port/interface, listening port/interface), and the remote host allowed to connect to the tunnel.
- Built-in port-scanning capabilities, with randomizer.
- Advanced usage options, such as buffered send-mode (one line every N seconds) and hexdump (to stderr or to a specified file) of transmitted and received data.

Optional RFC854 telnet codes parser and responder. You can use the Netcat tool for grabbing the banner of a website by following this process. Here, the banner grabbing is done on the www.Juggyboy.com web server for gathering the server fields such as server type, version, etc.

- # nc –vv www.juggyboy.com 80 - press[Enter]
- GET / HTTP/1.0 - Press [Enter] twice

From the screenshot, you can observe the area highlighted in red color is a server version (Microsoft-IIS/6.0).

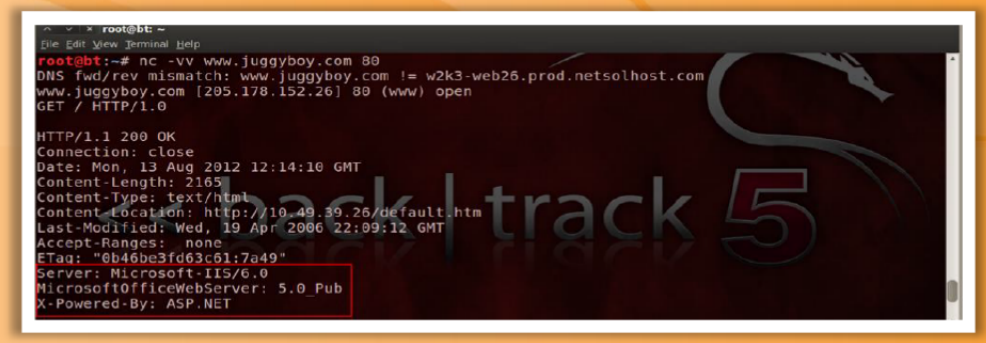


FIGURE 3.47: Netcat showing Banner Grabbing Result

Telnet

This technique probes **HTTP servers** to determine the Server field in the HTTP response header.

For instance, if you want to enumerate a host running http (tcp 80), then you have to follow this procedure:

- First, open the command prompt window.

  Go to **Start** → **Run**, type **cmd**, and press **Enter** or click **OK**.
- In the command prompt window, request telnet to connect to a host on a specific port: C:\telnet www.certifiedhacker.com 80 and press **Enter**.
- Next, you will get a blank screen where you have to type **GET / HTTP/1.0** and press **Enter** twice.
- In the final step, the http server responses with the server version, say Microsoft-IIS/6.0. From the screenshot you can see the area highlighted in red color is the server version details.

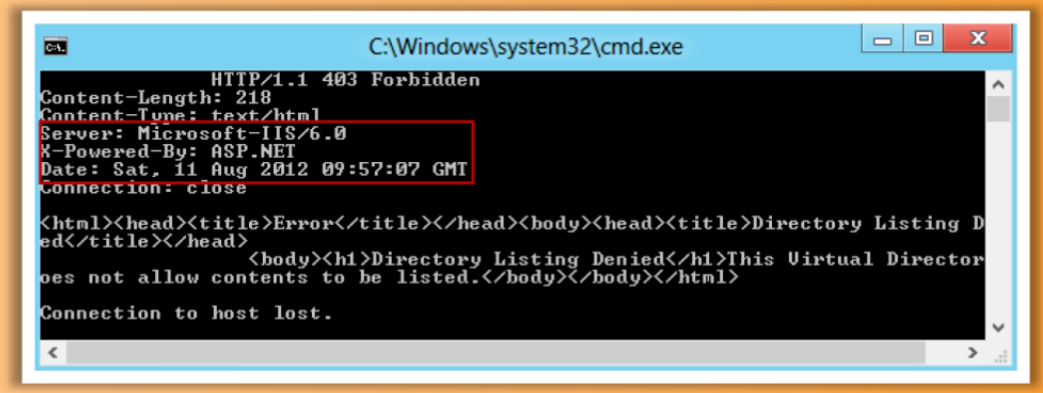
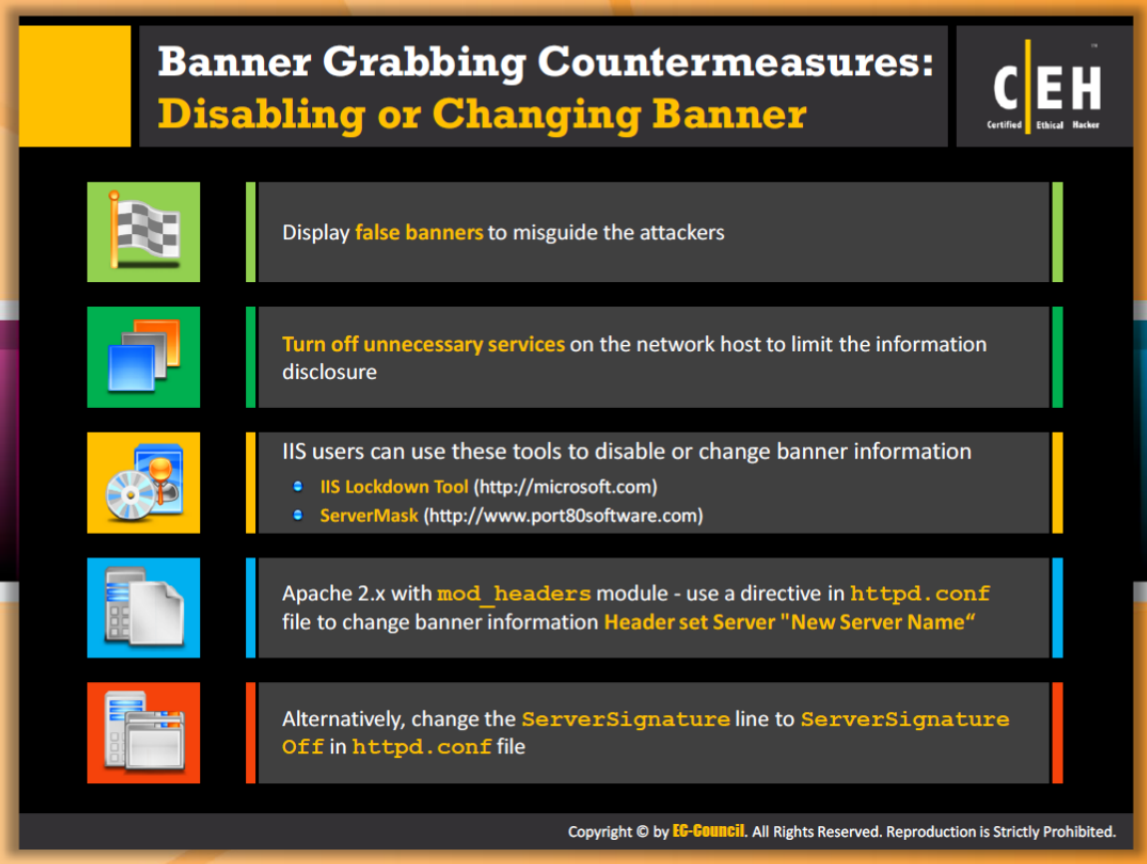


FIGURE 3.48: Command Prompt showing http server responses with the server version



Banner Grabbing Countermeasures: Disabling or Changing Banners

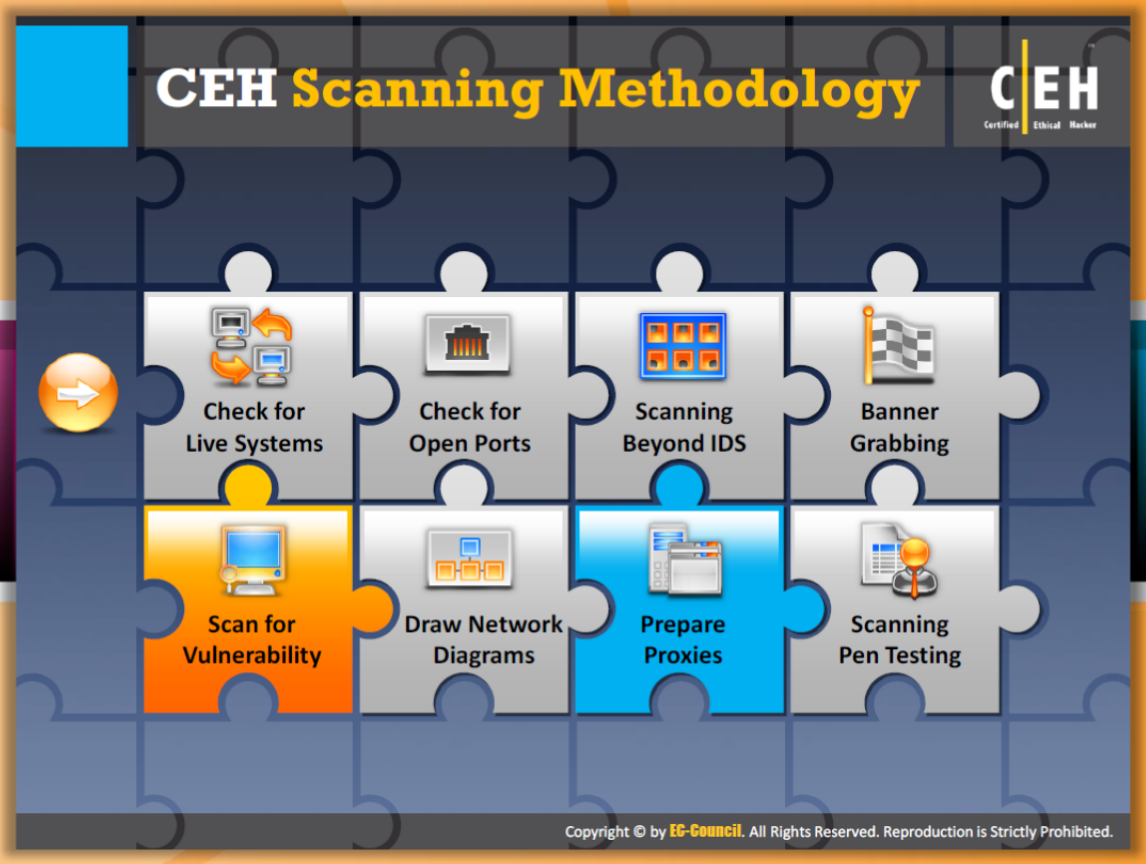
Attackers use banner grabbing techniques and find out **sensitive information** such as device types, operating systems, application version, etc. used by the victim. With the help of the gathered information, the attacker exploits the vulnerabilities that are not updated with the **security patches** and launches the attacks. So, to protect your system against banner grabbing attacks, a few countermeasures can be adopted and they are listed as follows:

Disabling or Changing Banner

- Display false banners to misguide attackers
- Turn off unnecessary services on the network host to limit information disclosure
- IIS users can use these tools to disable or change banner information:
  - IIS Lockdown Tool (http://microsoft.com)
  - ServerMask (http://www.port80software.com)
- 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 the `httpd.conf` file

Hiding File Extensions from Web Pages

File extensions provide information about the **underlying server technology**; attackers can use this information to search vulnerabilities and launch attacks. Hiding file extensions is a good practice to mask **technology-generating** dynamic pages. 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. Doing without file extensions altogether is an even better idea.

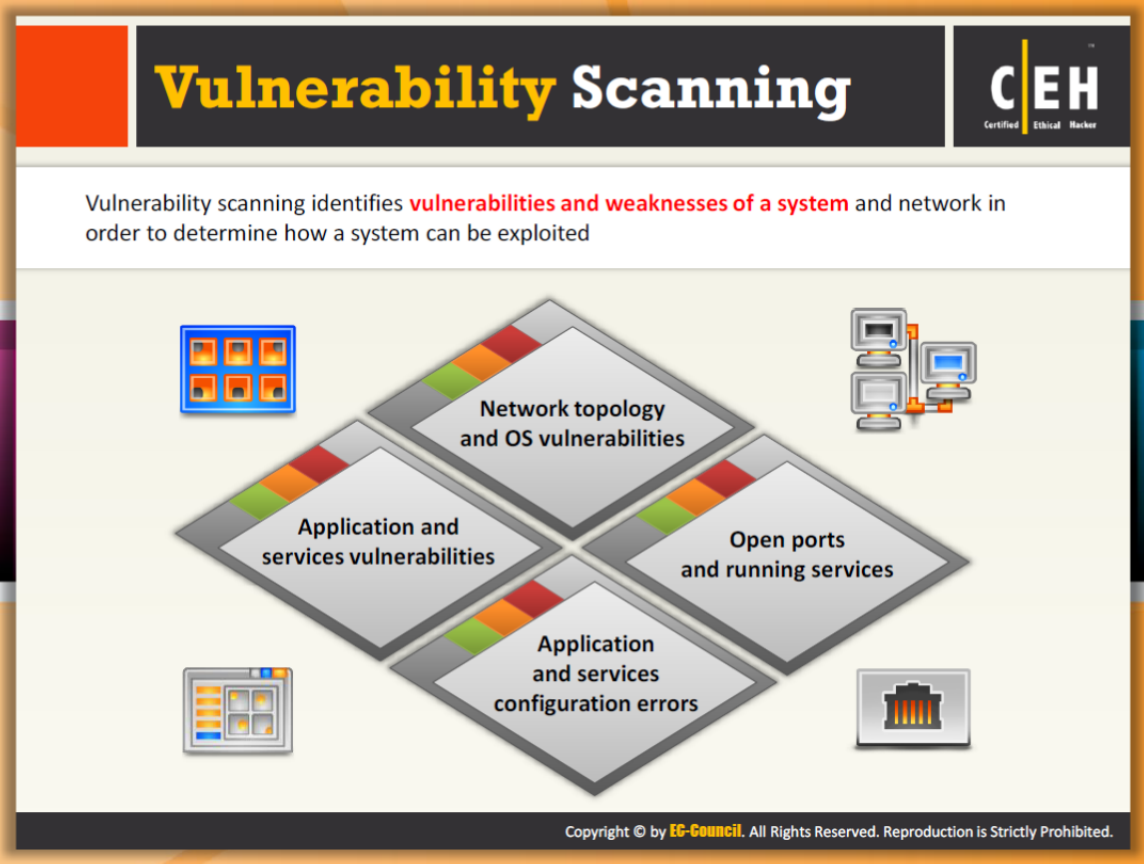


CEH Scanning Methodology

So far, we have discussed how to check for live systems, open ports, scan beyond IDS, and the use of banner grabbing. All these concepts help an attacker or security administrator to find the loopholes that may allow an attacker into their network. Now we will discuss vulnerability scanning, a more detailed tests to determine vulnerabilities in a network, and its resources.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section describes vulnerability scanning and various vulnerability scanning tools.



Vulnerability Scanning

Vulnerability scanning identifies **vulnerabilities** and **weaknesses** of a system and network in order to determine how a system can be exploited. Similar to other security tests such as **open port scanning** and **sniffing**, the vulnerability test also assists you in securing your network by determining the **loopholes** or vulnerabilities in your current security mechanism. This same concept can also be used by attackers in order to find the weak points of the target network. Once they find any weak points, they can exploit them and get in to the target network. Ethical hackers can use this concept to determine the security weaknesses of their target business and fix them before the bad guys find and exploit them.

Vulnerability scanning can find the vulnerabilities in:

- Network topology and OS vulnerabilities
- Open ports and running services
- Application and services configuration errors
- Application and services vulnerabilities

Vulnerability Scanning Tool: Nessus

Source: http://www.tenable.com

Nessus is a **vulnerability scanner—a program** that searches for bugs in software. This tool allows a person to discover a specific way to violate the security of a software product. The vulnerability, in various levels of detail, is then disclosed to the user of the tool. The various steps this tool follows are:

- Data gathering
- Host identification
- Port scan
- Plug-in selection
- Reporting of data

To obtain more accurate and detailed information from **Windows-based hosts** in a Windows domain, the user can create a domain group and account that have remote registry access privileges. After completing this task, he or she gets access not only to the registry key settings but also to the Service Pack patch levels, Internet Explorer vulnerabilities, and services running on the host.

It is a client-server application. The Nessus server runs on a UNIX system, keeps track of all the different vulnerability tests, and performs the actual scan. It has its own user database and secures authentication methods, so that remote users using the Nessus client can log in, configure a vulnerability scan, and send it on its way. Nessus includes **NASL (Nessus Attack Scripting Language)**, a language designed to write security tests.

The various features of Nessus are:

- Each security test is written as a **separate plug-in**. This way, the user can easily add tests without having to read the code of the Nessus engine.
- It performs smart service recognition. It assumes that the target hosts will respect the IANA assigned port numbers.
- The Nessus Security Scanner is made up of two parts: A server, which performs the attack, and a client, which is the front end. The server and the client can be run on different systems. That is, the user can audit his whole network from his personal computer, whereas the server performs its attacks from the main frame, which may be located in a different area.

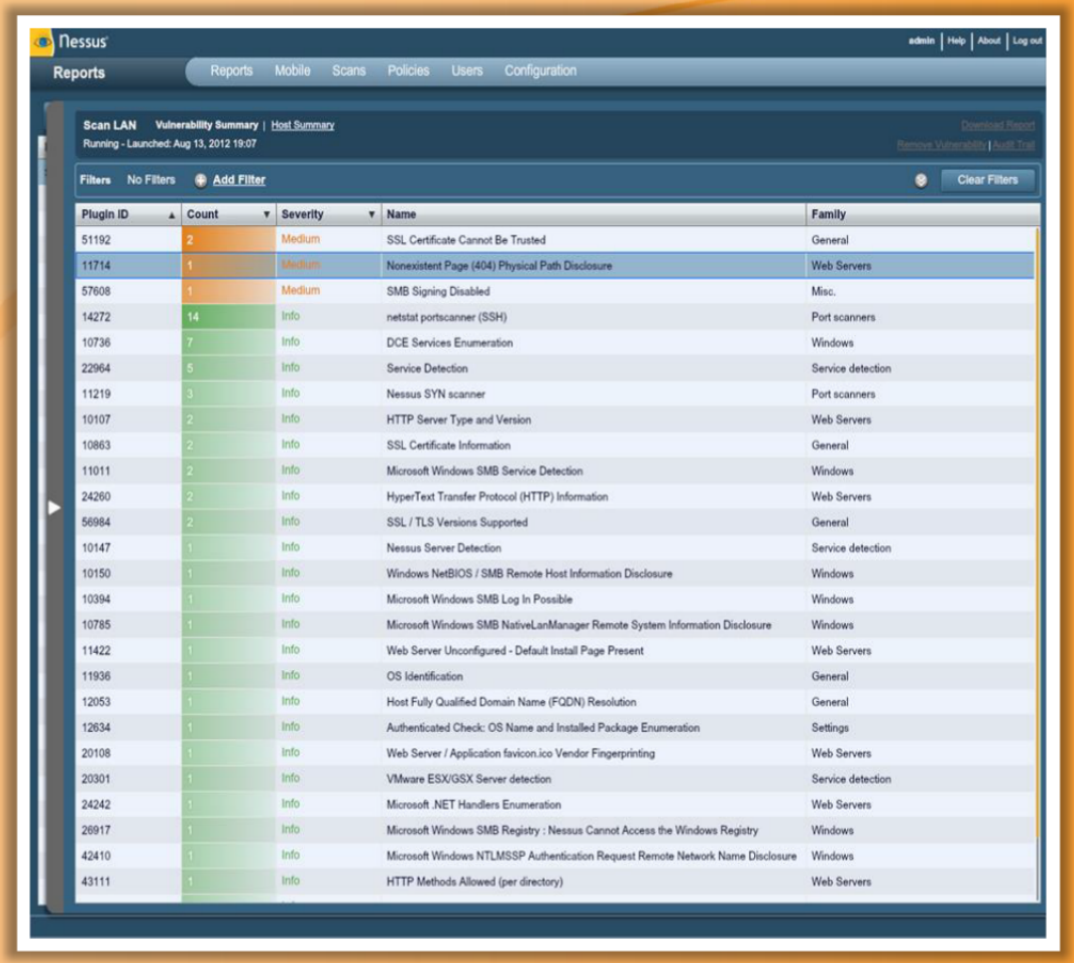
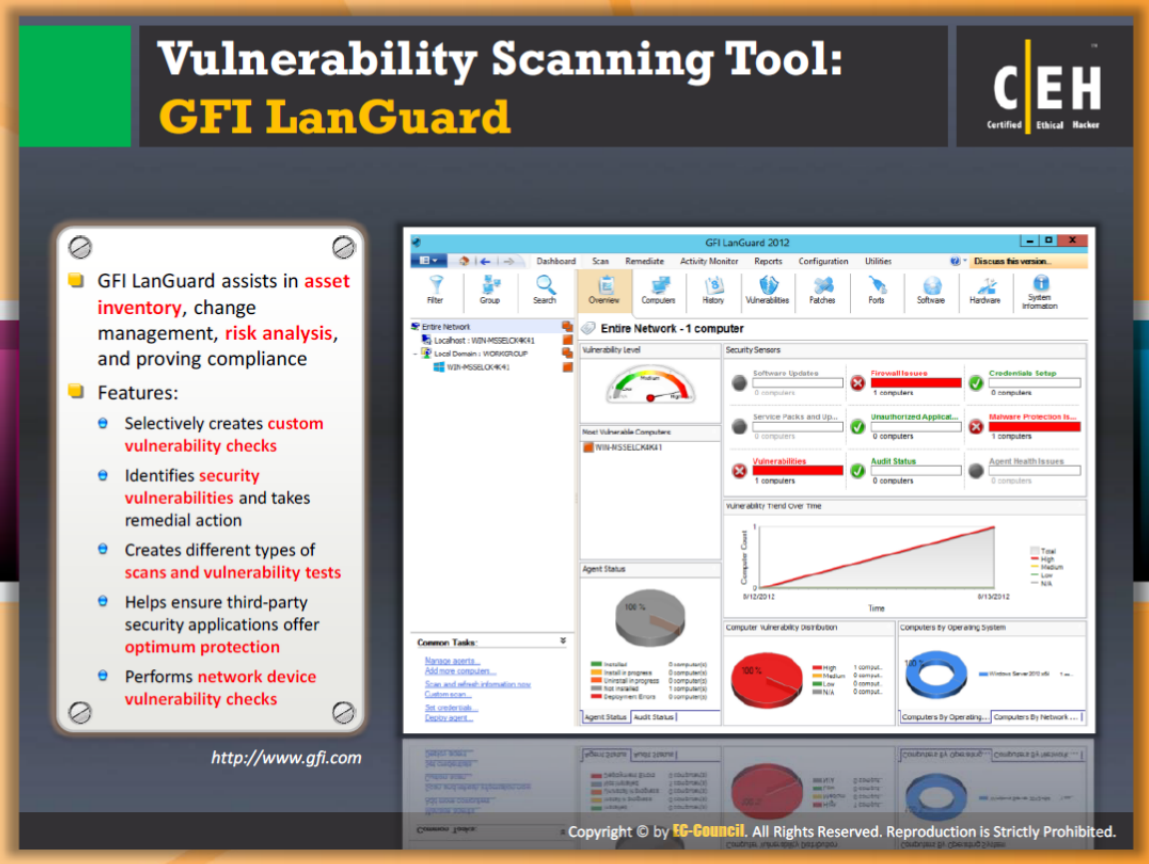


FIGURE 3.49: Nessus Screenshot



Vulnerability Scanning Tool: GFI LanGuard

Source: http://www.gfi.com

GFI LanGuard acts as a **virtual security consultant**. It offers patch management, vulnerability assessment, and network auditing services. It also assists you in asset inventory, change management, risk analysis, and proving compliance.

Features:

- Selectively creates custom vulnerability checks
- Identifies security vulnerabilities and takes remedial action
- Creates different types of scans and vulnerability tests
- Helps ensure third party security applications offer optimum protection
- Performs network device vulnerability checks

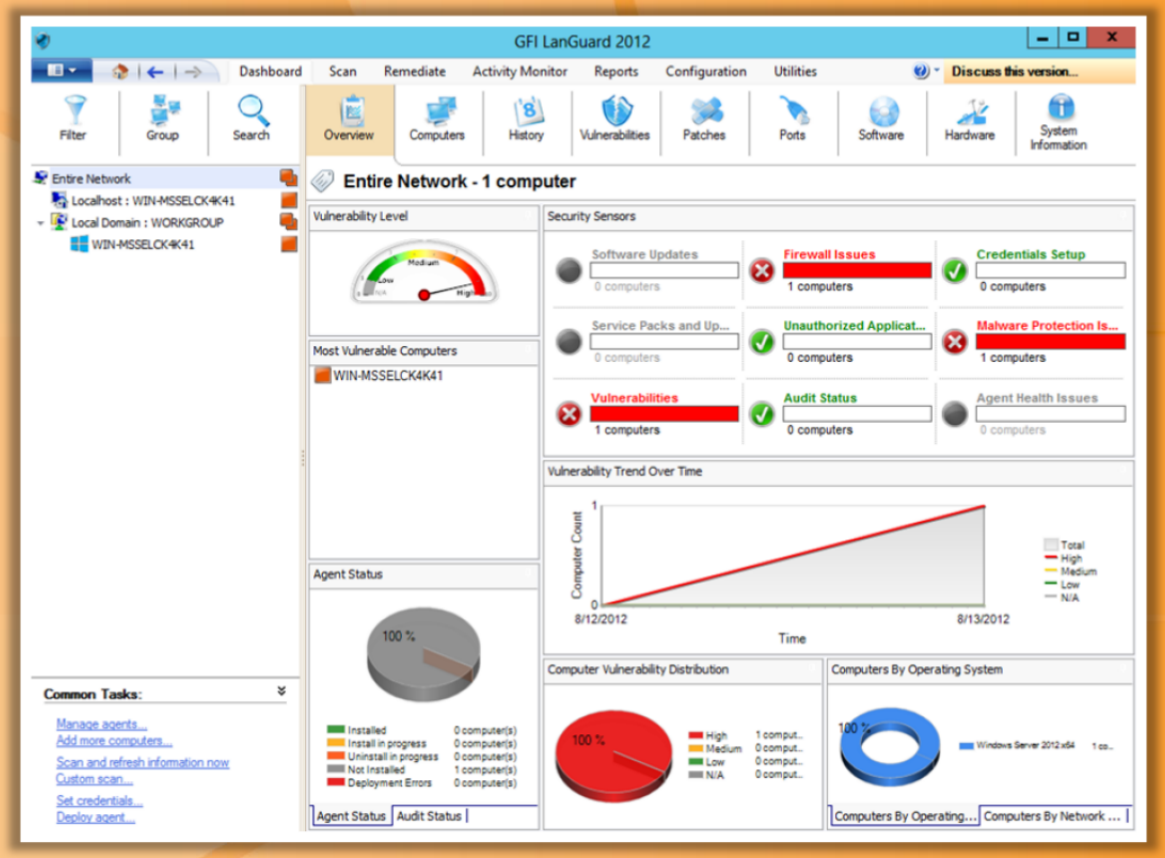
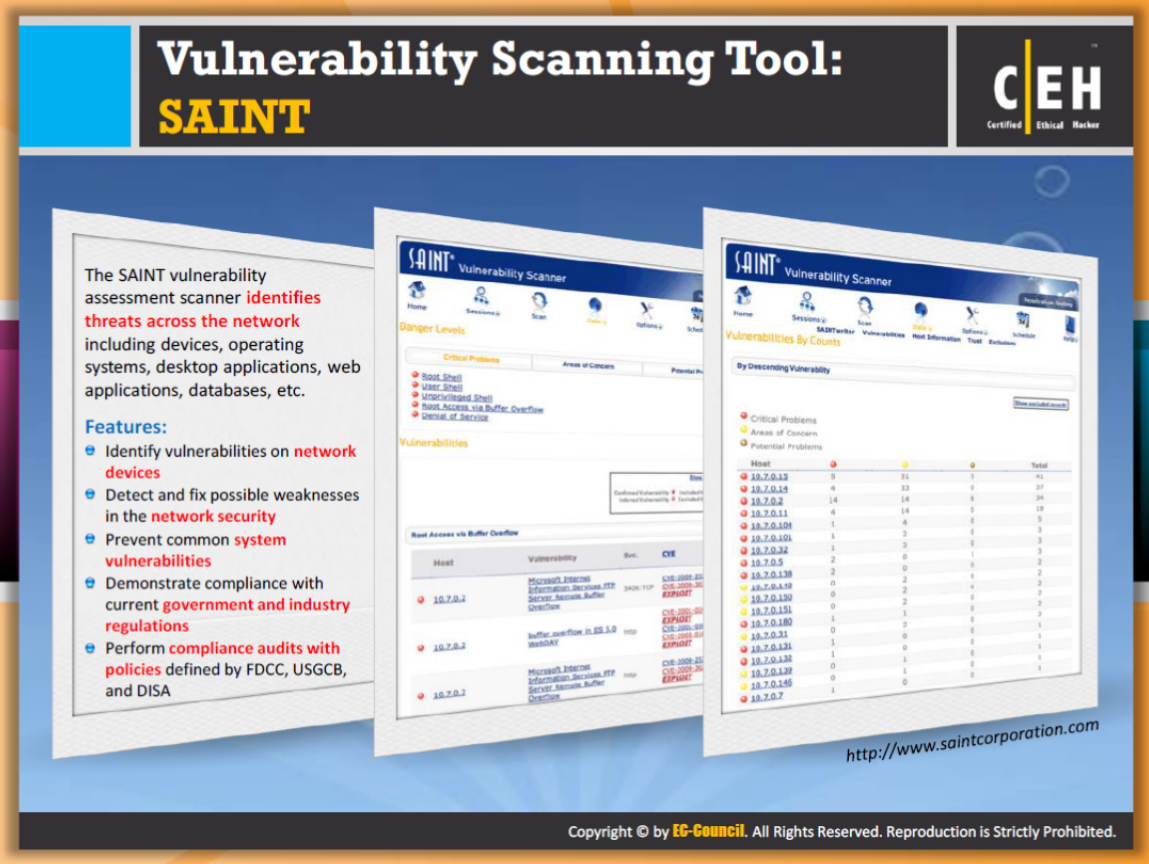


FIGURE 3.50: GFI LanGuard Screenshot



Vulnerability Scanning Tool: SAINT

Source: http://www.saintcorporation.com

SAINT is an **integrated network tools** for **security administrators**. Using this tool, you can find security vulnerabilities across the network including devices, operating systems, desktop applications, web applications, databases, etc. in a **non-intrusive manner**. It also enables you to gather information such as operating system types and open ports, etc. It allows you to scan and exploit targets with an IPv4, IPv6, and/or URL address.

Features:

- Detects and fixes possible weaknesses in your network's security
- Anticipates and prevents common system vulnerabilities
- Demonstrates compliance with current government and industry regulations such as PCI DSS, NERC, FISMA, SOX, GLBA, HIPAA, and COPPA
- Performs compliance audits with policies defined by FDCC, USGCB, and DISA

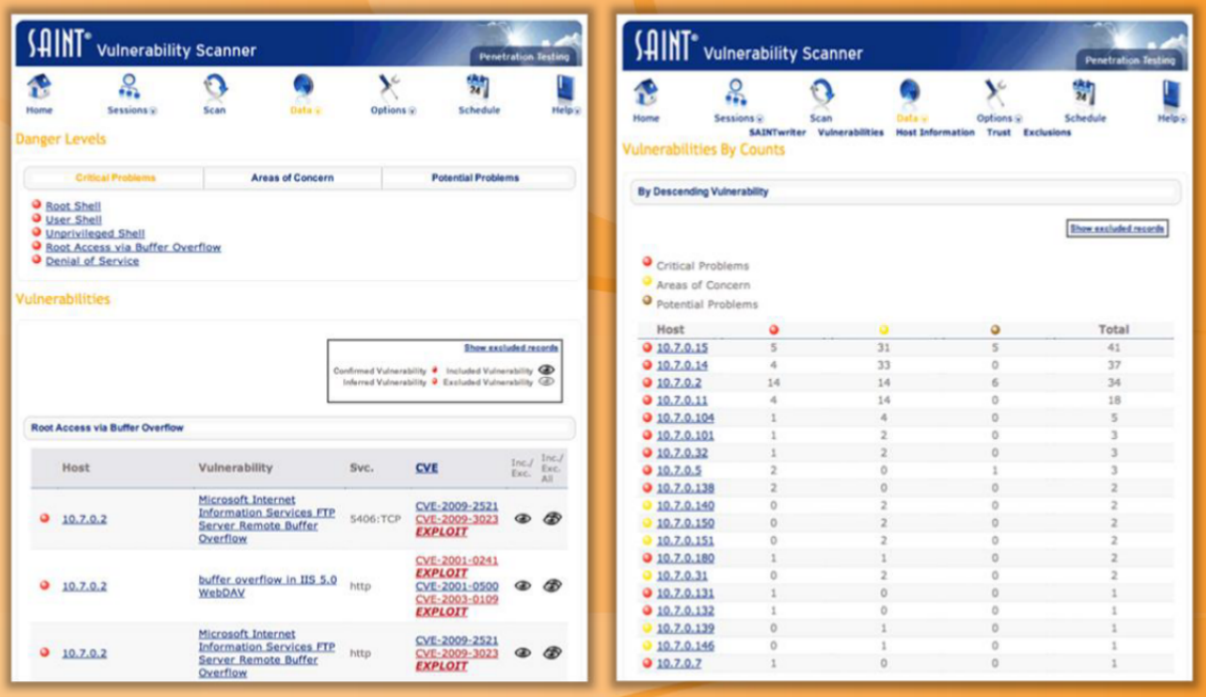
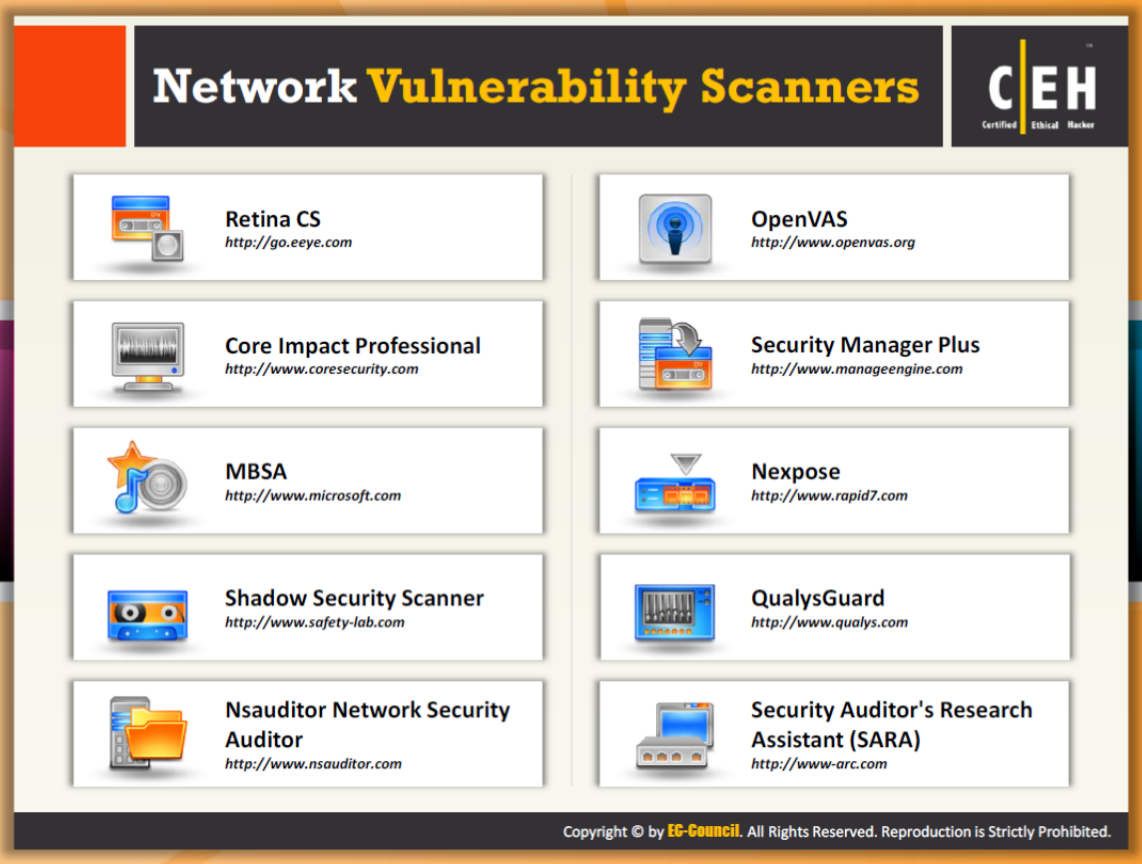


FIGURE 3.51: SAINT Screenshots

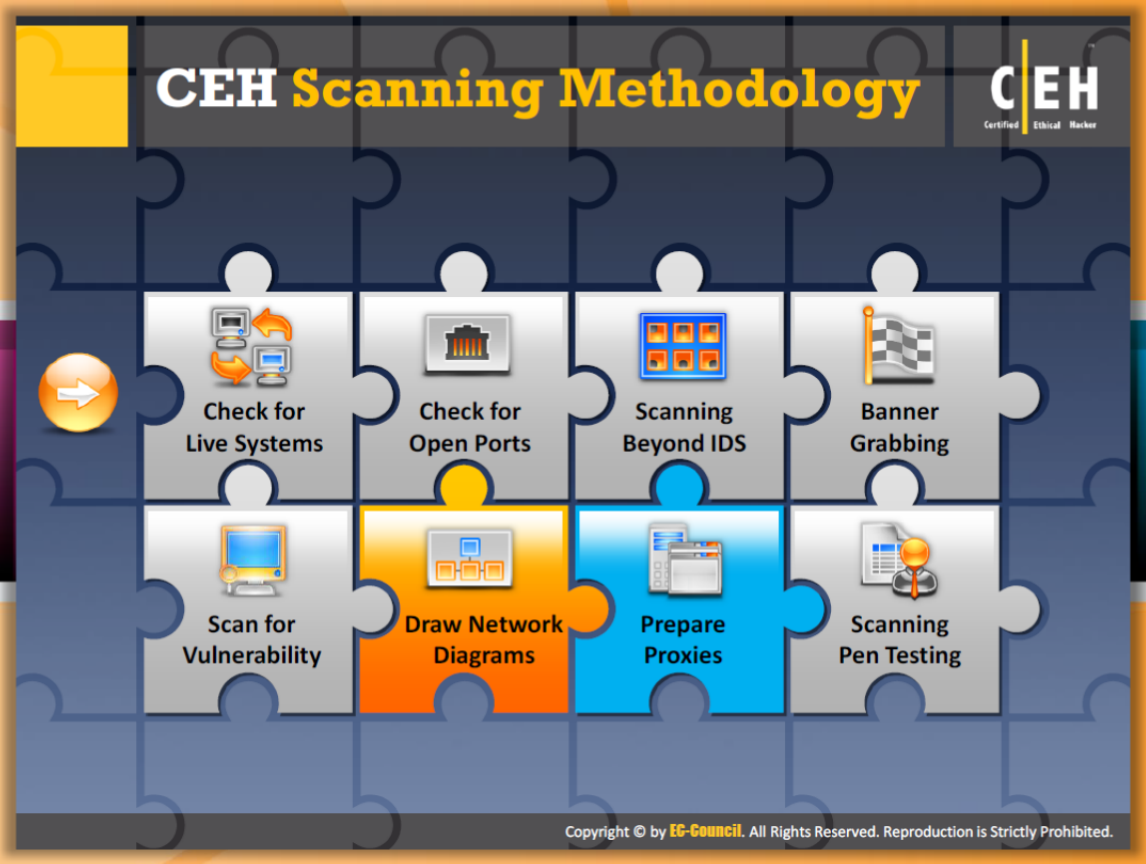


Network Vulnerability Scanners

Network vulnerability scanners are the tools that assist you in identifying the vulnerabilities in the **target network** or **network resources**. These scanners help you in vulnerability assessment and network auditing. Using these scanners, you can find vulnerabilities in networks, wired or wireless, operating systems, security configuration, server tuning, open ports, applications, etc.

A few network vulnerability scanners and their home sites are mentioned as follows, using which you can perform network scanning:

- Retina CS available at http://go.eeye.com
- Core Impact Professional available at http://www.coresecurity.com
- MBSA available at http://www.microsoft.com
- Shadow Security Scanner available at http://www.safety-lab.com
- Nsauditor Network Security Auditor available at http://www.nsauditor.com
- OpenVAS available at http://www.openvas.org
- Security Manager Plus available at http://www.manageengine.com
- Nexpose available at http://www.rapid7.com
- QualysGuard available at http://www.qualys.com
- Security Auditor's Research Assistant (SARA) available at http://www-arc.com

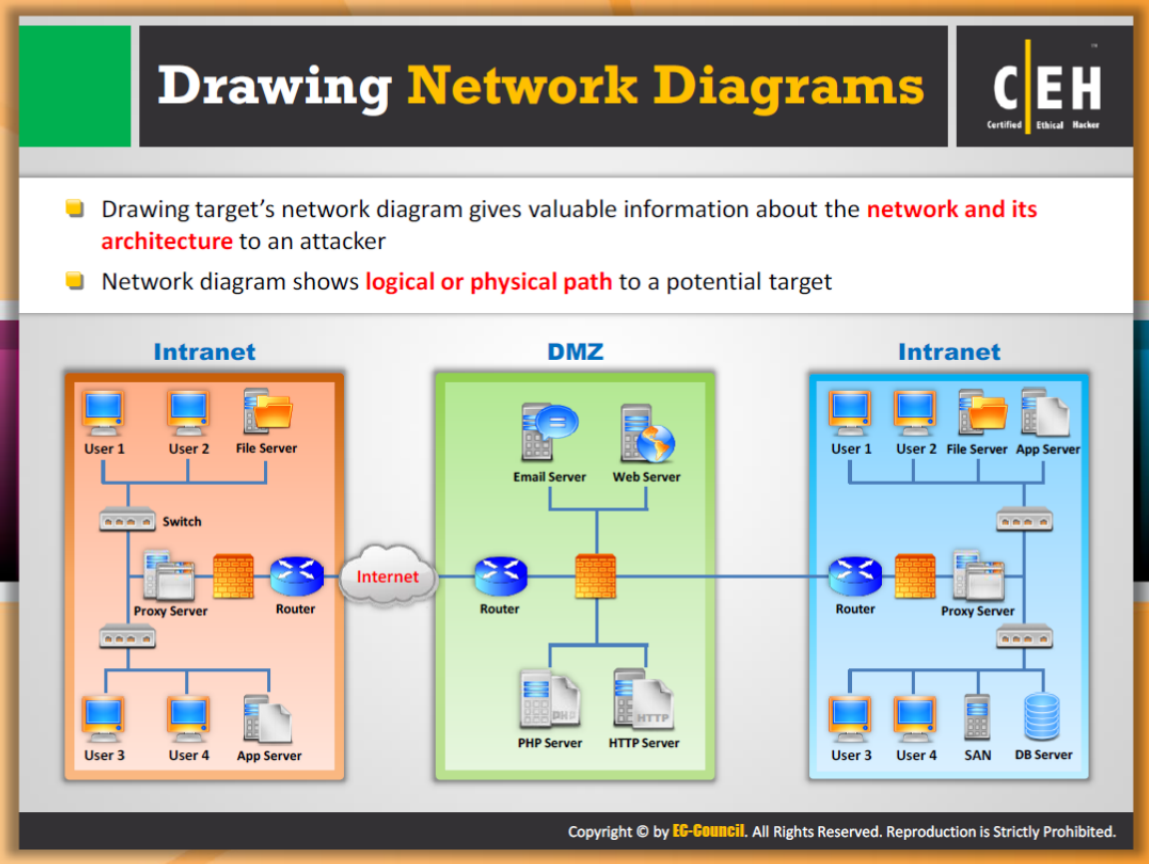


CEH Scanning Methodology

So far, we discussed various scanning concepts such as sources to be scanned, tools that can be used for scanning, and vulnerability scanning. Now we will discuss the network diagram, an important diagram that enables you to analyze the **complete network topology**.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section highlights the importance of the network diagram, how to draw the network diagram or maps, how attackers can use these launching attacks, and the tools that can be used to draw network maps.



Drawing Network Diagrams

The mapping of networks into diagrams helps you to **identify the topology** or the architecture of the target network. The network diagram also helps you to trace out the path to the target host in the network. It also allows you to understand the position of firewalls, routers, and other **access control devices**. Based on the network diagram, the attacker can analyze the target **network's topology** and **security mechanisms**. It helps an attacker to see the firewalls, IDSs, and other security mechanisms of the target network. Once the attacker has this information, he or she can try to figure out the vulnerabilities or weak points of those security mechanisms. Then the attacker can find his or her way into the target network by exploiting those security weaknesses.

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

The following figure depicts an example of a network diagram:

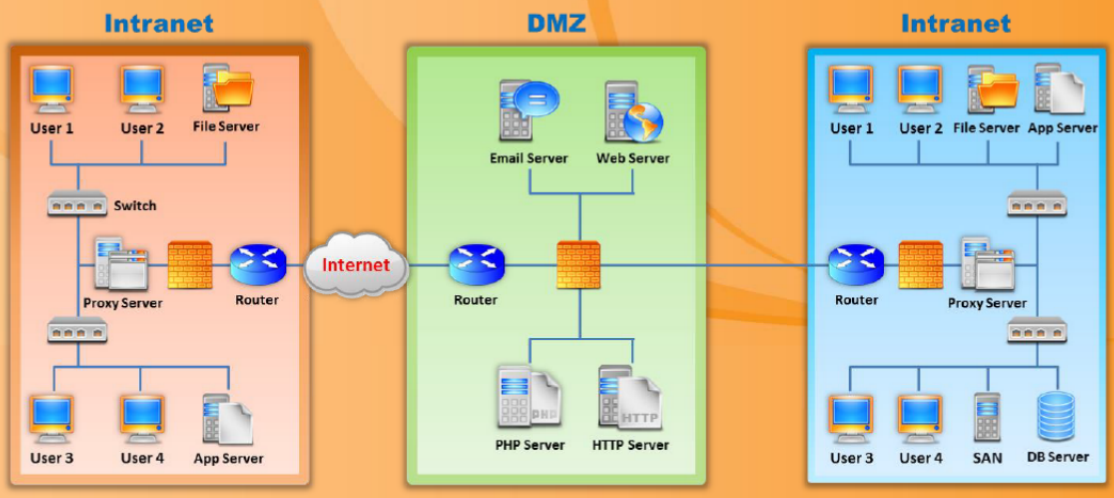
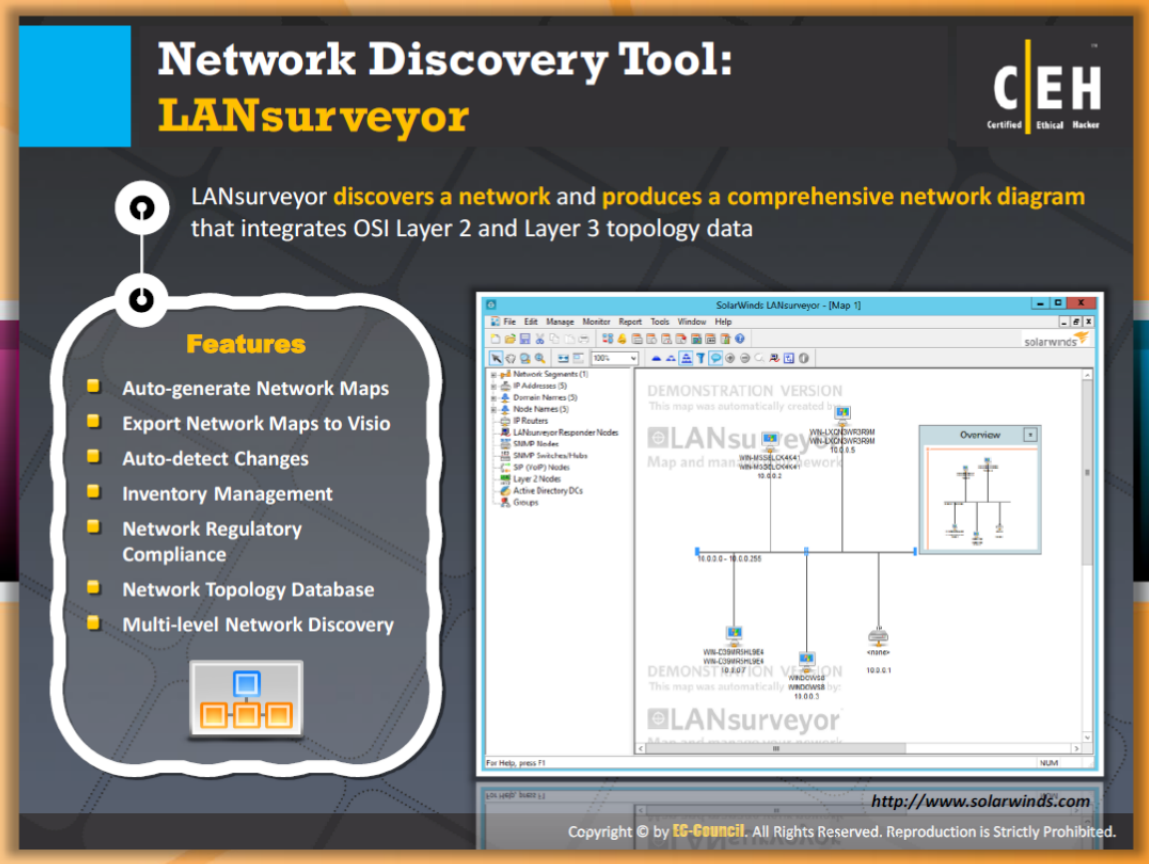


FIGURE 3.52: Network Diagram



Network Discovery Tool: LANsurveyor

Source: http://www.solarwinds.com

LANsurveyor allows you to **automatically discover** and **create a network map** of the target network. It is also able to display in-depth connections like **OSI Layer 2** and **Layer 3** topology data such as displaying switch to switch, switch to node, and switch to router connection. You can export the network map created into Microsoft Office Visio. It can also keep track of changes that occur in the network. It allows the user to perform inventory management of hardware and software assets.

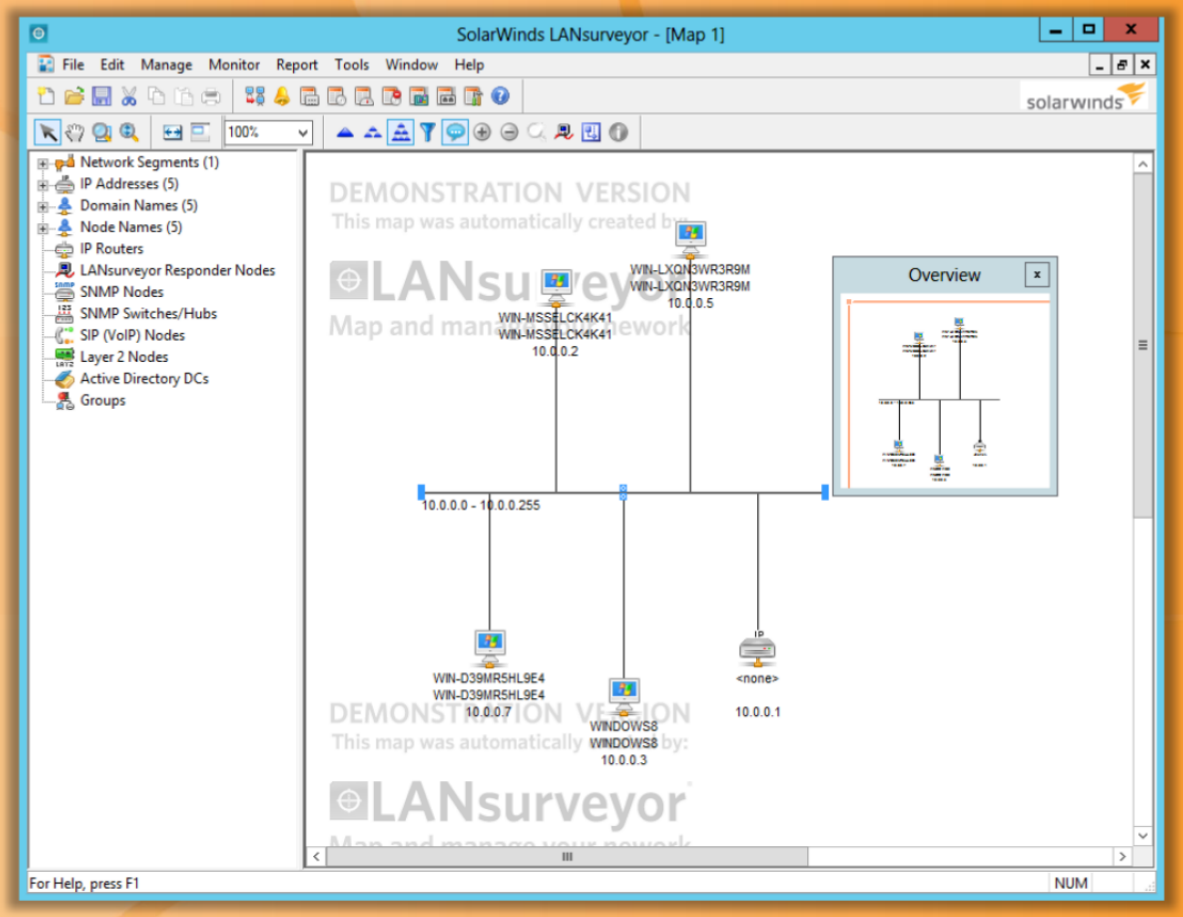
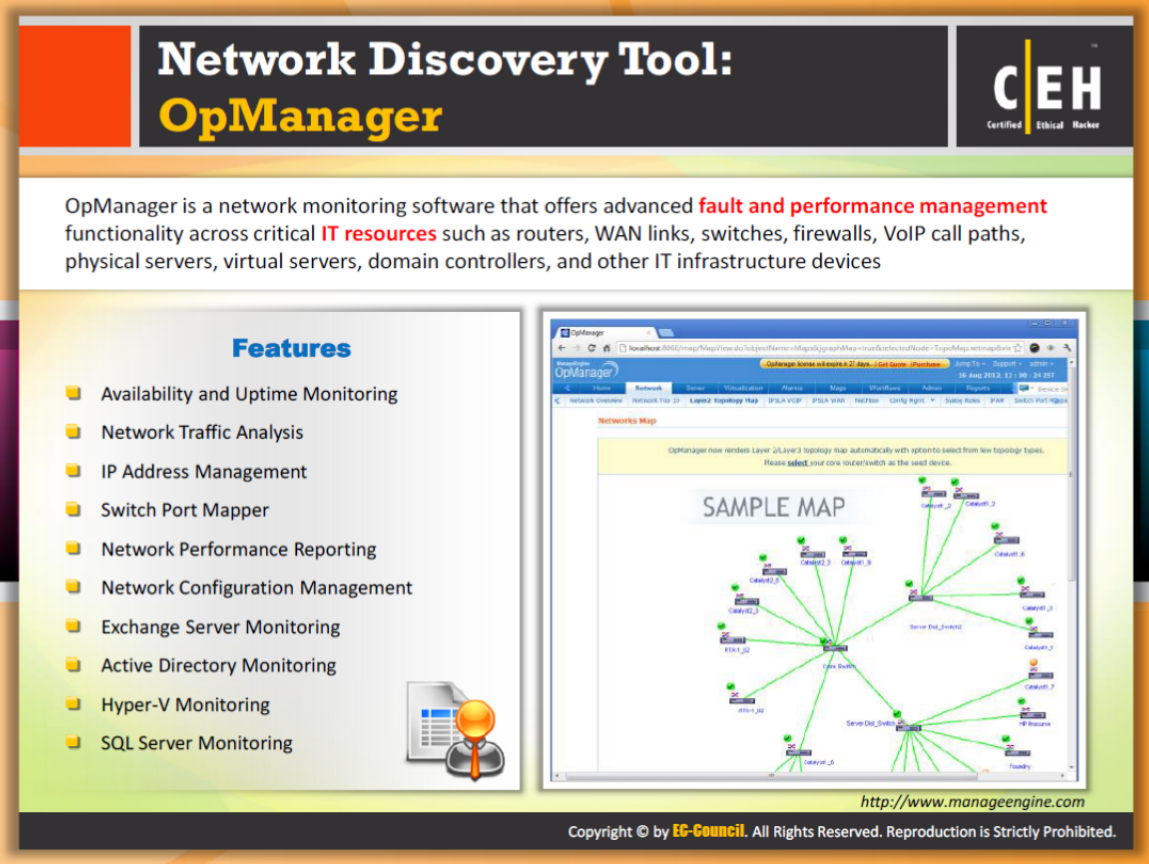


FIGURE 3.53: LANsurveyor Screenshot



Network Discovery Tool: OpManager

Source: http://www.manageengine.com

OpManager is basically a **network performance management** and **monitoring tool** that offers advanced fault and performance management functionality across critical IT resources such as routers, WAN links, switches, firewalls, VoIP call paths, physical servers, virtual servers, domain controllers, and other IT infrastructure devices. This tool is helpful in discovering the specific network automatically. It can also present a live network diagram of your network.

Here are some of the features of OpManager:

- Availability and uptime monitoring
- Network traffic analysis
- IP address management
- Switch port mapper
- Network performance reporting
- Network configuration management
- Exchange server monitoring

- Active directory monitoring
- Hyper-V monitoring
- SQL Server monitoring

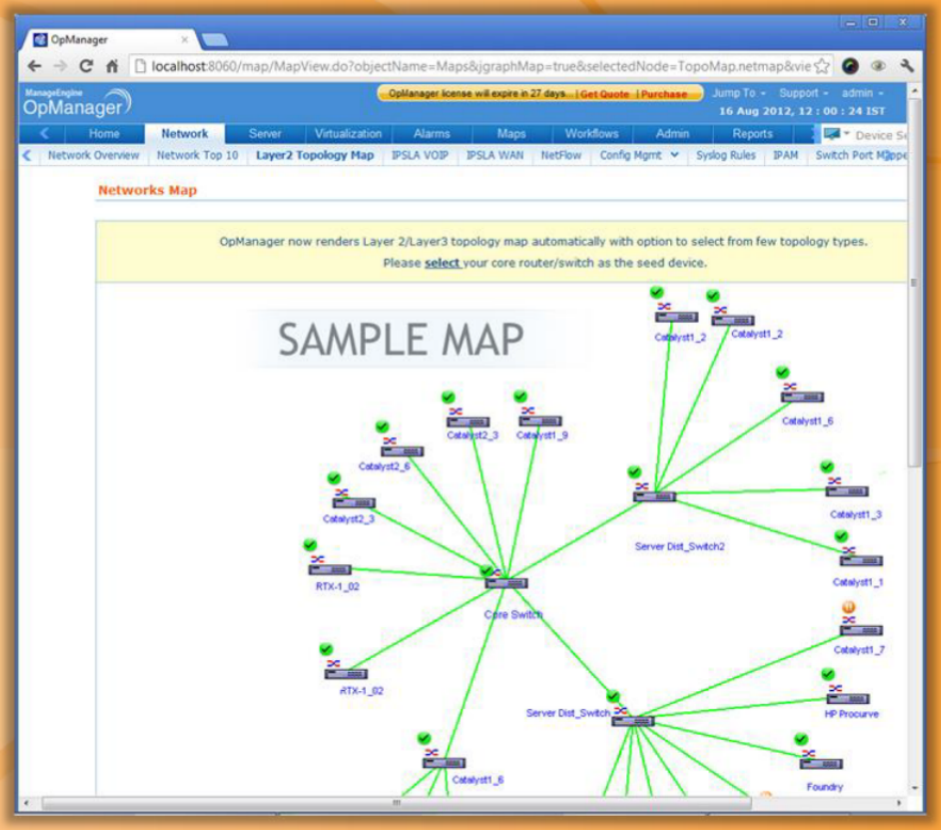
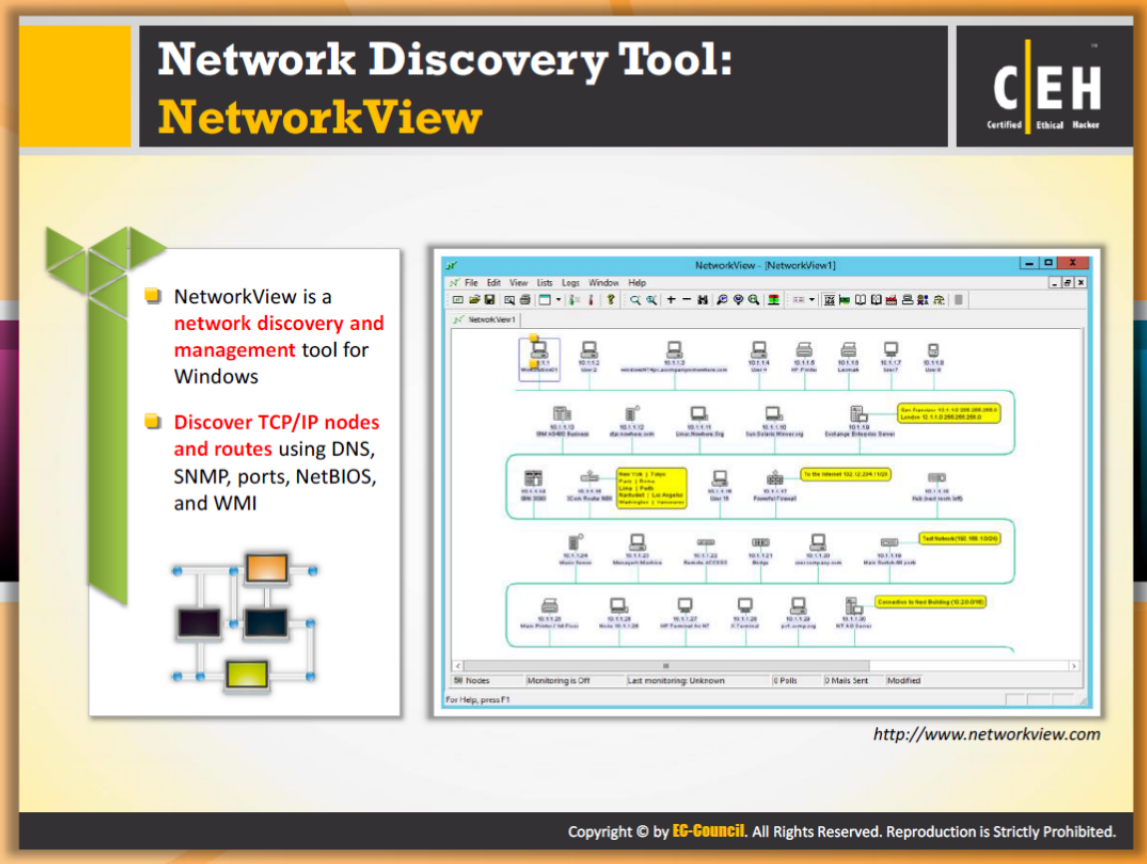


FIGURE 3.54: OpManager showing Sample Map



Network Discovery Tool: NetworkView

Source: http://www.networkview.com

NetworkView is a **network discovery** and **management tool** for Windows.

Its key features include:

- Discover TCP/IP nodes and routes using DNS, SNMP, Ports, NetBIOS, and WMI
- Get **MAC addresses** and NIC manufacturer names
- Monitor nodes and receive alerts
- Document with printed **maps** and **reports**
- Control and secure your network with the **SNMP MIB browser**, the **WMI browser**, and the **port scanner**

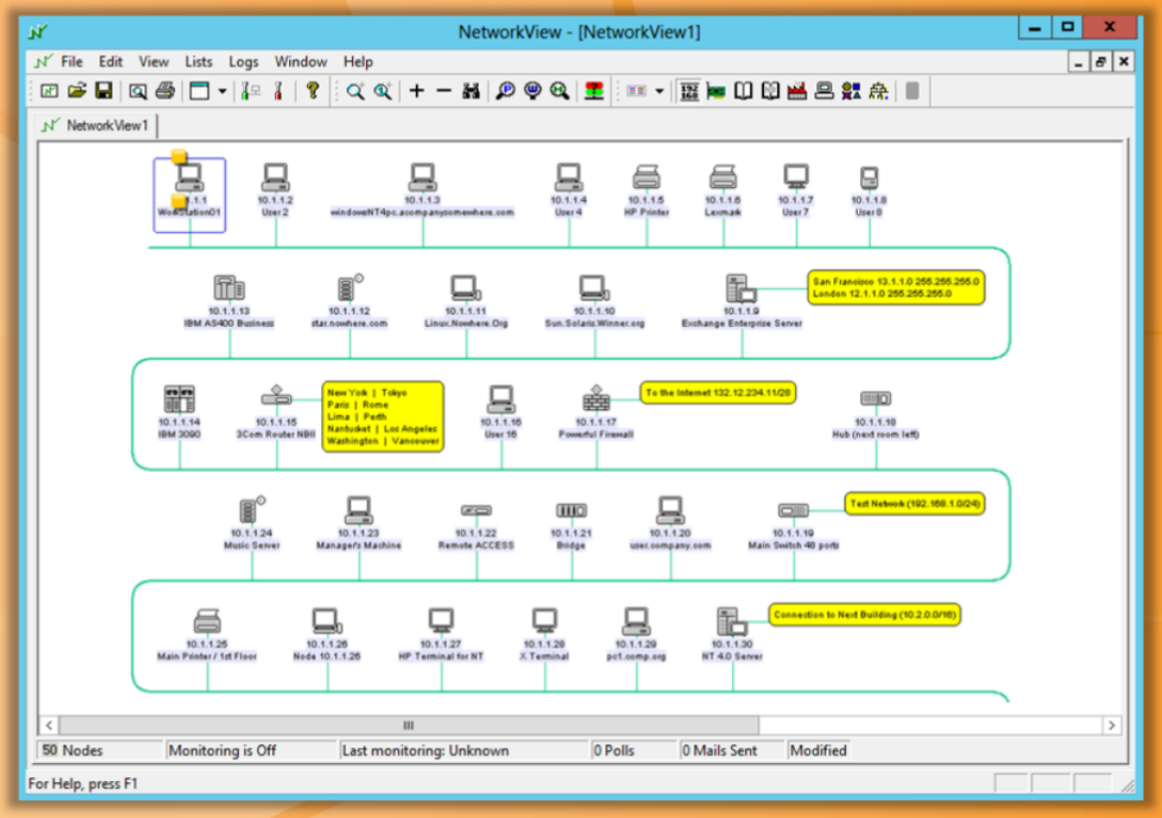
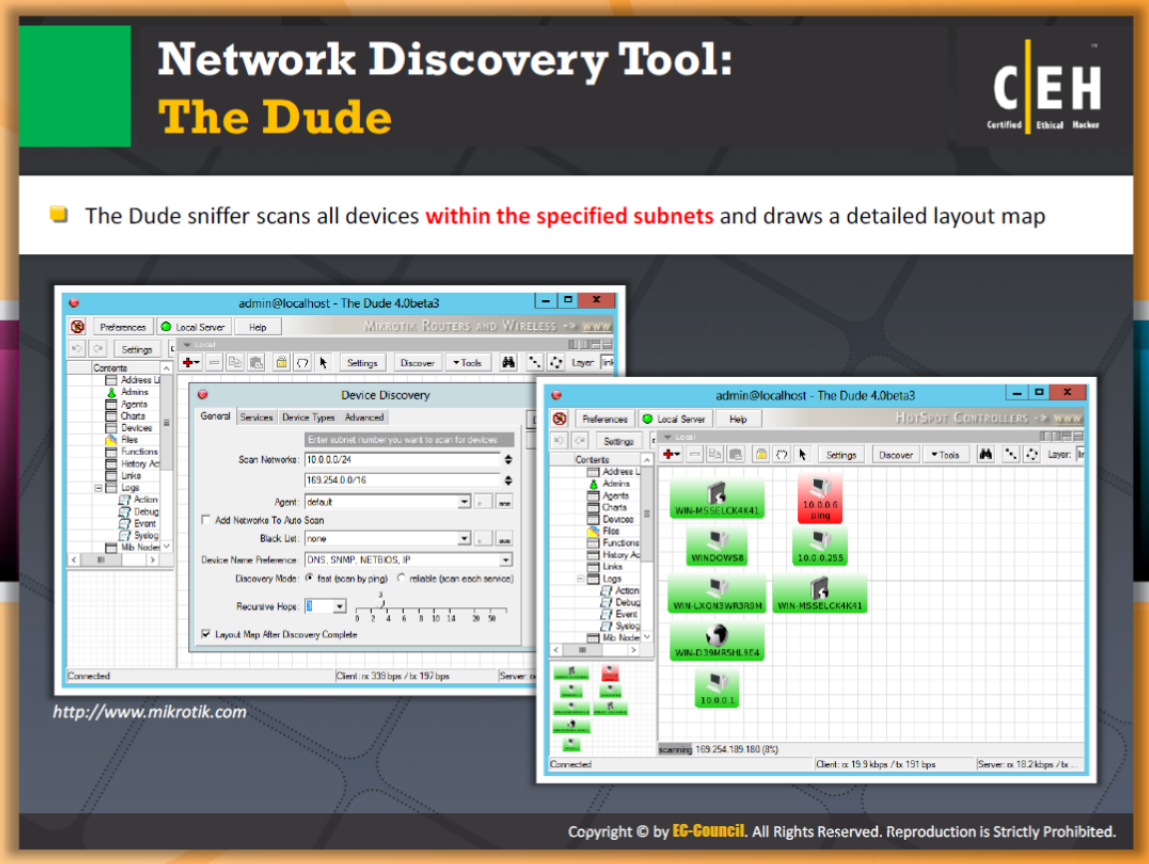


FIGURE 3.55: NetworkView Screenshot



Network Discovery Tool: The Dude

Source: http://www.mikrotik.com

The Dude will **automatically scan** all devices within specified subnets, draw and lay out a map of your networks, monitor services of your devices, and alert you in case any service has problems.

A few features of the Dude include:

- Auto network discovery and layout
- Discovers any type or brand of device
- Device, link monitoring, and notifications
- Allows you to draw your own maps and add custom devices
- Supports SNMP, ICMP, DNS, and TCP monitoring for devices that support it
- Direct access to remote control tools for device management
- Supports remote Dude server and local client

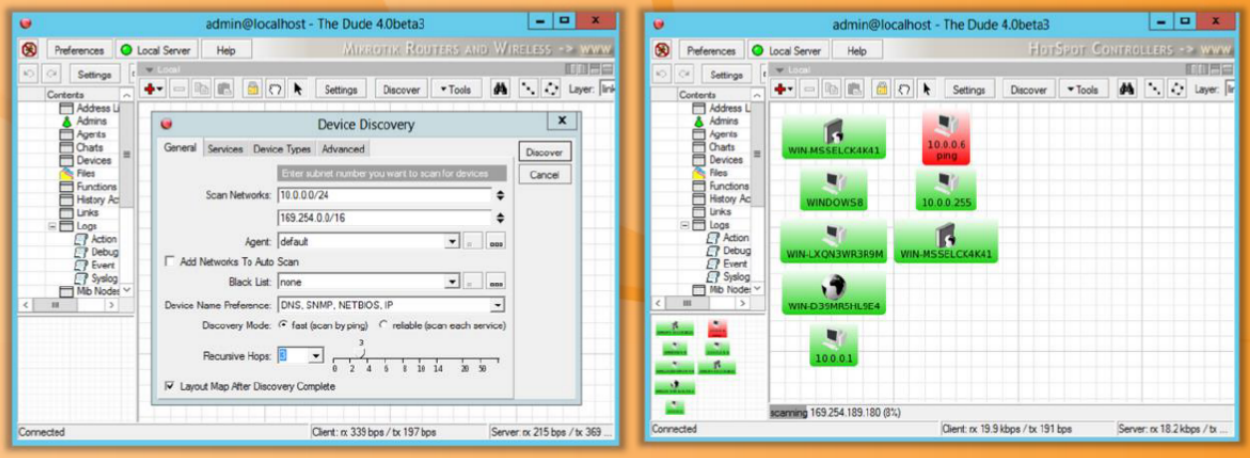
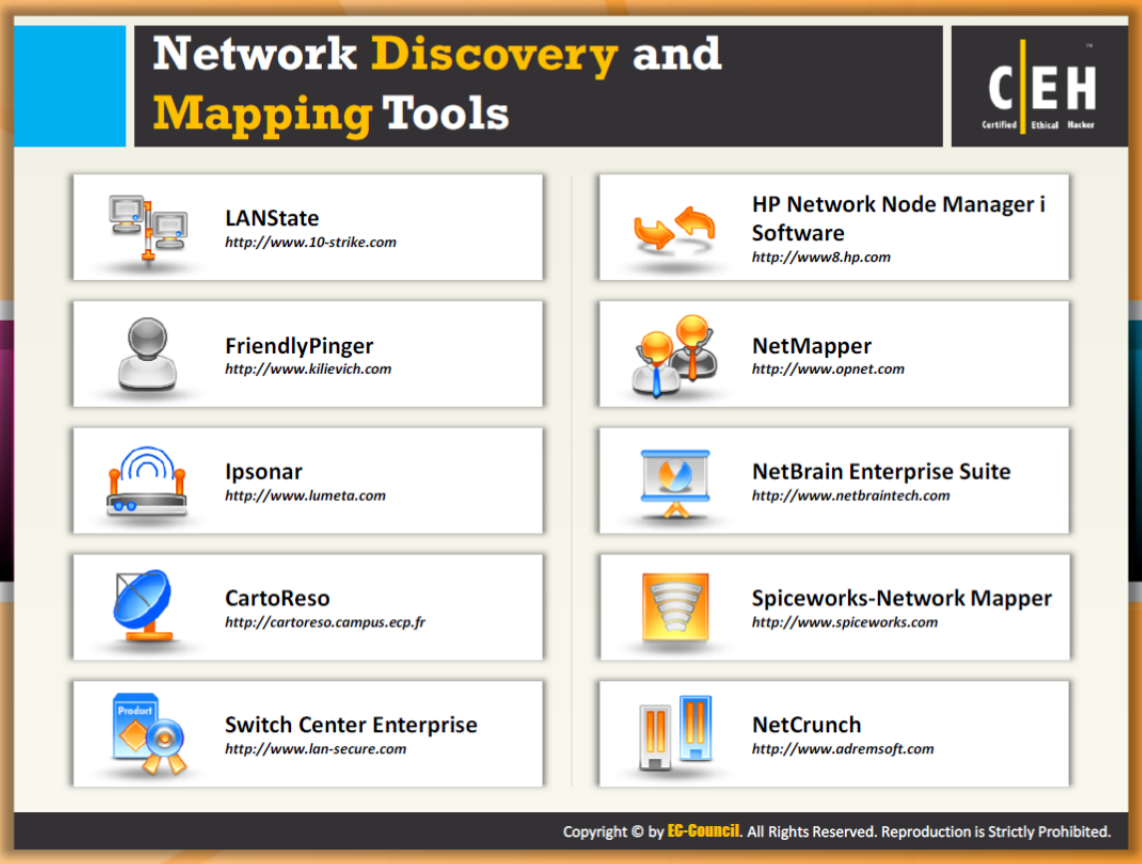


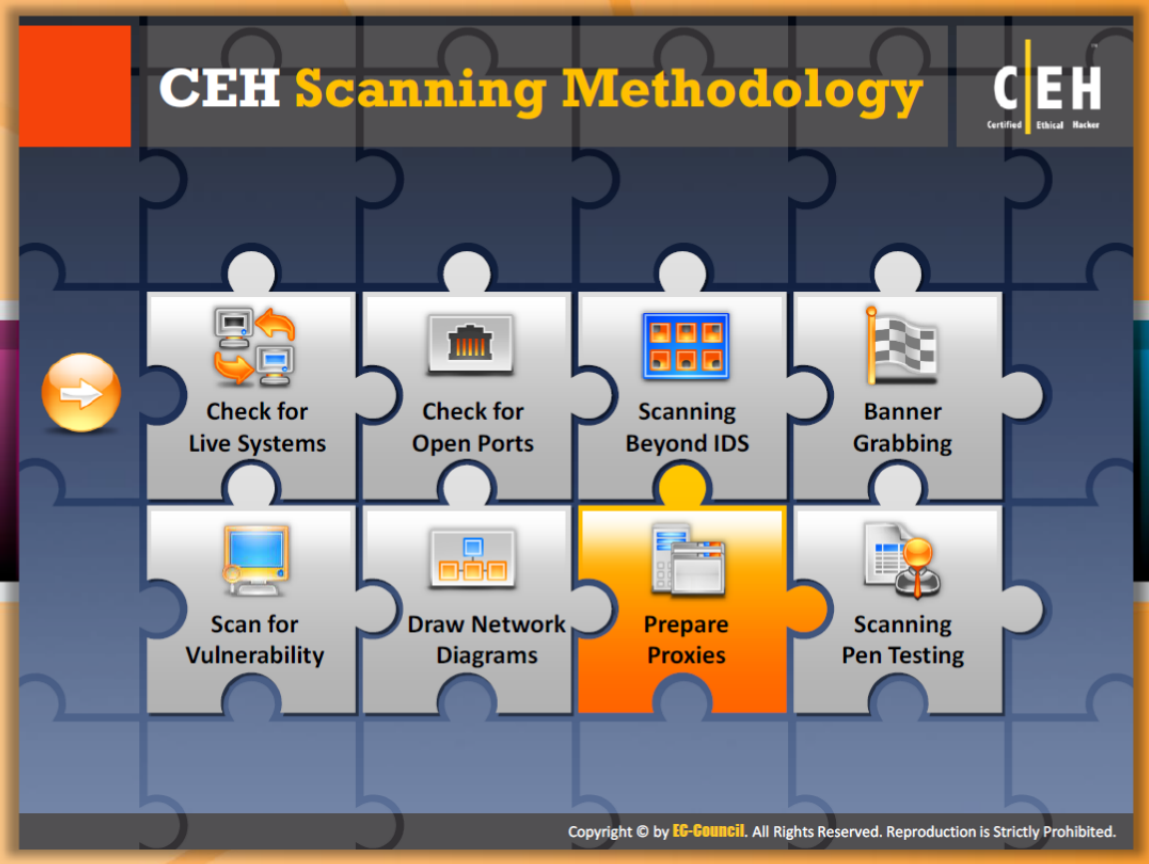
FIGURE 3.56: The Dude Screenshots



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**. It notifies you whenever a particular host becomes active or goes down. Thus, you can also figure out the server outages or problems related to performance. This is the purpose of network discovery and mapping tools in terms of security. The **same tools** can be used by attackers to launch attacks on your network. Using these tools, the attacker draws the network diagram of the target network, analyzes the topology, find outs the vulnerabilities or weak points, and launches an attack by exploiting them. The attacker may use the following tools to create a map of the network:

- LANState available at http://www.10-strike.com
- FriendlyPinger avaialble at http://www.kilievich.com
- Ipsonar available at http://www.lumeta.com
- CartoReso available at http://cartoreso.campus.ecp.fr
- Switch Center Enterprise available at http://www.lan-secure.com
- HP Network Node Manager i Software available at http://www8.hp.com
- NetMapper available at http://www.opnet.com
- NetBrain Enterprise Suite available at http://www.netbraintech.com
- Spiceworks-Network Mapper available at http://www.spiceworks.com
- NetCrunch available at http://www.adremsoft.com

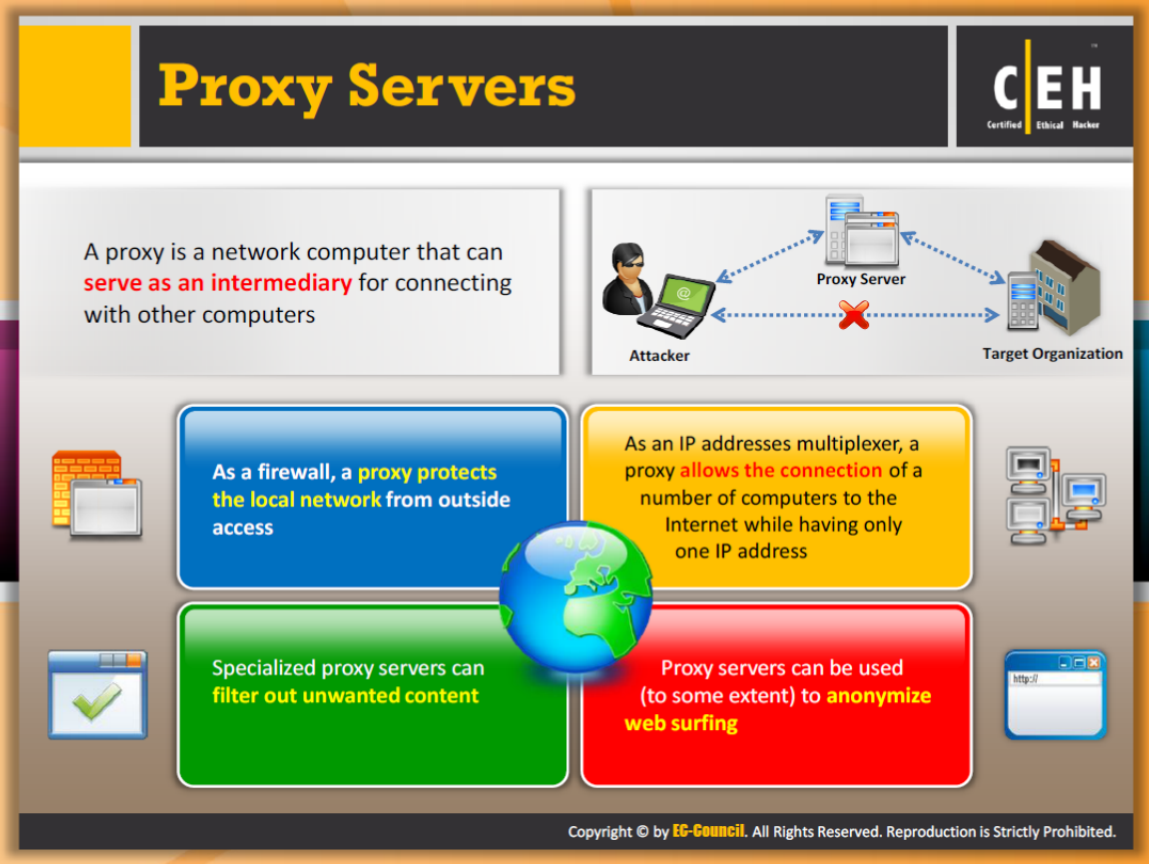


CEH Scanning Methodology

So far, we have discussed various means of scanning and the sources to be scanned. Now we will discuss proxies and important mechanisms used by attackers to access the restricted sources and also to avoid their identity.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section describes how to prepare proxies and how an attacker can use them to launch attacks.



Proxy Servers

A proxy is a **network computer** that can serve as an intermediary for connecting with other computers. You can use a proxy server in many ways such as:

- As a firewall, a proxy protects the local network from the outside access
- As an IP address multiplexer, a proxy allows a number of computers to connect to the Internet when you have only one IP address
- To anonymize web surfing (to some extent)
- To filter out unwanted content, such as ads or "unsuitable" material (using specialized proxy servers)
- To provide some protection against hacking attacks
- To save bandwidth

Let's see how a proxy server works.

When you use a proxy to request a particular web page on an actual server, it first sends your **request** to the **proxy server**. The proxy server then **sends** your request to the **actual server** on

behalf of your request, i.e., it mediates between you and the actual server to send and respond to the request as shown in the following figure.

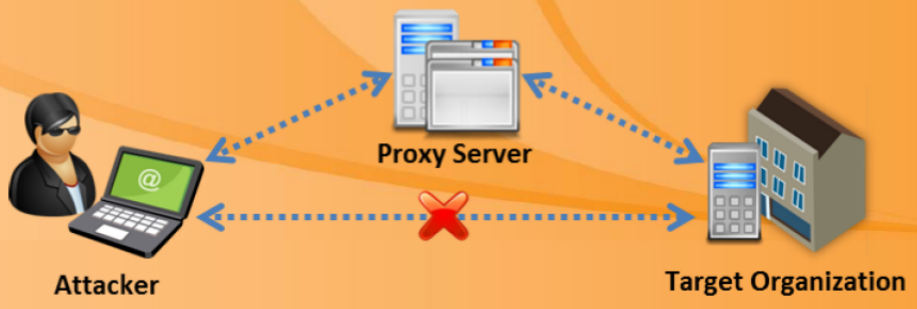


FIGURE 3.57: Attacker using Proxy Server

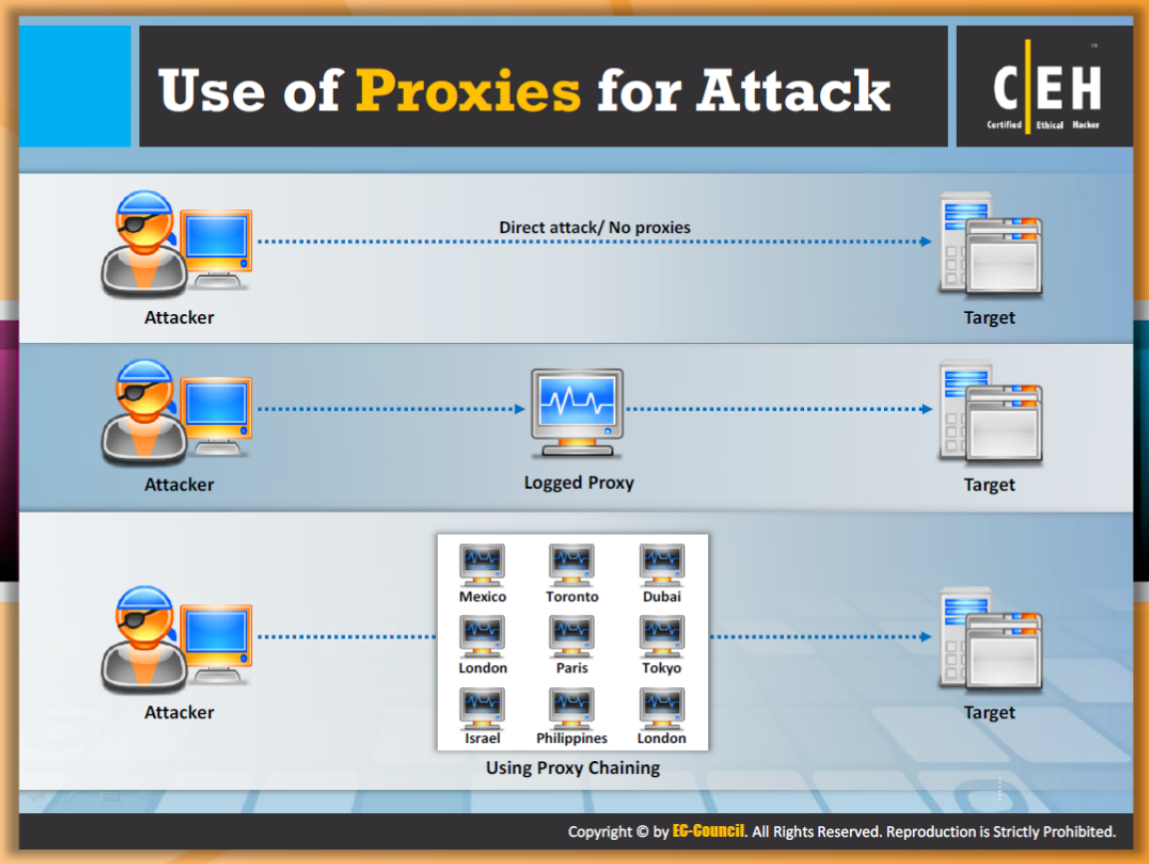
In this process, the proxy receives the communication between the **client** and the **destination** application. In order to take advantage of a proxy server, **client programs** must be configured so they can send their requests to the proxy server instead of the final destination.

Why Attackers Use Proxy Servers

For an attacker, it is easy to **attack** or **hack** a particular system than to conceal the attack source**.** So the main challenge for an attacker is to hide his identity so that no one can trace him or her. To conceal the identity, the attacker uses the proxy server. The main cause behind using a proxy is to **avoid detection** of **attack evidence**. With help of the proxy server, an attacker can **mask** his or her **IP address** so that he or she can hack the computer system without any fear of legal repercussion. When the attacker uses a proxy to connect to the destination, the proxy's source address will be recorded in the server logs instead of the actual source address of the attacker.

In addition to this, the reasons for which attackers use proxy servers include:

- Attacker appears in a victim server's log files with a fake source address of the proxy rather than with the attacker's actual address
- To remotely access intranets and other website resources that are normally off limits
- To interrupt all the requests sent by an attacker and transmit them to a third destination, hence victims will only be able to identify the proxy server address
- To use multiple proxy servers for scanning and attacking, making it difficult for administrators to trace the real source of attack



Use of Proxies for Attack

Quite a number of proxies are intentionally open to easy access. **Anonymous proxies** hide the **real IP address** (and sometimes other information) from websites that the user visits. There are two types or **anonymous proxies**: One that can be used in the same way as the non-anonymous proxies and others that are web-based anonymizers.

Let's see how many different ways that attackers can use proxies to commit attacks on the target.

Case 1: In the first case, the attacker performs attacks directly without using proxy. The attacker may be at risk to be traced out as the server logs may contain information about the IP address of the source.

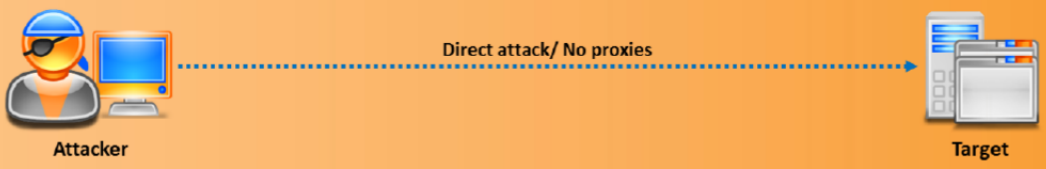


FIGURE 3.58: Attacker Communicating with Target Directly (No Proxy)

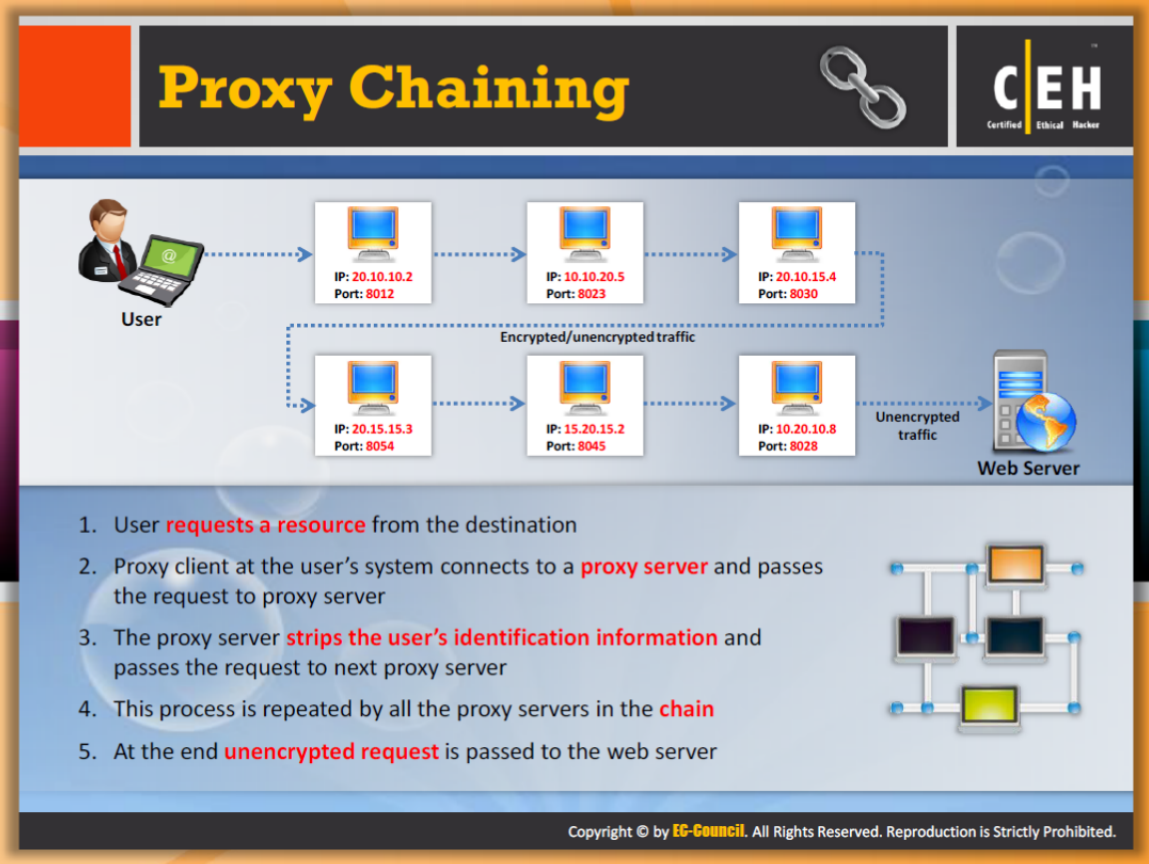
Case 2: The attacker uses the proxy to fetch the target application. In this case, the server log will show the IP address of the proxy instead of the attacker's IP address, thereby hiding his or

her identity; thus, the attacker will be at minimum risk of being caught. This will give the attacker the chance to be anonymous on the Internet.

FIGURE 3.59: Attacker Communicating with Target through Proxy

Case 3: To become more anonymous on the Internet, the attacker may use the proxy chaining technique to fetch the target application. If he or she uses proxy chaining, then it is highly difficult to trace out his or her IP address. Proxy chaining is a technique of using more numbers of proxies to fetch the target.

FIGURE 3.60: Attacker using Proxy Chaining for the attack

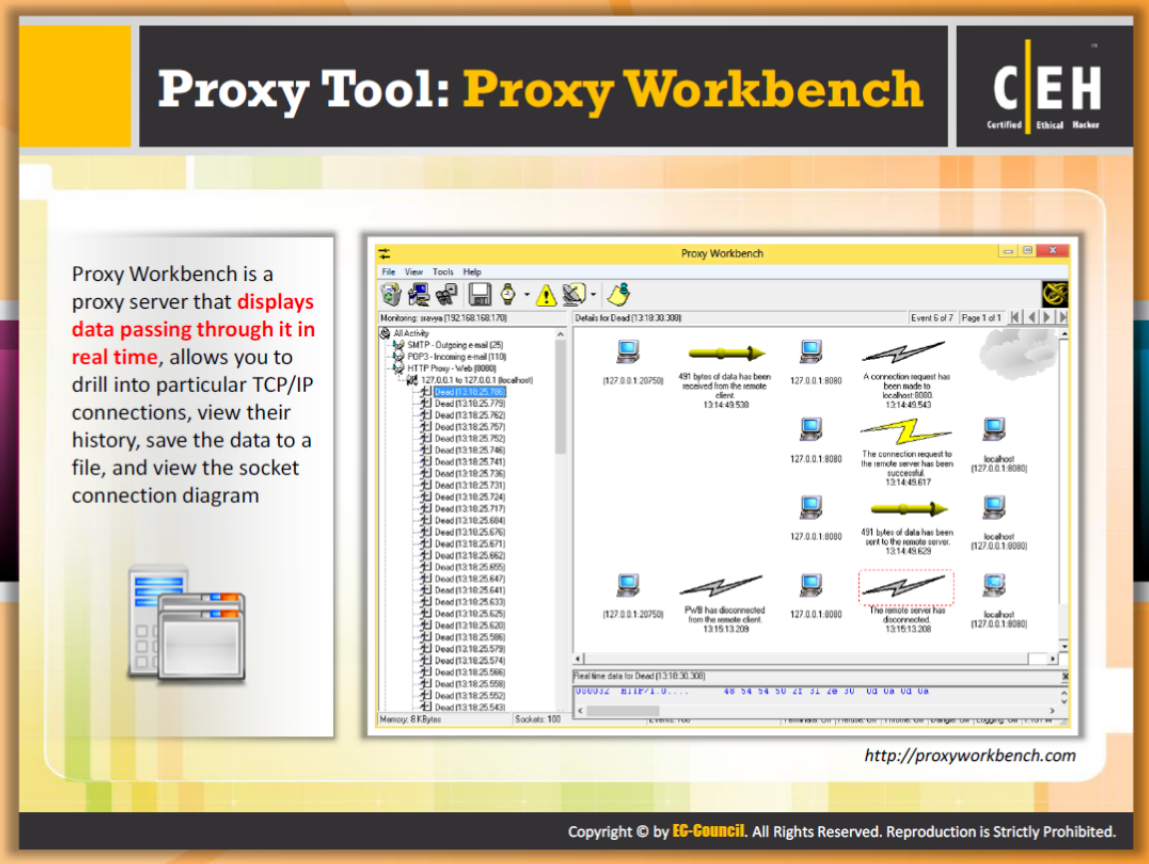


Proxy Chaining

Proxy chaining helps you to become **more anonymous** on the Internet. Your anonymity on the Internet depends on the number of proxies used for fetching the target application. If you use a larger number of **proxy servers**, then you will become more anonymous on the Internet and vice versa.

When the attacker first requests the proxy server1, this proxy server1 in turn requests another proxy server2. The proxy server1 strips the user's identification information and passes the request to the next proxy server. This may again request another proxy server, server3, and so on, up to target server, where finally the request is sent. Thus, it forms the chain of the proxy server to reach the destination server as shown in the following figure:

FIGURE 3.61: Proxy Chaining



Proxy Tool: Proxy Workbench

Source: http://proxyworkbench.com

Proxy Workbench is a **proxy server** that displays the **data passing** through it in real time, 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. It is able to handle **HTTPS (secure sockets)** and **POP3** natively.

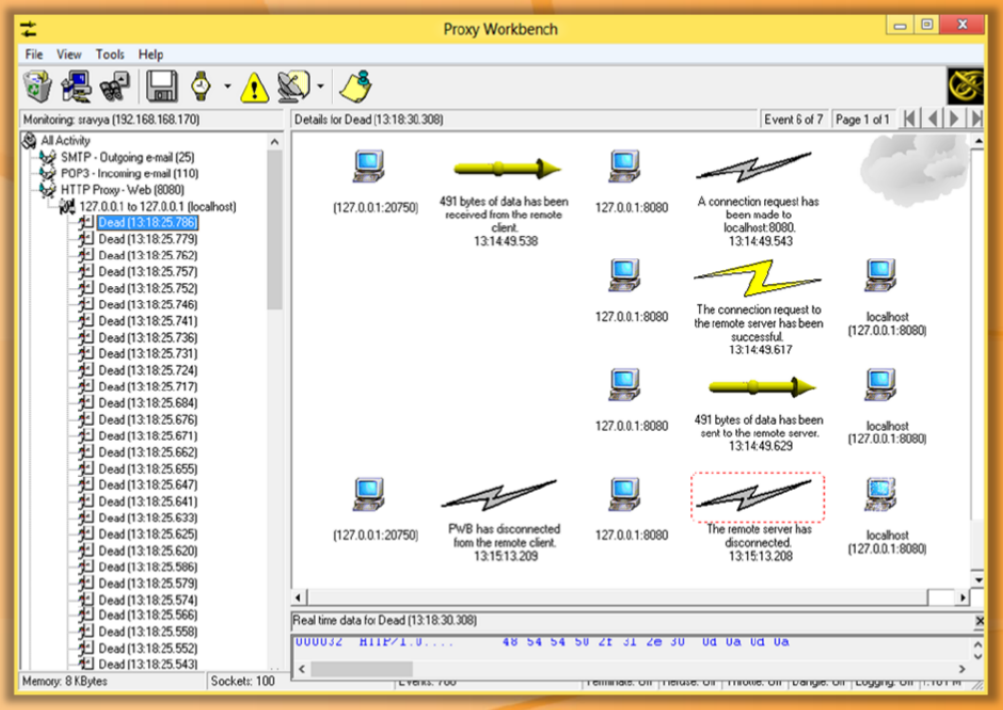


FIGURE 3.62: Proxy Workbench Screenshot

Proxy Tool: Proxifier

Source: http://www.proxifier.com

Proxifier allows **network applications** that do not support working through proxy servers to operate through a **SOCKS** or **HTTPS** proxy and chains. It allows you to surf websites that are restricted or blocked by your government, organization, etc. by bypassing the firewalls rules.

Features:

- You can access the Internet from a restricted network through a proxy server gateway
- It hides your IP address
- It can work through a chain of proxy servers using different protocols
- It allows you to bypass firewalls and any access control mechanisms

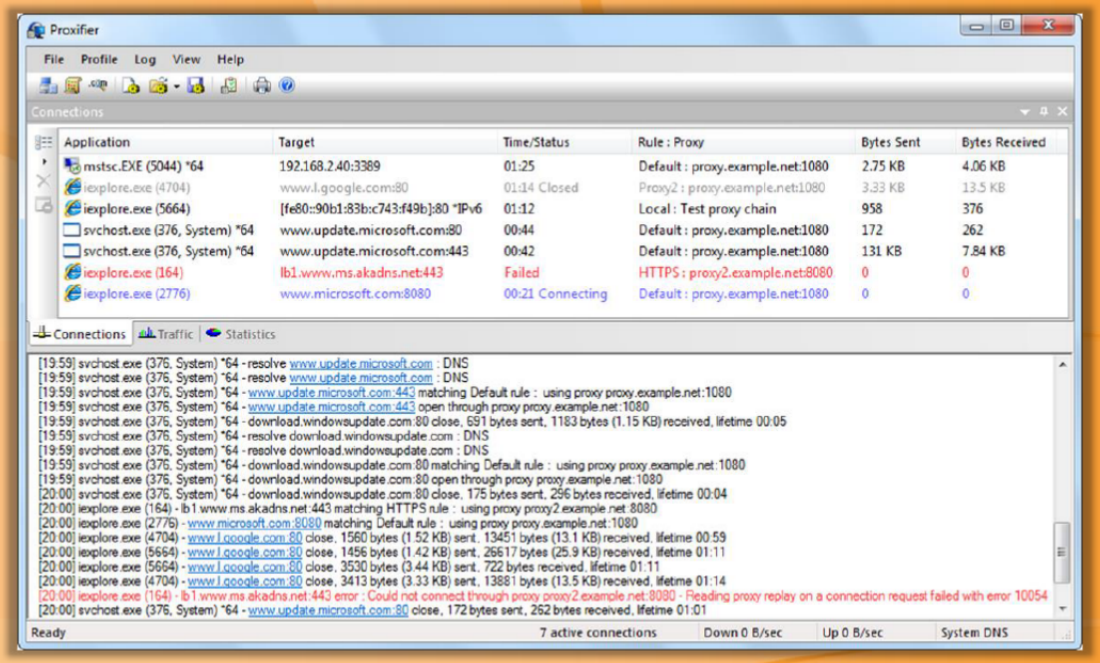


FIGURE 3.63: Proxifier Screenshot

Proxy Tool: Proxy Switcher

Source: http://www.proxyswitcher.com

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

Features:

- It hides your IP address
- It allows you to access restricted sites
- It has full support of password-protected servers

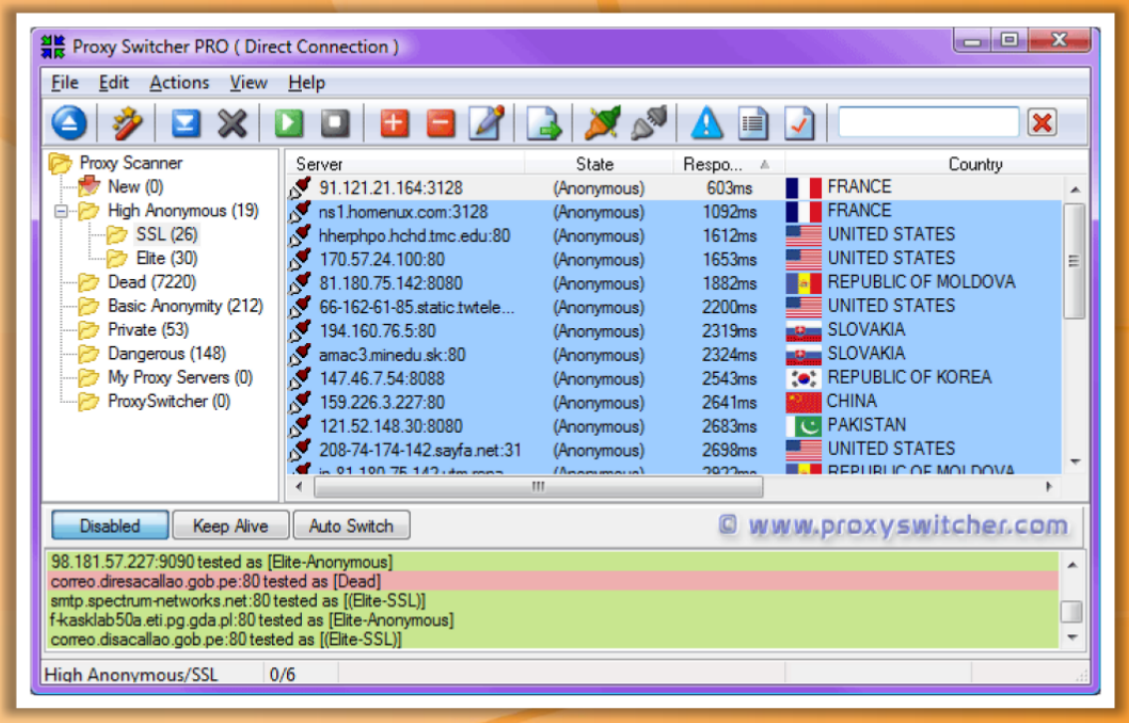


FIGURE 3.64: Proxy Switcher PRO Screenshot

Proxy Tool: SocksChain

Source: http://ufasoft.com

SocksChain is a program that allows you to work with any **Internet service** through a chain of **SOCKS** or **HTTP proxies** to hide the real IP address. It can function as a usual SOCKS-server that transmits queries through a chain of proxies. It can be used with client programs that do not support the SOCKS protocol, but work with one **TCP-connection**, such as TELNET, HTTP, IRC, etc. It hides your IP from being displayed in the server's log or mail headers.

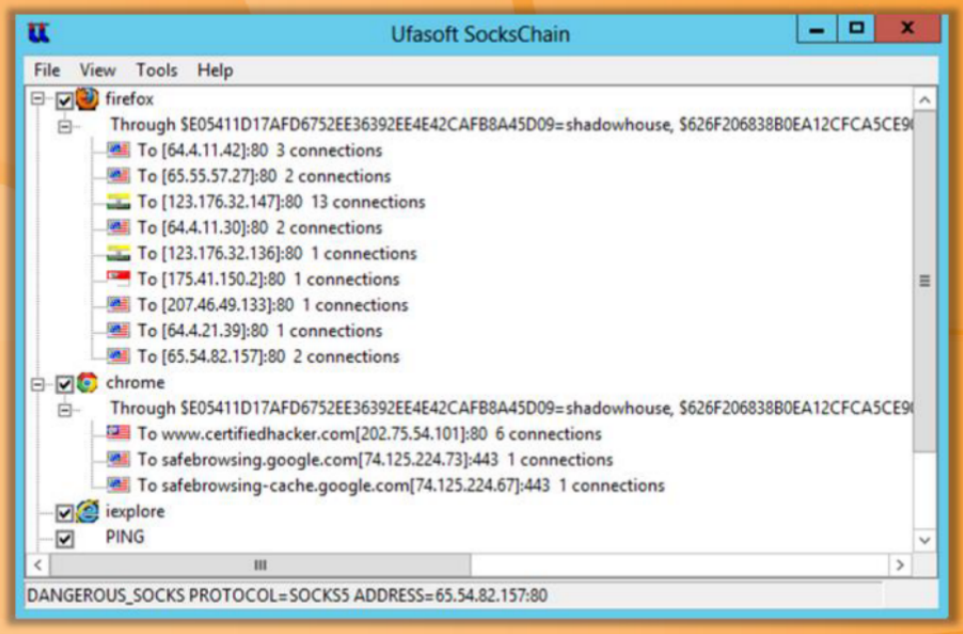
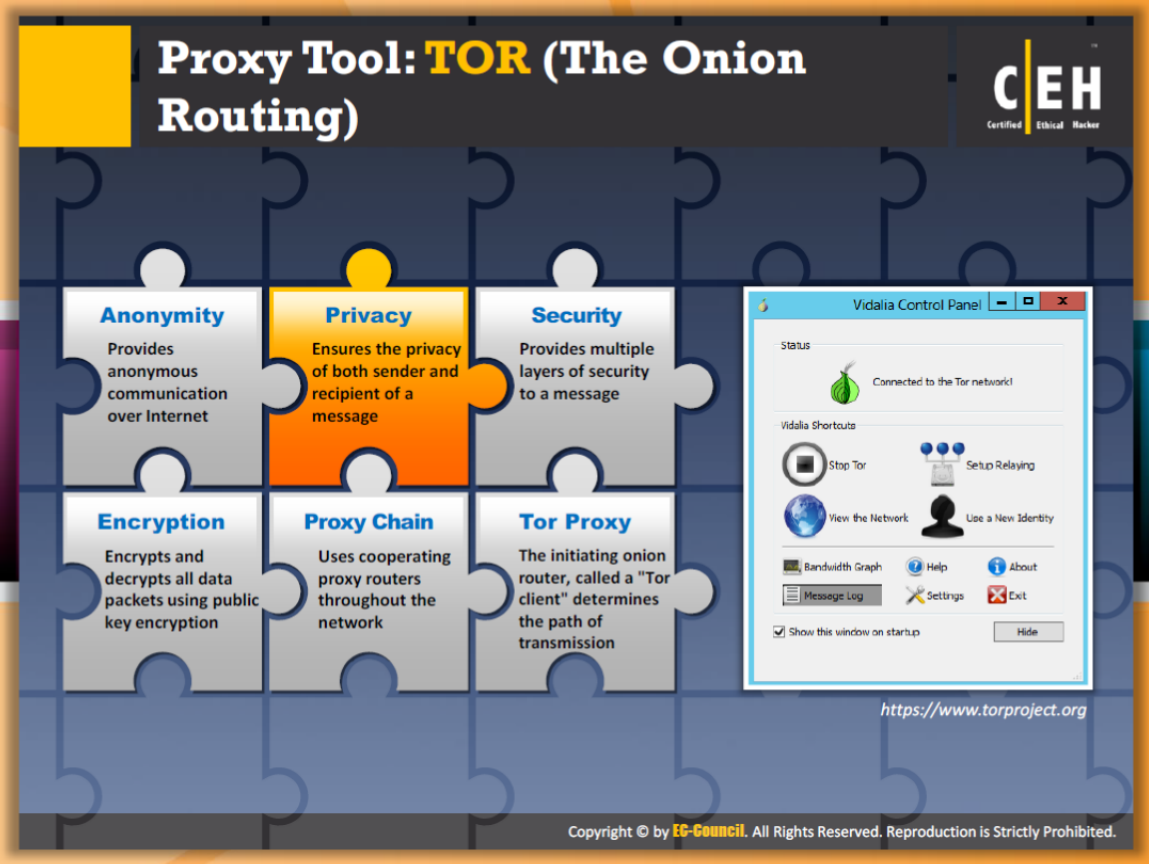


FIGURE 3.65: Ufasoft SocksChain Screenshot



Proxy Tool: TOR (The Onion Routing)

Source: https://www.torproject.org

Tor is software and an open network that helps you defend against a form of network surveillance that threatens **personal freedom** and **privacy**, confidential business activities and relationships, and state security known as traffic analysis. You can use Tor to prevent websites from tracking you on the Internet. You can also connect to news sites and instant messaging services when these sites are blocked by your network administrator. Tor makes it difficult to trace your Internet activity as it conceals a user's location or usage.

Features:

- Provides anonymous communication over the Internet
- Ensures the privacy of both sender and recipient of a message
- Provides multiple layers of security to a message
- Encrypts and decrypts all data packets using public key encryption
- Uses cooperating proxy routers throughout the network
- The initiating onion router, called a "Tor client" determines the path of transmission

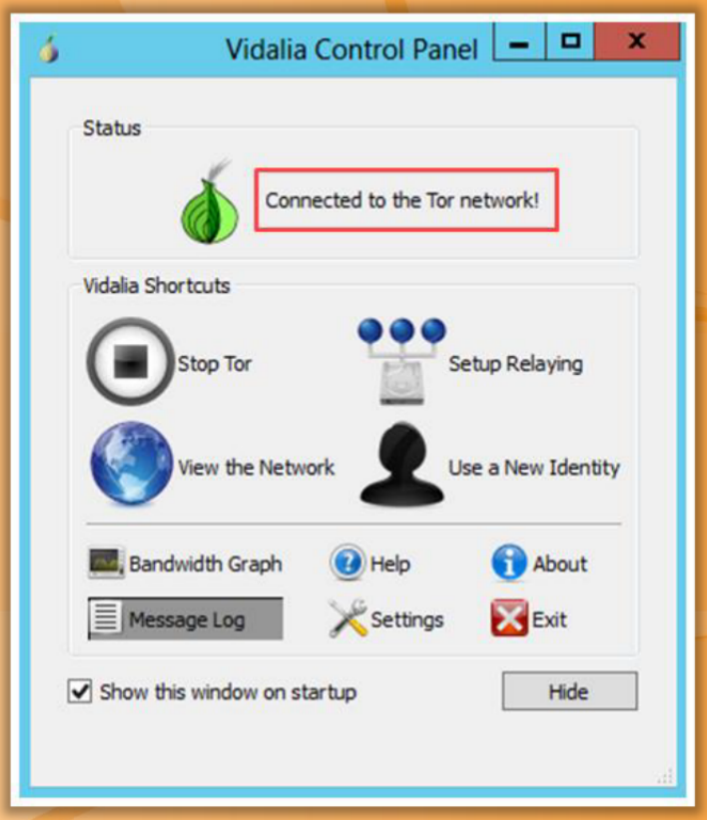
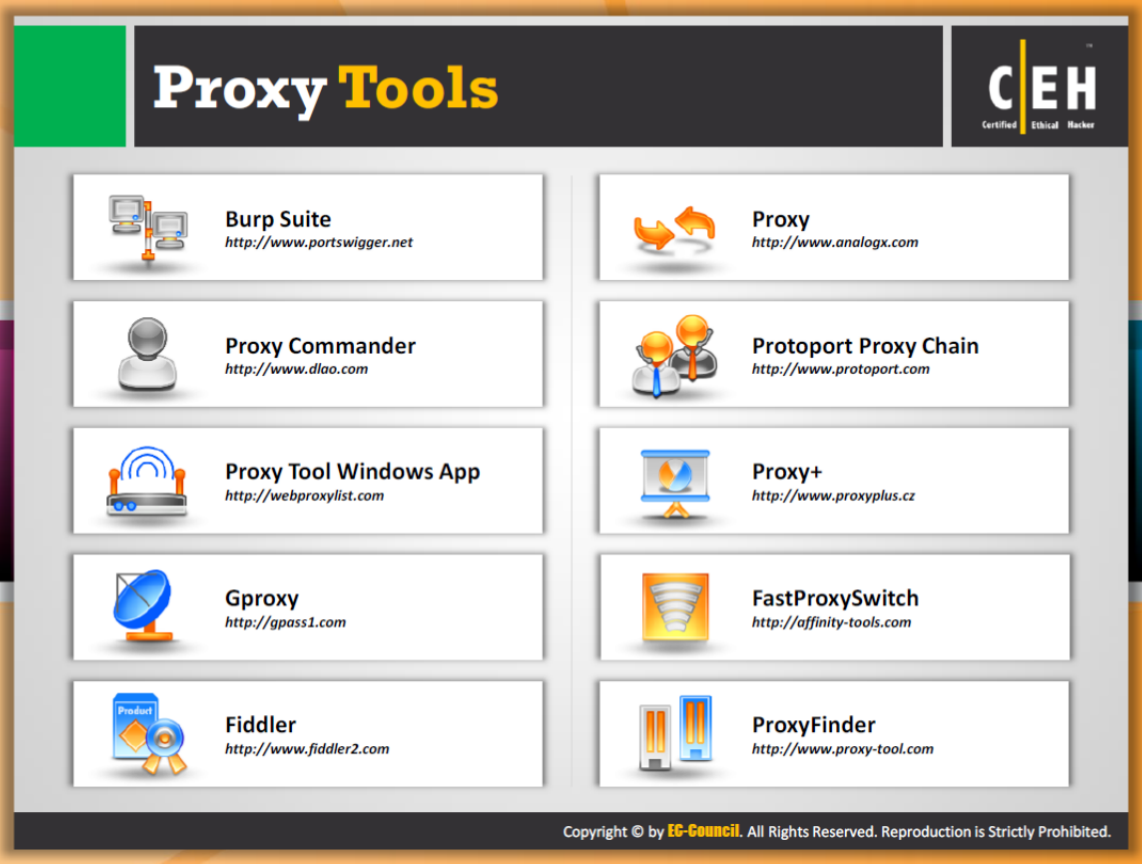


FIGURE 3.66: Vidalia Control Panel showing the Status



Proxy Tools

In addition to these proxy tools, there are many more proxy tools intended to allow users to surf the Internet anonymously. A few are listed as follows:

- Burp Suite available at http://www.portswigger.net
- Proxy Commander available at http://www.dlao.com
- Proxy Tool Windows App available at http://webproxylist.com
- Gproxy available at http://gpass1.com
- Fiddler available at http://www.fiddler2.com
- Proxy available at http://www.analogx.com
- Protoport Proxy Chain available at http://www.protoport.com
- Proxy+ available at http://www.proxyplus.cz
- FastProxySwitch available at http://affinity-tools.com
- ProxyFinder available at http://www.proxy-tool.com

Proxy Tools (Cont'd)

The list of proxy tools mentioned in the previous slide continues as follows:

- ProxyFinder Enterprise available at http://www.proxy-tool.com
- ezProxy available at http://www.oclc.org
- JAP Anonymity and Privacy available at http://anon.inf.tu-dresden.de/index_en.html
- CC Proxy Server available at http://www.youngzsoft.net
- FoxyProxy Standard available at https://addons.mozilla.org
- Socks Proxy Scanner available at http://www.mylanviewer.com
- Charles available at http://www.charlesproxy.com
- UltraSurf available at http://www.ultrasurf.us
- WideCap available at http://widecap.ru
- ProxyCap available at http://www.proxycap.com

Free Proxy Servers

Besides proxy tools discussed previously, you can find a number of free proxy sites available on the Internet that can help you to access restricted sites without revealing your IP address. Just type Free Proxy Servers in the Google search engine and you will get numerous proxy server websites.

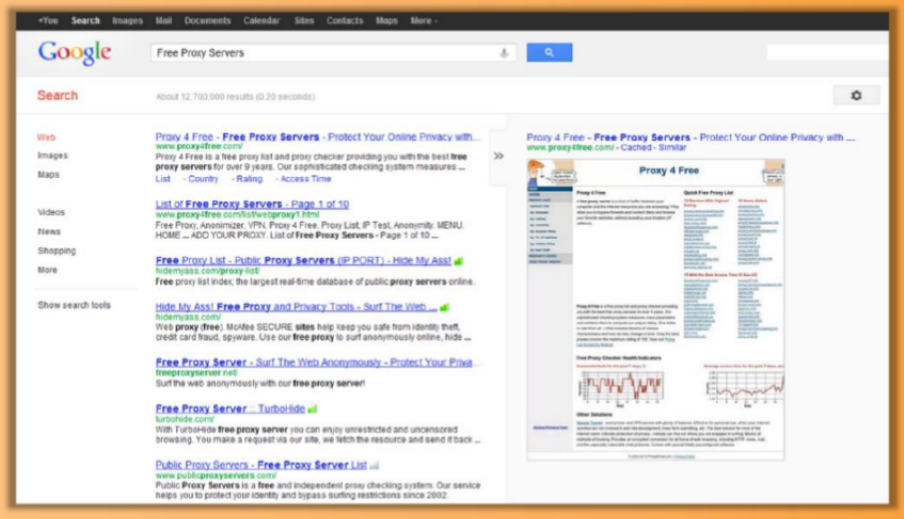
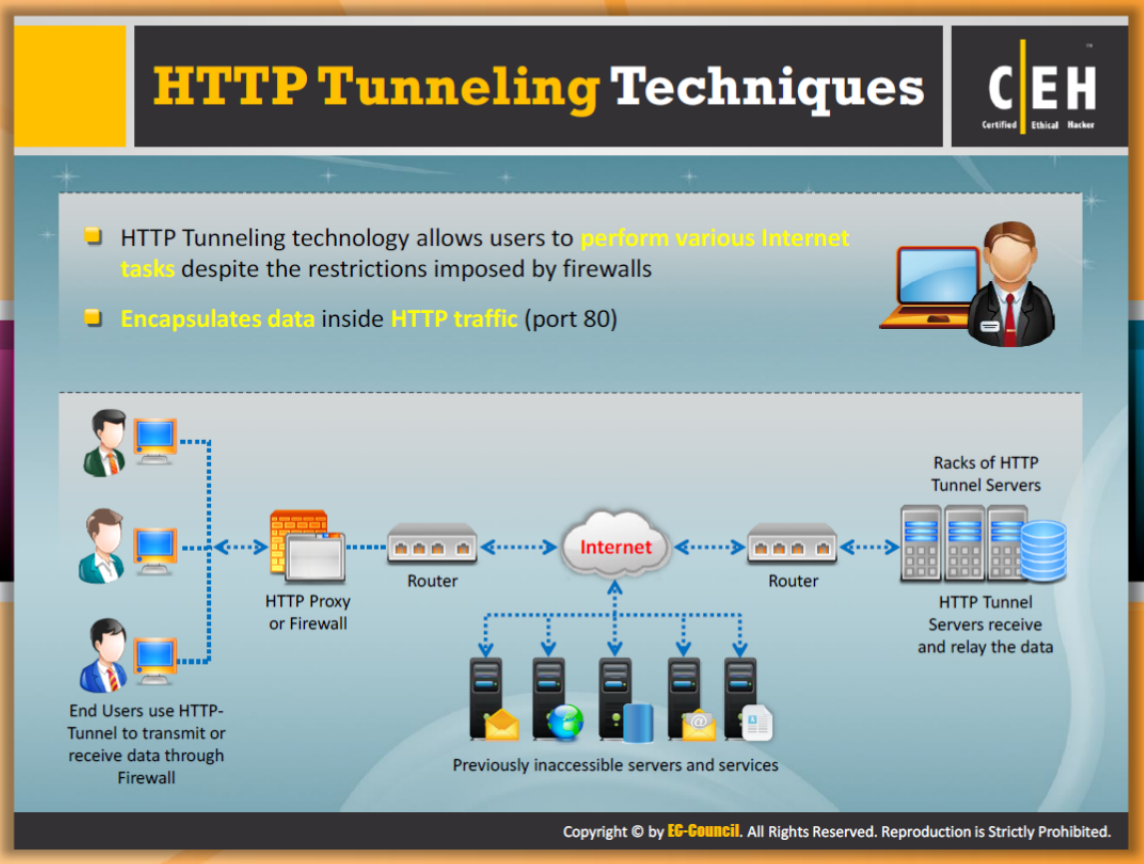


FIGURE 3.67: Google Search showing Free Proxy Servers



HTTP Tunneling Techniques

HTTP Tunneling is another technique that allows you to use the Internet despite restrictions imposed by the firewalls. The **HTTP protocol** acts as wrapper for communication channels.

An attacker uses **HTTP tunnel software** to perform HTTP tunneling. It is a client-server-based application used to communicate through the HTTP protocol. This software creates an HTTP tunnel between two machines, using a web proxy option. The technique involves sending POST requests to an "**HTTP server**" and receiving replies.

The attacker uses the client application of HTTP tunnel software installed on his or her system to communicate with other machines. All requests sent through the HTTP tunnel client application go through the HTTP protocol.

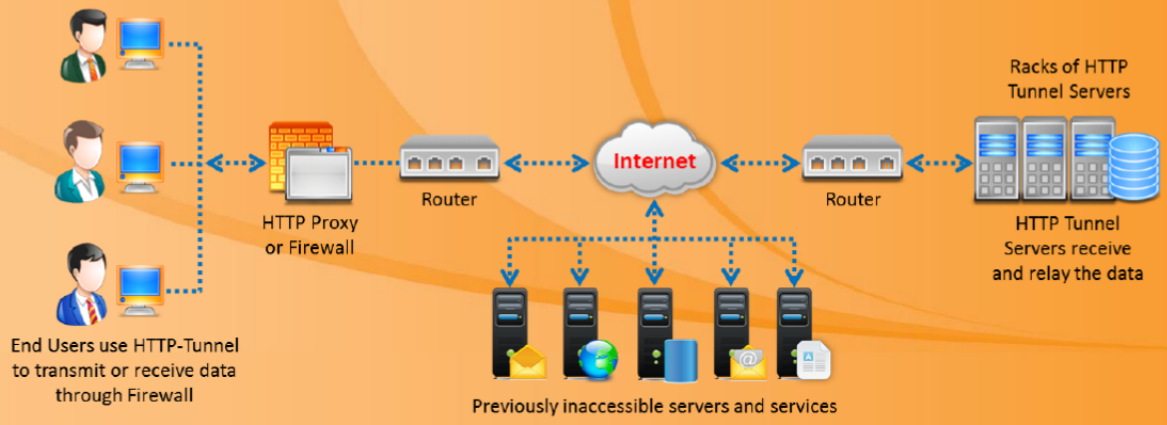
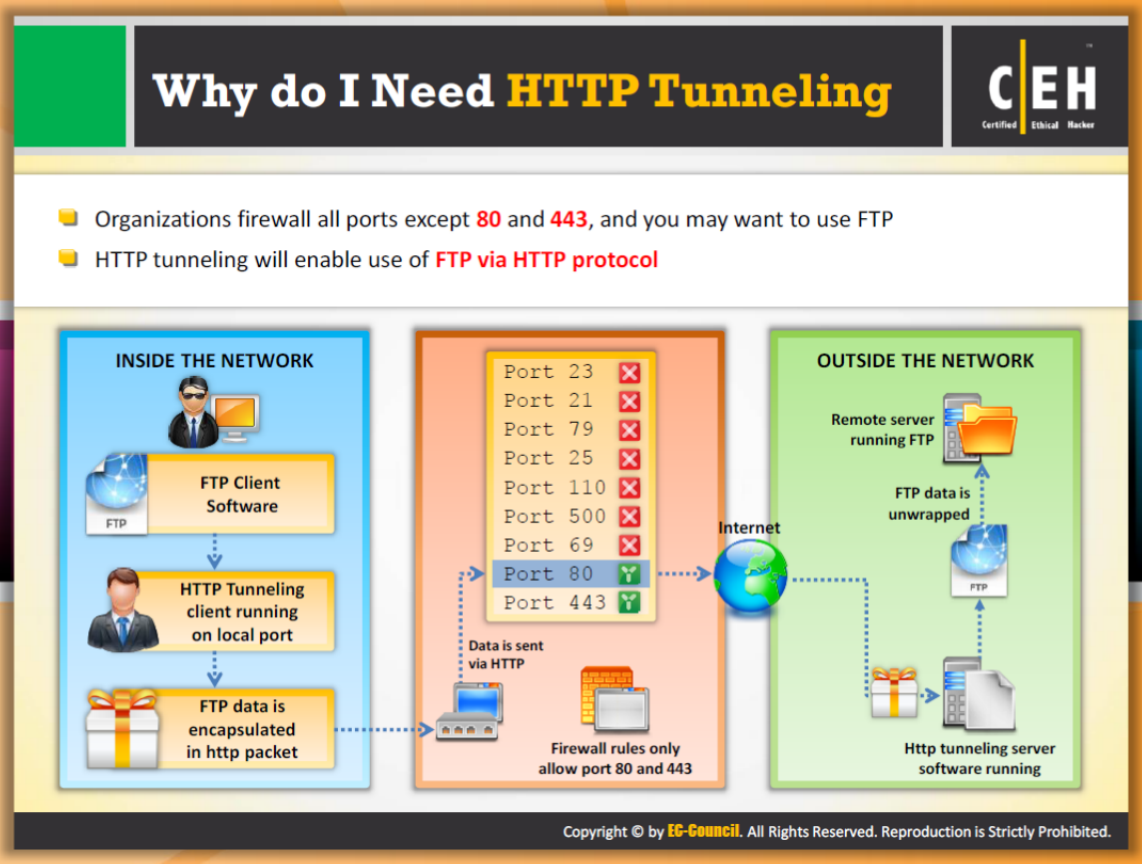


FIGURE 3.68: HTTP Tunneling Process

The HTTP tunneling technique is used in network activities such as:

- Streaming video and audio
- Remote procedure calls for network management
- For intrusion detection alerts
- Firewalls



Why do I Need HTTP Tunneling?

HTTP tunneling allows you to use the Internet despite having **firewall restrictions** such as **blocking specific firewall ports** to restrict specific protocol communication. HTTP tunneling helps you to overcome this firewall restriction by sending specific protocol communication through HTTP protocol.

The attacker may use this technique for the following reasons:

- It assures the attacker that no one will monitor him or her while browsing
- It helps the attacker to bypass firewall restrictions
- It ensures secure browsing
- The attacker can hide his or her IP address from being trapped
- It assures that it is highly impossible for others to identify him or her online

Suppose the organization has blocked all ports in your firewall and only allows port **80/443**, and you want to use FTP to connect to some remote server on the Internet. In this case, you can send your packets via HTTP protocol as shown in the following figure:

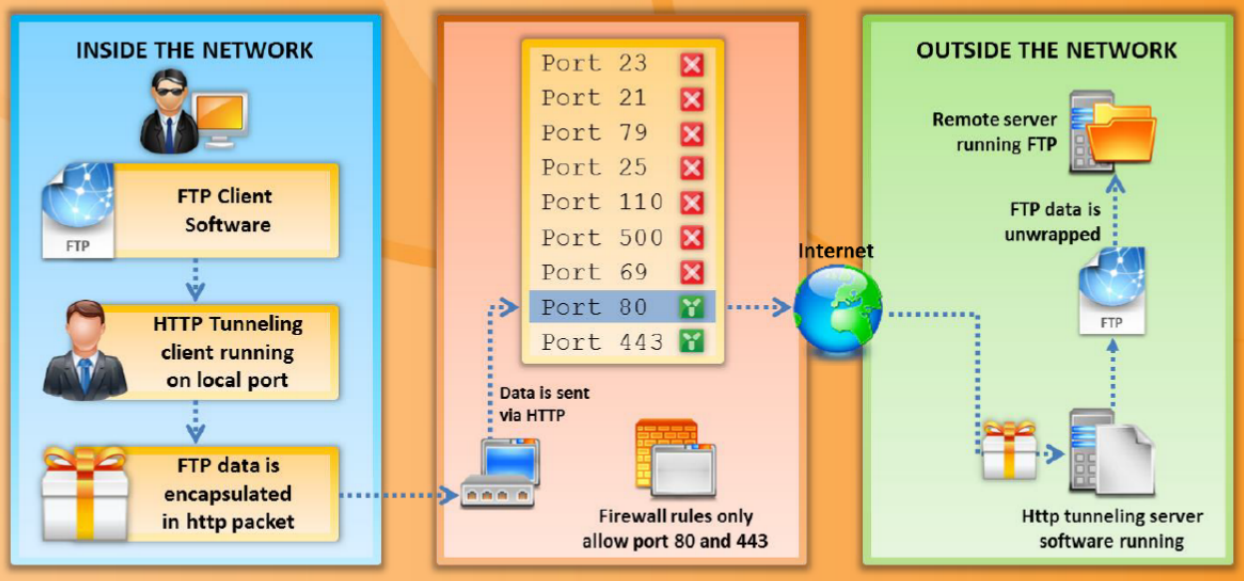
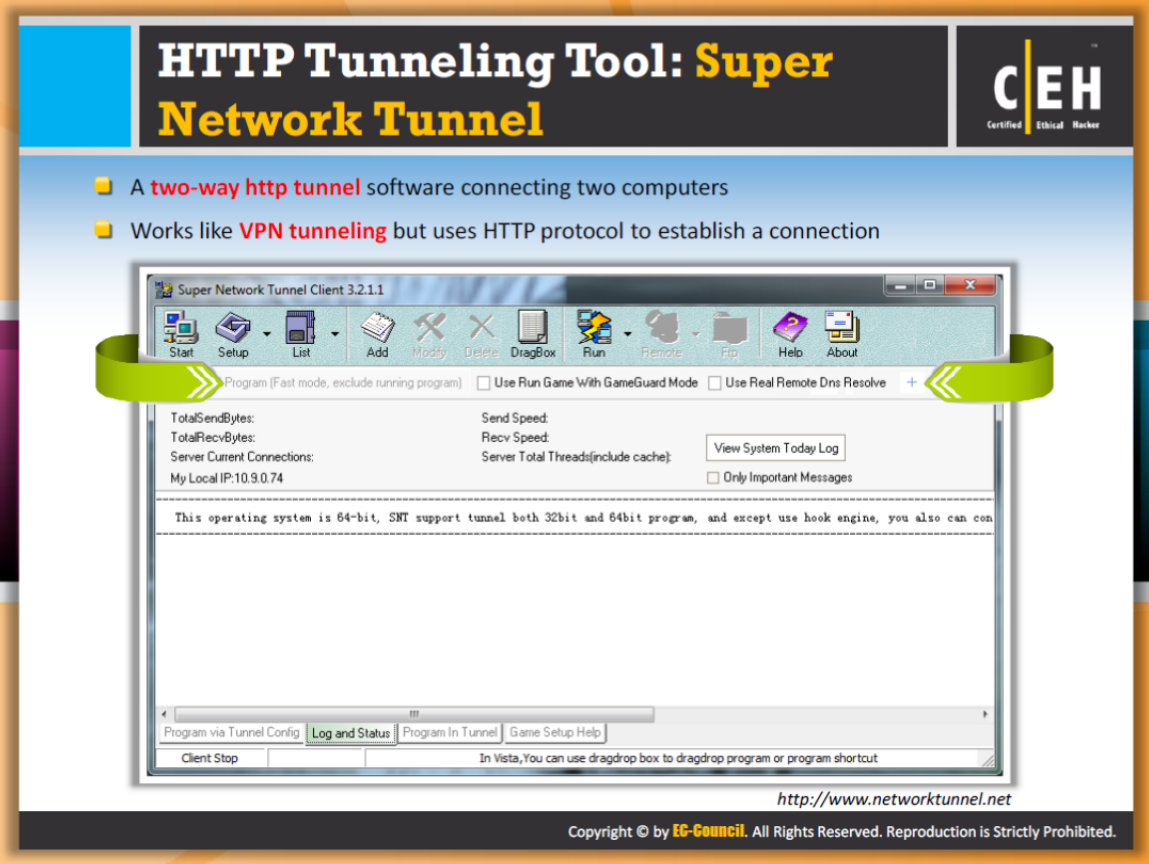


FIGURE 3.69: Effect of HTTP Tunneling on both Inside and Outside the Network



HTTP Tunneling Tool: Super Network Tunnel

Source: http://www.networktunnel.net

Super Network Tunnel is a professional HTTP tunneling software, which includes **HTTP tunnel client** and **server software**. It is like secure VPN software that allows you to access your Internet programs without being monitored by your work, school, or the government, and gives you an extra layer of protection against hackers, spyware, or ID theft. It can bypass any firewall.

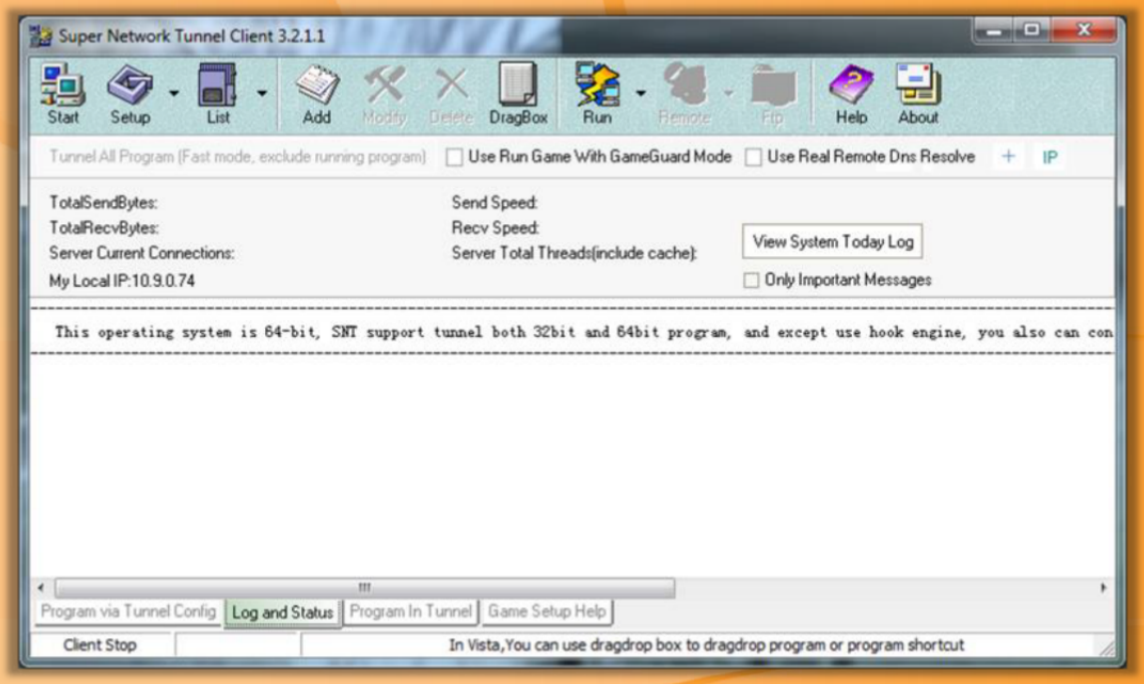
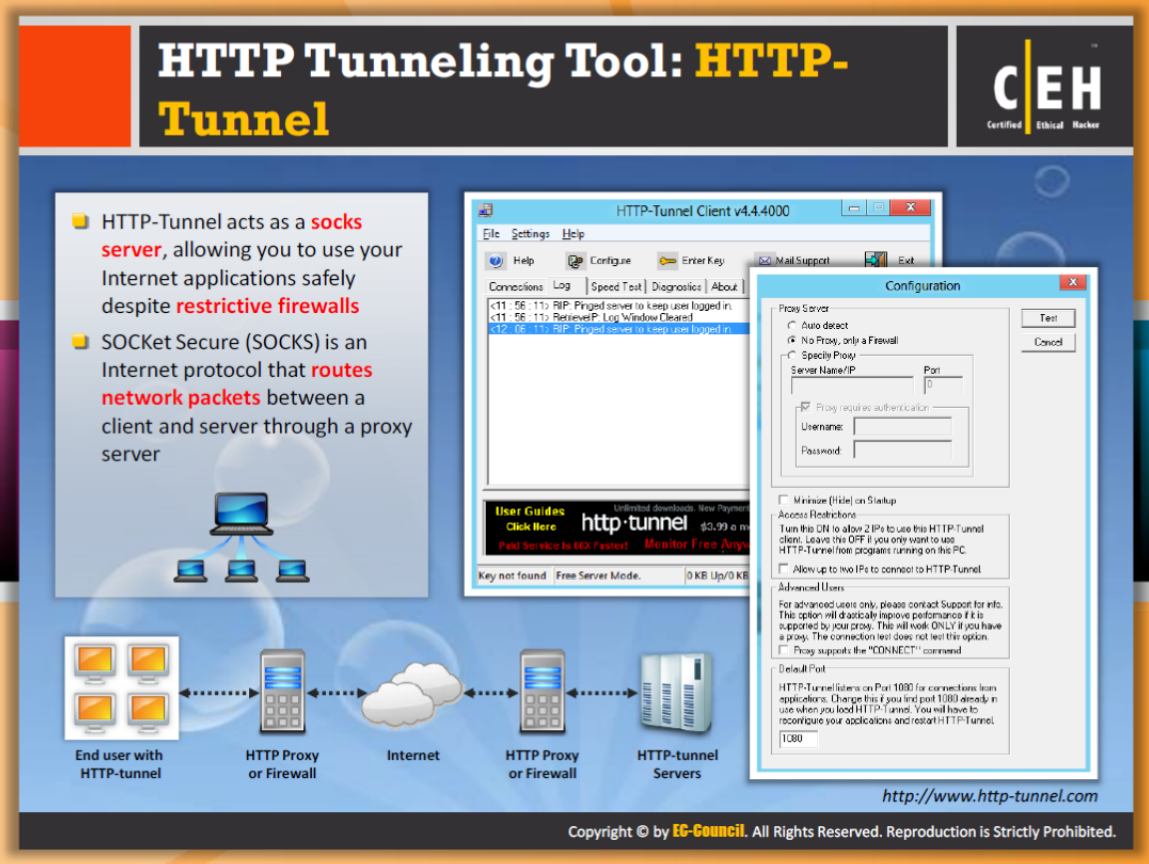


FIGURE 3.70: Super Network Tunnel Screenshot



HTTP Tunneling Tool: HTTP-Tunnel

Source: http://www.http-tunnel.com

HTTP Tunnel acts as a **SOCKS server,** allowing you to access the Internet by **bypassing firewall** restrictions. It is very secure software. Using this software does not allow others to monitor your Internet activities. It hides your IP address; therefore, it does not allow tracing of your system. It allows you the unlimited transfer of data. It runs in your system tray acting as a SOCKS server, managing all data transmissions between the computer and the network.

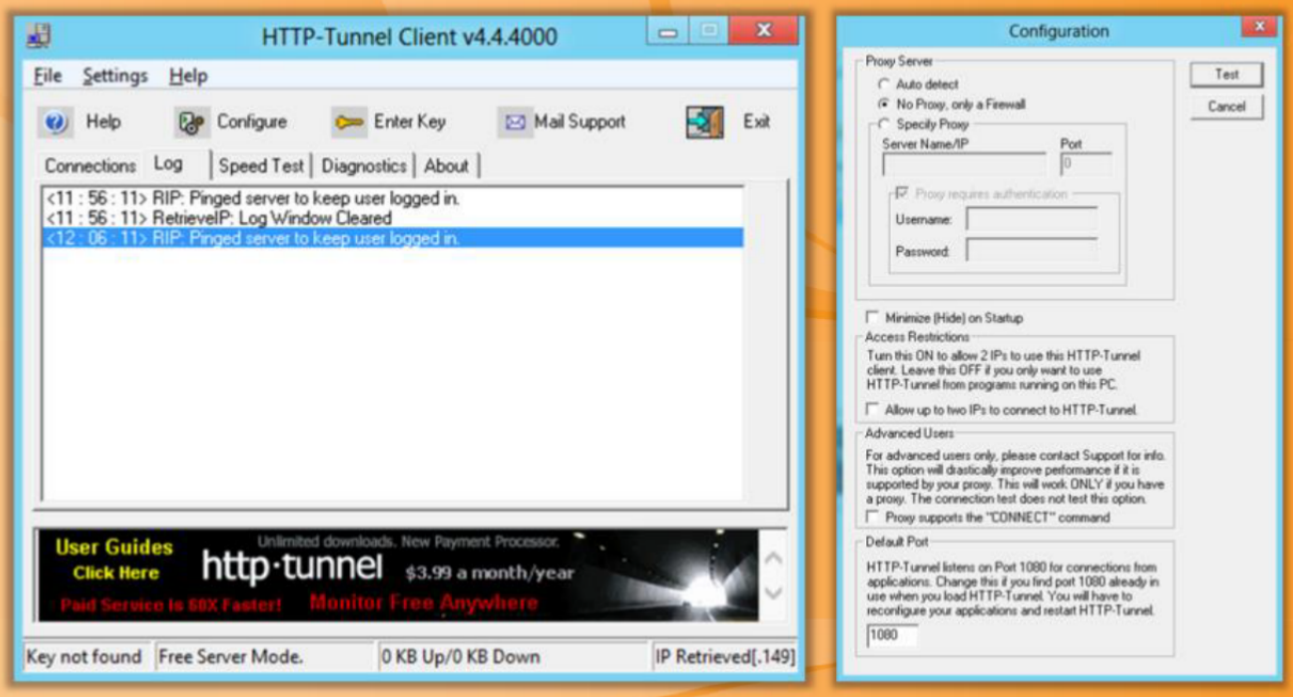
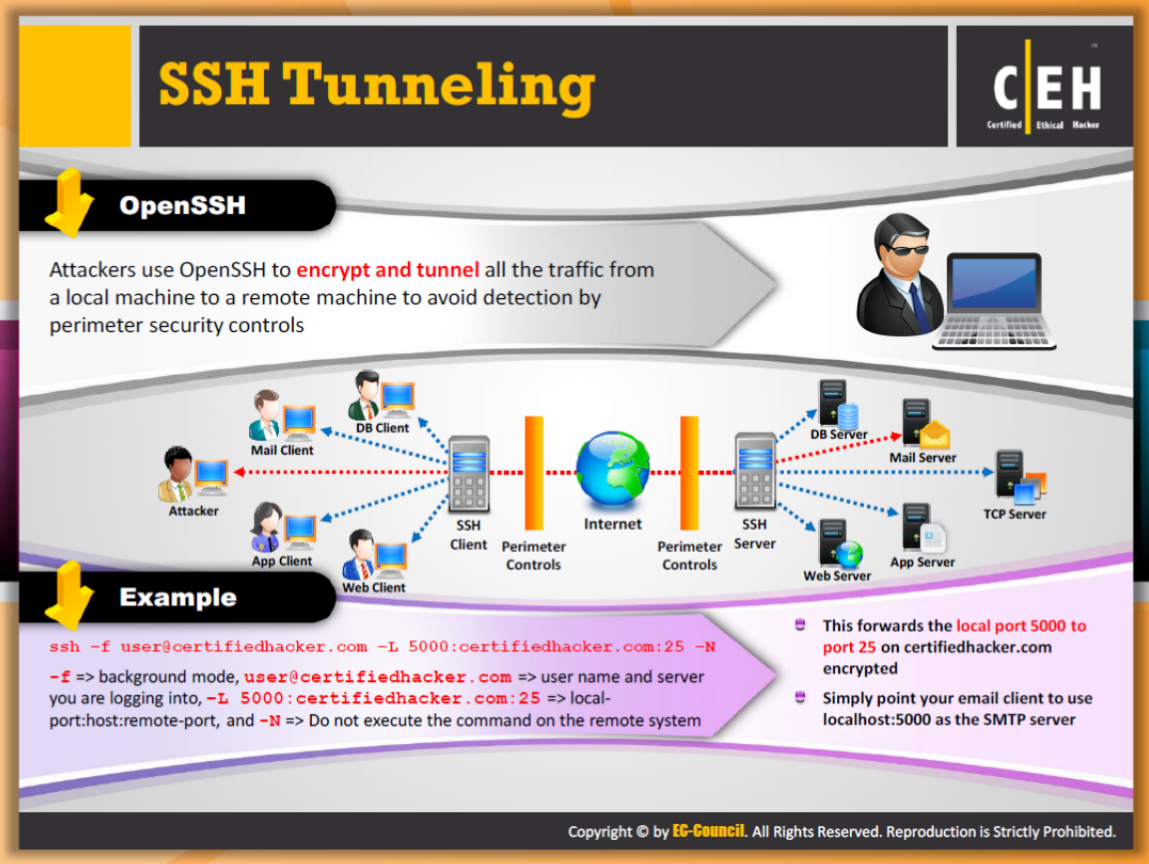


FIGURE 3.71: HTTP-Tunnel Client and Configuration Windows



SSH Tunneling

SSH tunneling is another technique that an attacker can use to **bypass firewall restrictions**. It also helps you hide your IP address on the Internet; therefore, no one can trace or monitor you.

The prerequisite of SSH tunneling is raised from the problems caused by the **public IP address**, the means for accessing computers from anywhere in the world. The computers networked with the public IP address are universally accessible, so they could be attacked by anyone on the global Internet easily and can be **victimized by attackers**. The development of SSH tunneling solves the problems faced by the public IP address.

An SSH tunnel is a link that proceeds traffic from an indiscriminate port on one machine to a remote machine through an intermediate machine. An SSH tunnel comprises an encrypted tunnel, so all your data is encrypted as it uses a secure shell to create the tunnel.

Creating a tunnel for a privately addressed machine needs to implement three basic steps and also requires three machines. The three requisite machines are:

- Local machine
- An intermediate machine with a public IP address
- Target machine with a private address to which the connection must be established

You can create a tunnel as follows:

- Start an SSH connection from local machine to the intermediate machine with public IP address.
- Instruct the SSH connection to wait and observe traffic on the local port, and use intermediate machine to send the traffic to an explicit port on the target machine with a private address. This is called port acceleration or port forwarding.
- On the local machine, select the application that you want to use for connection with the remote machine and configure it to use port forwarding on the local machine. Now, when you connect to the local port, it will redirect the traffic to the remote machine.

To secure communication between computers, SSH uses private and public encryption keys. The public encryption keys used by the SSH tunneling deed like the identifiers of the authorized computer. On initiating an SSH connection, each machine exchanges public keys, but only the computer that has the matching private key can attain access to the remote computer applications and information and can read encrypted communications with the public key.

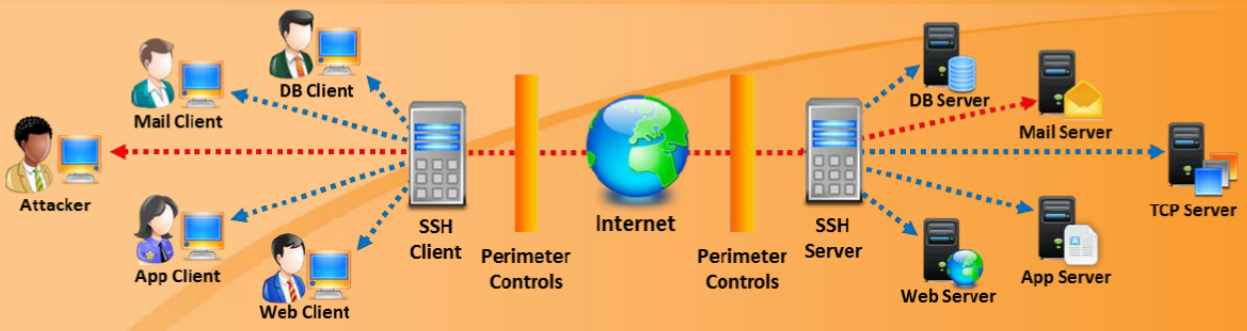


FIGURE 3.72: SSH Tunneling Process

OpenSSH

Source: http://www.openssh.org

OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other attacks. Additionally, OpenSSH provides secure tunneling capabilities and several authentication methods, and supports all SSH protocol versions. OpenSSH can be used to tunnel the traffic on local machine to a remote machine that you have an account on.

```
ssh -f user@certifiedhacker.com -L 2000:certifiedhacker.com:25 -N
```

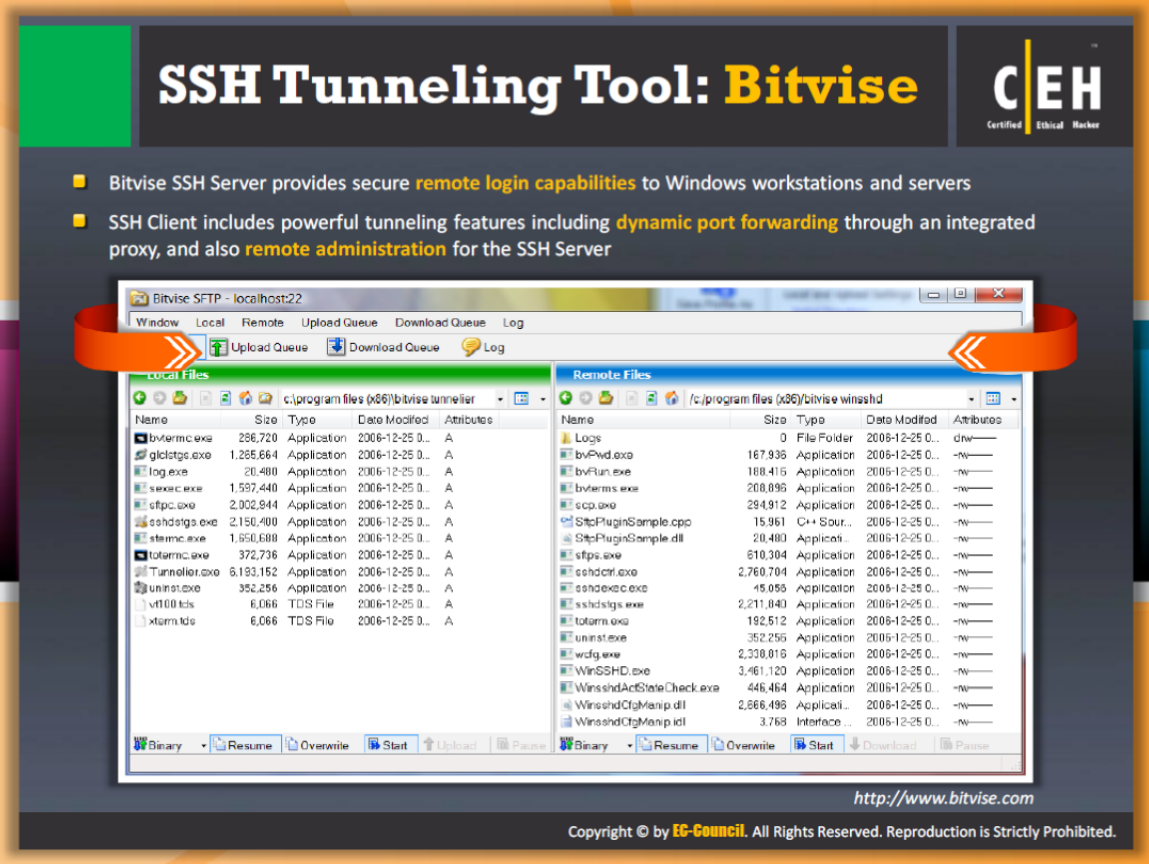
`-f` => backgroung mode

`user@certifiedhacker.com` => user name and server you are logging into

`-L 2000:certifiedhacker.com:25` => local-port:host:remote-port

`-N` => Do not execute the command on the remote system

This essentially forwards the local port 2000 to port 25 on certifiedhacker.com encrypted. Simply point your email client to use localhost:2000 as the SMTP server.



SSH Tunneling Tool: Bitvise

Source: http://www.bitvise.com

Bitvise is client **server-based application** used for SSH tunneling. The server provides you secure remote login capabilities to Windows workstations and servers. With Bitvise SSH Server, you can administer the **Windows server remotely**. The Bitvise server even has the ability to encrypt the data during transmission so that no one can sniff your data during transmission.

Bitvise SSH Client includes graphical as well as command line SFTP support, an FTP-to-SFTP bridge, tunneling features that can be helpful for port forwarding and remote administration.

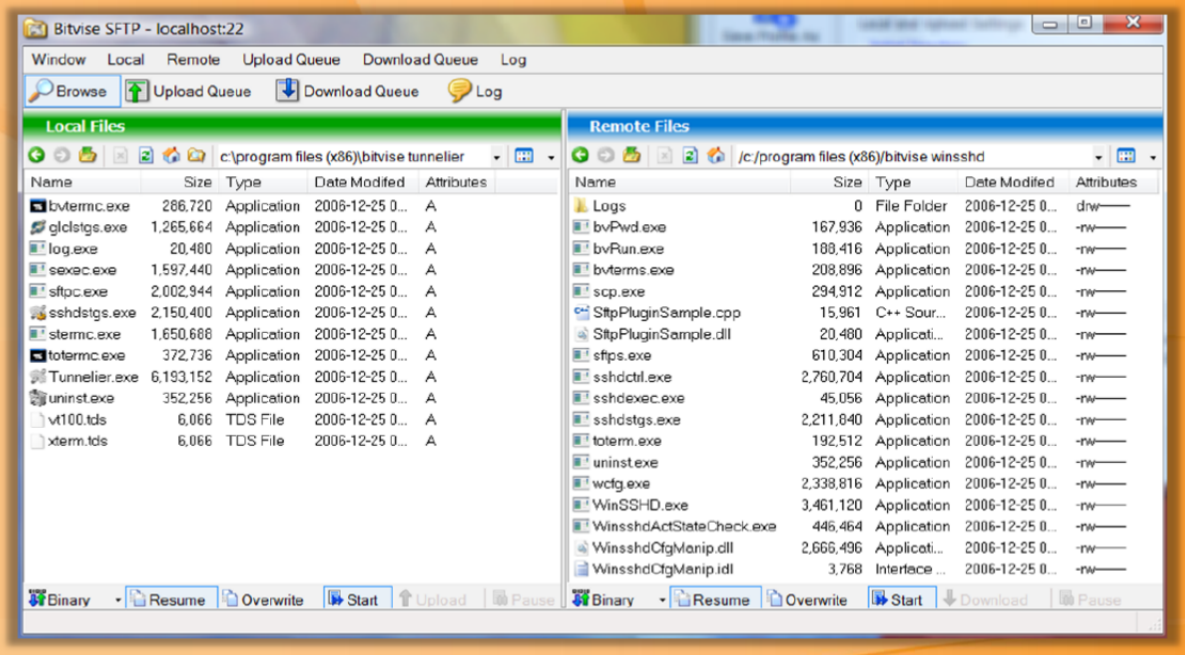


FIGURE 3.73: Bitvise Tool Screenshot

Anonymizers

An anonymizer is an **intermediate server** placed in between the **end user** and **web site** that accesses the website on behalf of you, making your web surfing untraceable. 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, so that every subsequent web access will be anonymized. Apart from this, you can choose to anonymously provide passwords and other information to sites that request you, without revealing any other 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:

- **Ensures privacy**: It protects your identity by making your **web navigation** activities **untraceable**. Your privacy is maintained until and unless you disclose your personal information on the web by filling out forms, etc.
- **Accesses government-restricted content**: Most governments prevent their citizens from accessing certain websites or content in order to avoid them from accessing inappropriate information or sensitive information. But these people can access even these types of resources by an anonymizer located outside the country.
- **Protect you from online attacks**: Anonymizers protect you from all instances of online pharming attacks by routing all customer Internet traffic via the anonymizer's protected **DNS servers**.
- **Bypass IDS and firewall rules**: Bypassing of firewalls is mostly done in organizations or schools by employees or students accessing websites 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 doing such, firewalls can see only the connection from you to anonymizer's web address. The anonymizer will then connect to Twitter or any website you wanted to access with the help of an Internet connection and sends the content back to you. For your organization, it looks like your system is connected to an anonymizer's web address, but not to Twitter or other sites.

Anonymizers, apart from protecting users' identities, can also attack the website and no one can actually detect where the attack came from.

Types of Anonymizers

An anonymizer is a service through which one can hide their identity when using certain services of the Internet. It basically works by encrypting the data from your computer, so that is cannot be understood by Internet service providers or anyone who might try to access it. Basically, anonymizers are of two types:

- Networked anonymizers
- Single-point anonymizers

Networked Anonymizers

These type of anonymizer first transfers your information through a network of Internet computers before sending it to the website. Since 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 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 prior to going to the website. Then after being opened, the page will be transferred back through C, B, and A and then to you.

Advantage: Complication of the communications makes **traffic analysis complex**

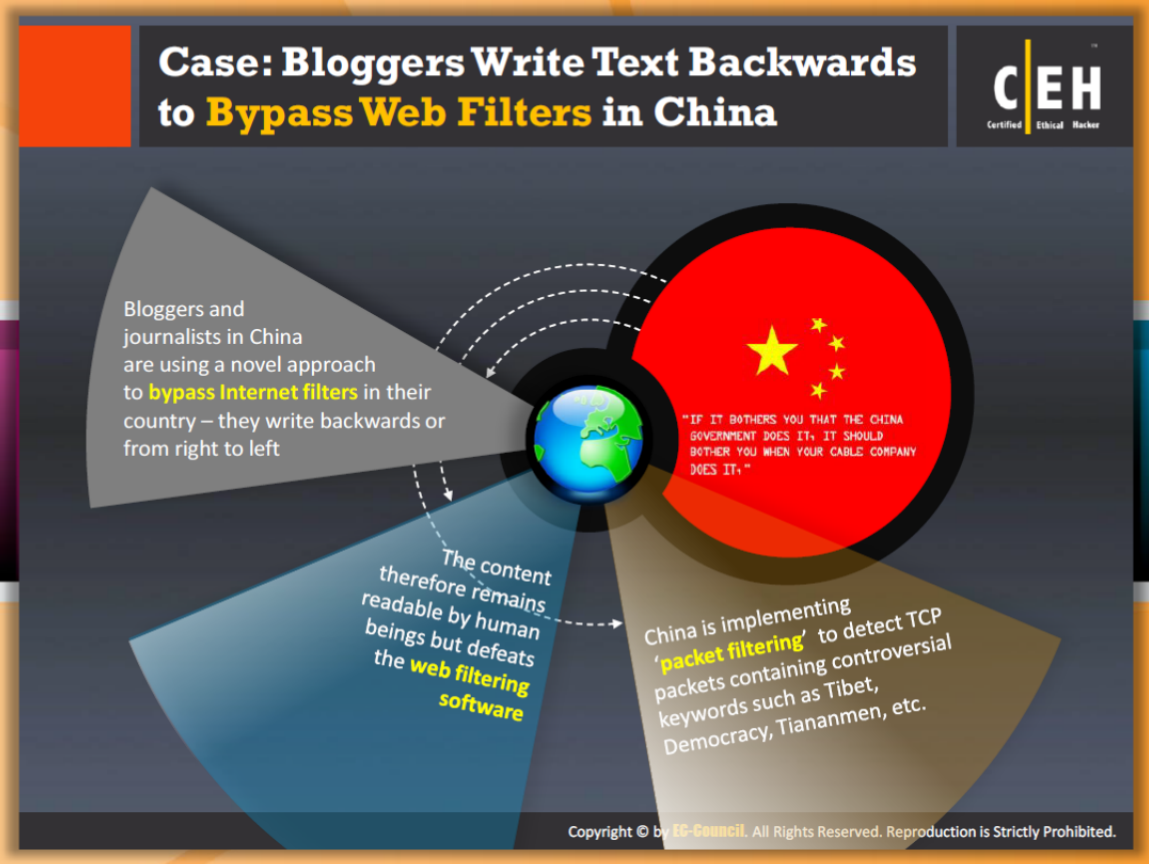
Disadvantage: Any multi-node network communications have some degree of risk at each node for compromising confidentiality

Single-point Anonymizers

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

Advantage: IP address and related identifying information are protected by the arms-length communications

Disadvantage: It offers less resistance to sophisticated traffic analysis



Case: Bloggers Write Text Backwards to Bypass Web Filters in China

China is well known for its implementation of the "**packet filtering**" technique. This technique **detects TCP packets** that contain controversial keywords such as Tibet, Democracy, Tiananmen, etc. To bypass Internet filters and dodge the censors, bloggers and journalists in China are writing the text backwards or from right to left. By doing so, though the content is still in human readable form, the text is successful in defeating web filtering software. Bloggers and journalists use vertical text converter tools to write the text backwards or from right to left and vertically instead of horizontally.

Censorship Circumvention Tool: Psiphon

Source: http://psiphon.ca

Psiphon is a censorship circumvention system that allows users to **bypass firewalls** and access **blocked sites in countries** where the Internet is censored. It uses a secure, encrypted HTTP tunnel connection to receive requests from psiphonite to psiphonode, which in turn then transports the results back to the requested psophonite. It acts as a web proxy for authenticated psiphonites, and works on mobile browsers. It bypasses content-filtering systems of countries such as China, North Korea, Iran, Saudi Arabia, Egypt, and others.

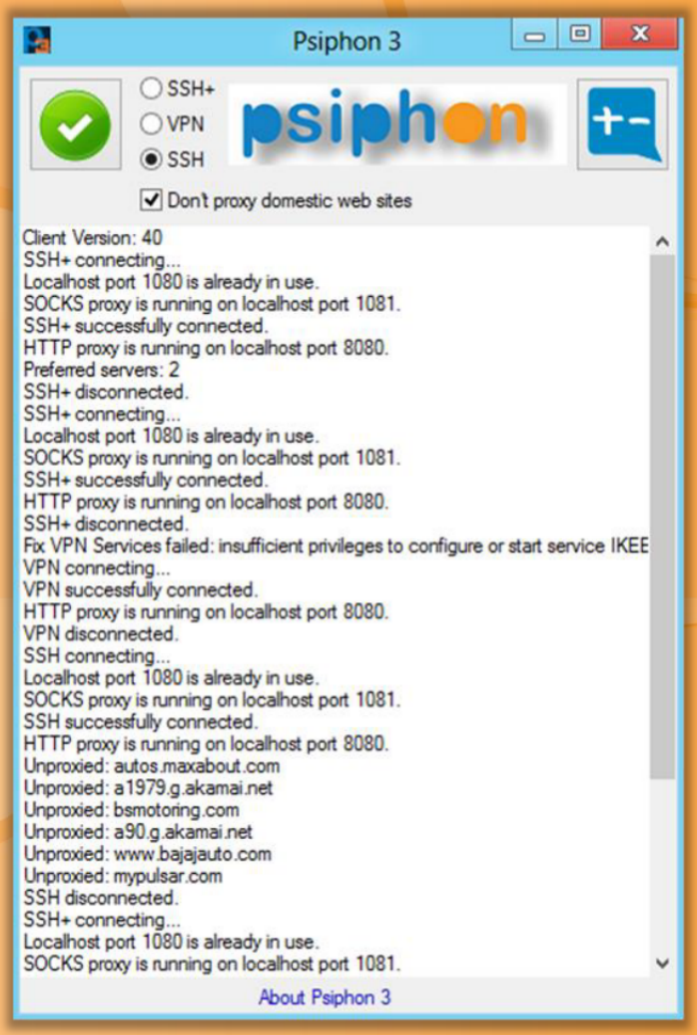


FIGURE 3.74: Psiphon Screenshot

Censorship Circumvention Tool: Your-Freedom

Source: http://www.your-freedom.net

Censorship circumvention tools allow you to **access websites** that are not accessible to you by **bypassing firewalls**. The Your Freedom services makes accessible what is unaccessible to you, and they hide your network address from those who don't need to know. This tool turns your PC into an uncensored, anonymous web proxy and an uncensored, anonymous **SOCKS proxy** that your applications can use, and if that's not enough, it can even get you connected to the Internet just as if you were using an unrestricted DSL or cable connection.

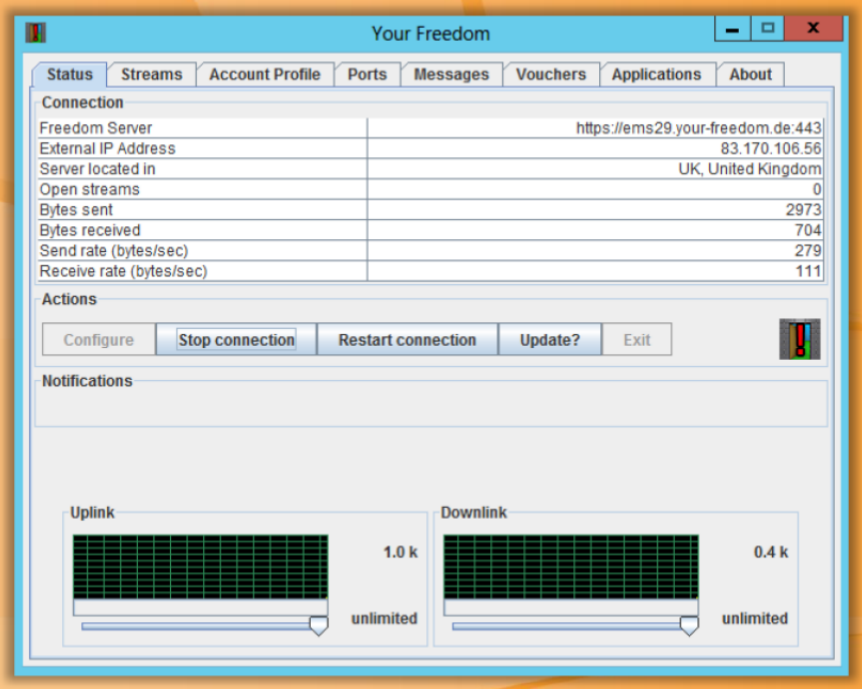


FIGURE 3.75: Your-Freedom Screenshot

How to Check if Your Website is Blocked in China or Not?

If a "**packets lost**" error is received or there is a connection **time-out message** is displayed while connecting to your site, chances are that the site is blocked. To find out whether the website at xyz.com is accessible by Chinese web users, you can use tools such as just ping and WebSitePulse.

Just ping

Source: http://www.just-ping.com

Just ping is an online web-based ping tool that allows you to ping from various locations worldwide. It pings a website or IP address and displays the result as shown as follows:

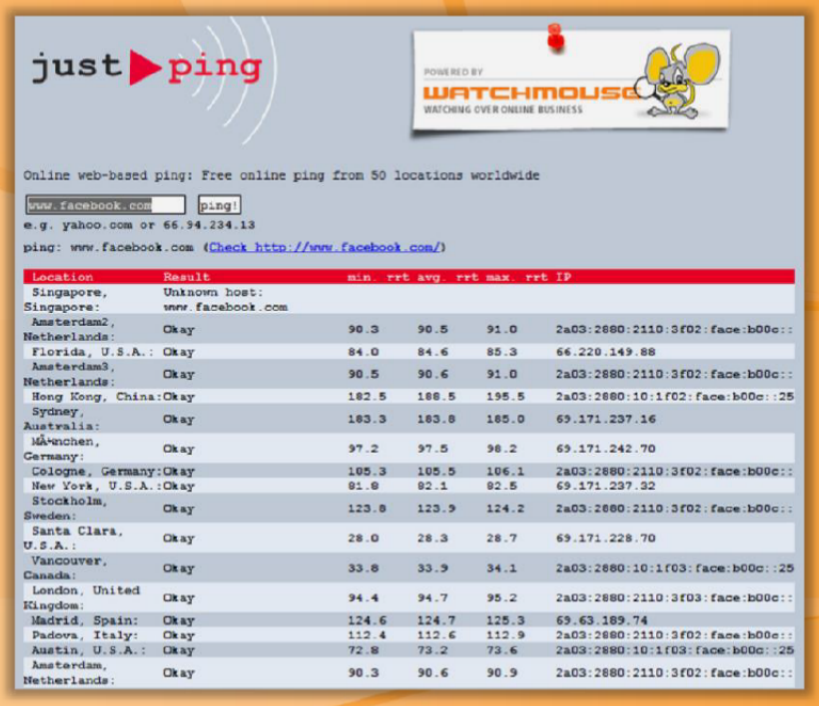


FIGURE 3.76: Just Ping Screenshot

WebsitePulse

Source: http://www.websitepulse.com

WebsitePulse provides remote monitoring services. It simultaneously pops websites from around the globe.

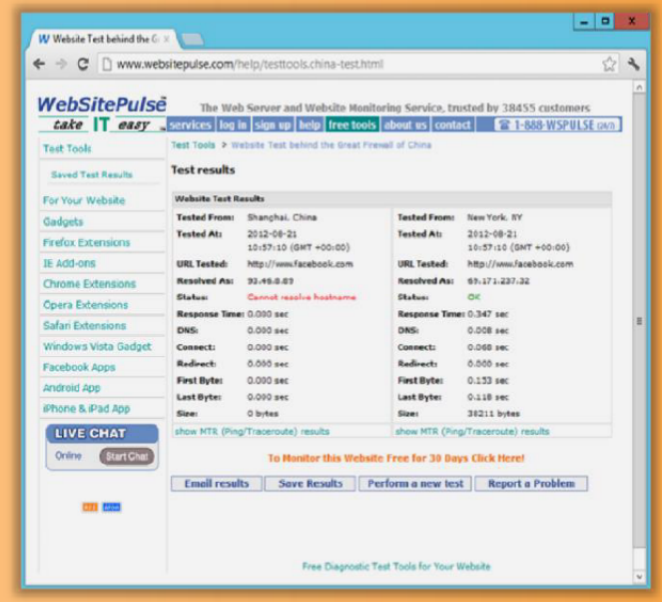
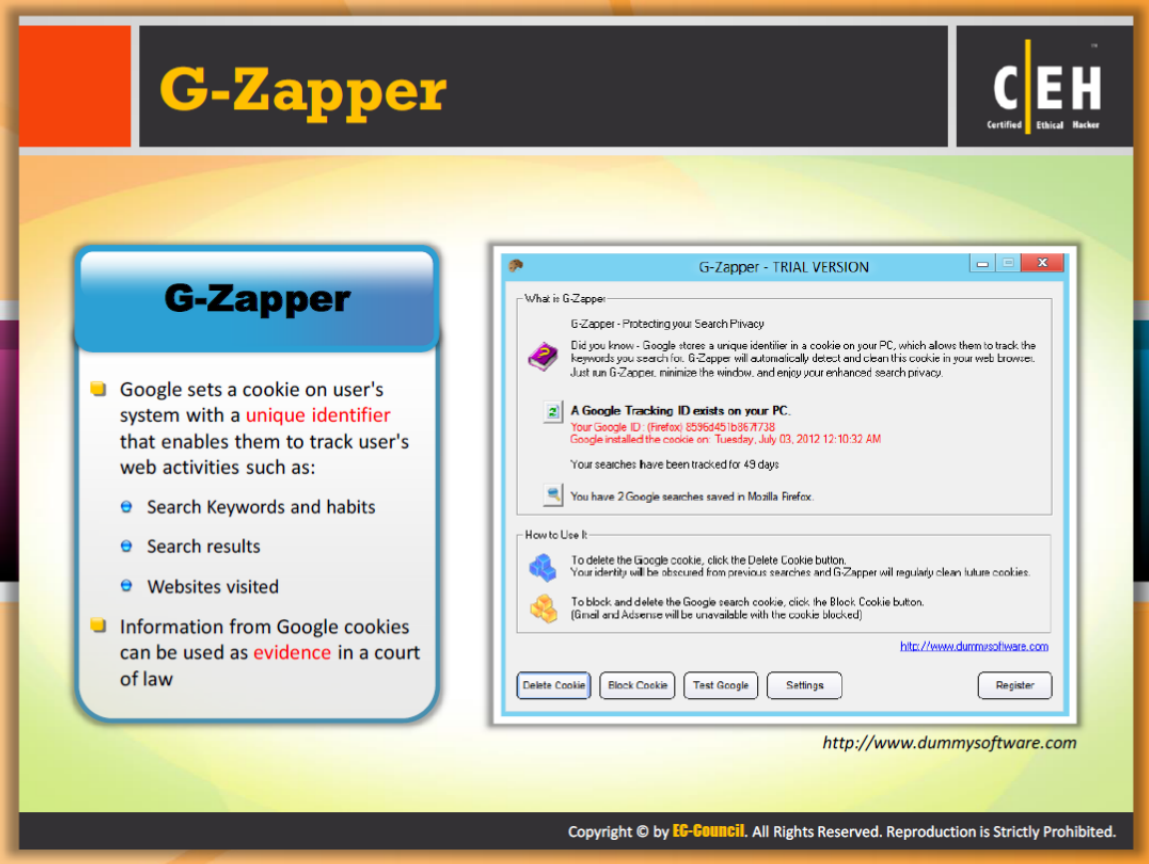


FIGURE 3.77: WebsitePulse Screenshot



G-Zapper

Source: http://www.dummysoftware.com

G-Zapper is a utility to **block Google cookies**, **clean Google cookies**, and help you stay anonymous while searching online. It automatically detects and cleans Google cookies each time you use your web browser.

It is compatible with **Windows 95/98/ME/NT/2000/XP/Vista/Windows7**. It requires Microsoft Internet Explorer, Mozilla Firefox, or Google Chrome and is compatible with Gmail, Adsense, and other Google services.

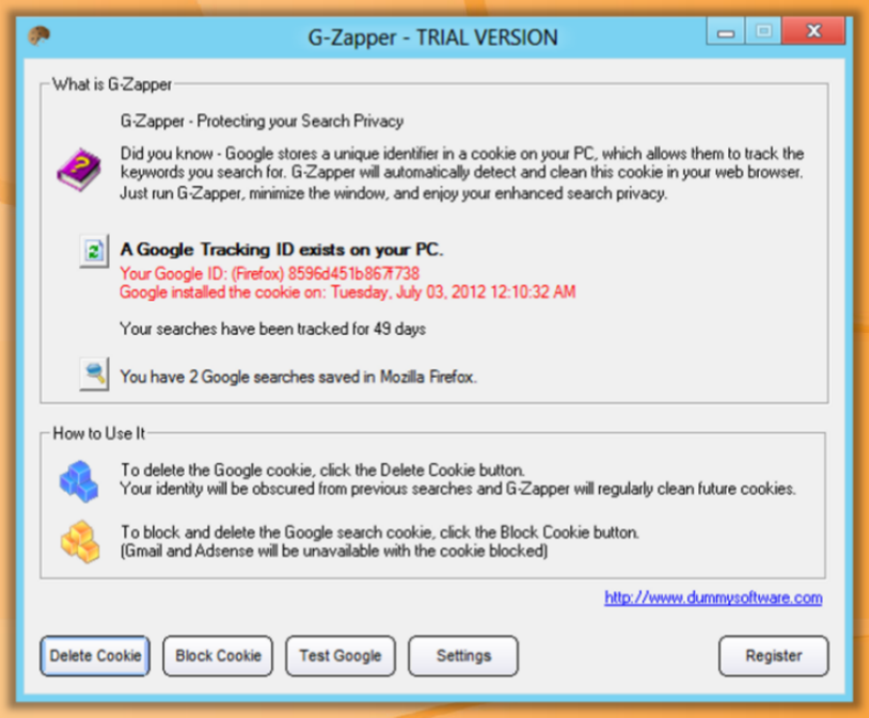


FIGURE 3.78: G-Zapper-Trial Version Screenshot

Anonymizers

An anonymizer is a tool that allows you to mask your IP address to visit websites without being tracked or identified, keeping your activity private. It allows you to access blocked content on the Internet with omitted advertisements. A few anonymizers that are readily available in the market are listed as follows:

- Mowser available at http://www.mowser.com
- Anonymous Web Surfing Tool available at http://www.anonymous-surfing.com
- Hide Your IP Address available at http://www.hideyouripaddress.net
- Anonymizer Universal available at http://www.anonymizer.com
- Guardster available at http://www.guardster.com
- Spotflux available at http://www.spotflux.com
- U-Surf available at http://ultimate-anonymity.com
- WarpProxy available at http://silent-surf.com
- Hope Proxy available at http://www.hopeproxy.com
- Hide My IP available at http://www.privacy-pro.com

Spoofing IP Addresses

Spoofing IP addresses enables attacks like hijacking. When spoofing, an attacker a fake IP in place of the attacker's assigned IP. When the attacker sends a connection request to the target host, the target host replys to the attacker's request. But the reply is sent to the spoofed address. When spoofing an address that doesn't exist, the target replies to a non-existent system and then hangs until the session times out, consuming target resources.

IP spoofing using Hping2:

```
Hping2 www.cretifiedhacker.com -a 7.7.7.7
```

Using Hping2 you can perform IP spoofing. It helps you to send arbitrary **TCP/IP** packets to network hosts.

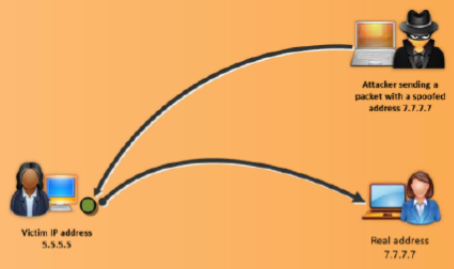
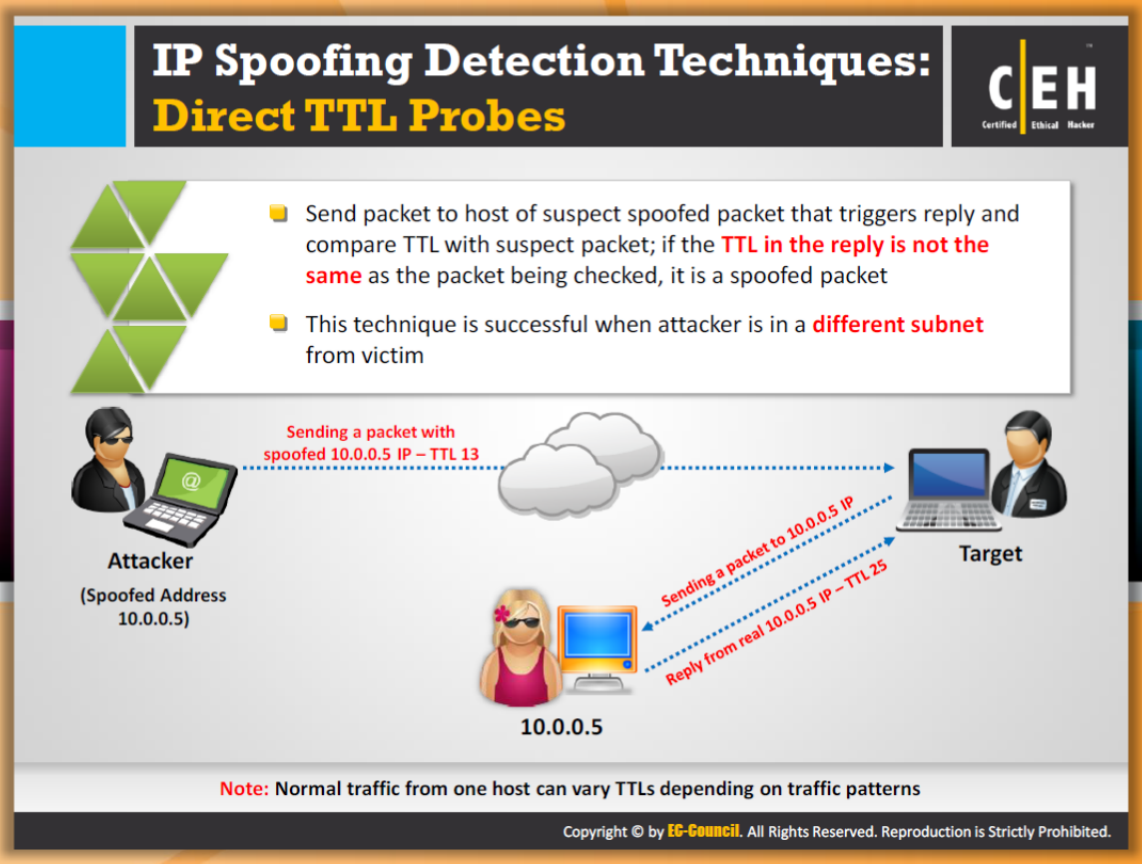


FIGURE 3.79: Attacker Sending Spoofed Packet to The Victim



IP Spoofing Detection Techniques: Direct TTL Probes

Initially send a packet to the host of suspect spoofed packet and wait for the reply. Check whether the TTL value in the reply matches with the TTL value of the packet that you are checking. Both will have the same TTL if they are the same protocol. Though, initial **TTL values vary** based on the protocol used, a few initial TTL values are commonly used. For TCP/UDP, the commonly used initial values are **64** and **128** and for ICMP, the values are **128** and **255**. If the reply is from a different protocol, then you should check the actual hop count to detect the spoofed packets. The hop count can be determined by deducting the TTL value in the reply from the initial TTL value. If the TTL in the reply is not matching with the TTL of the packet that you are checking, it is a spoofed packet. If the attacker knows the hop count between source and host, it will be very easy for the attacker to launch an attack. In this case, the test results in a **false negative**.

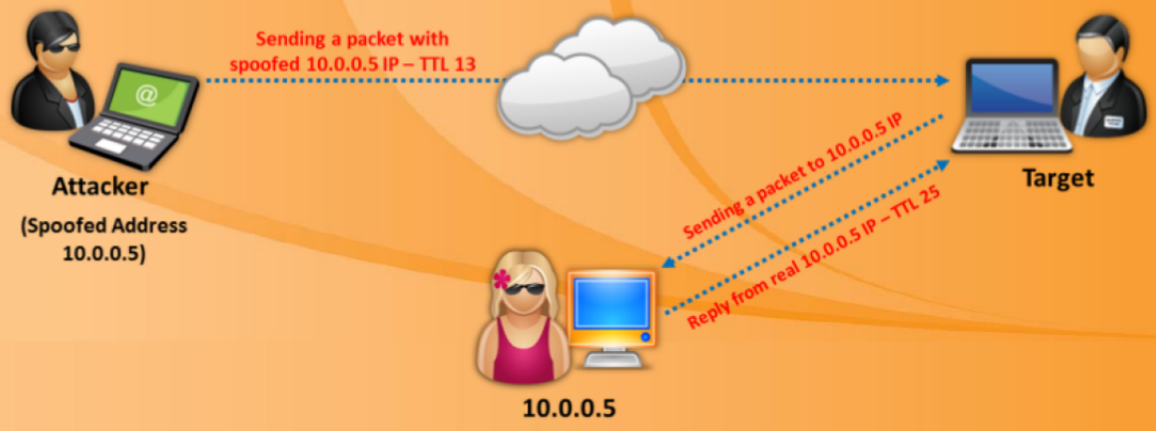
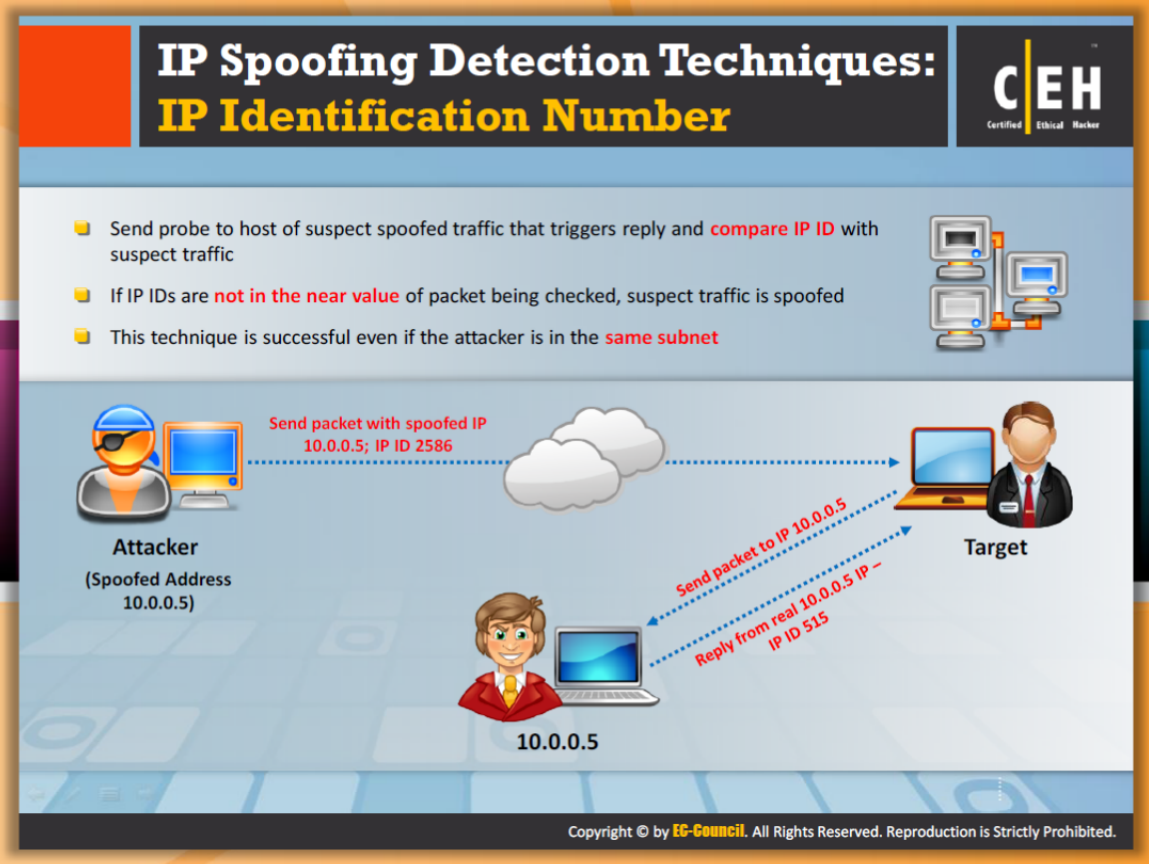


FIGURE 3.80: Using Direct TTL Probes for IP Spoofing Detection



IP Spoofing Detection Techniques: IP Identification Number

Spoofed packets can be identified based on the identification number (IP ID) in the IP header that increases each time a packet is sent. This method is effective even when both the attacker and victim are on same subnet.

To identify whether the packet is spoofed or not, send a probe packet to the target and observe the IP ID number in the reply. If it is in the near value as the packet that you are checking, then it is not a spoofed packet, otherwise it is a spoofed packet.

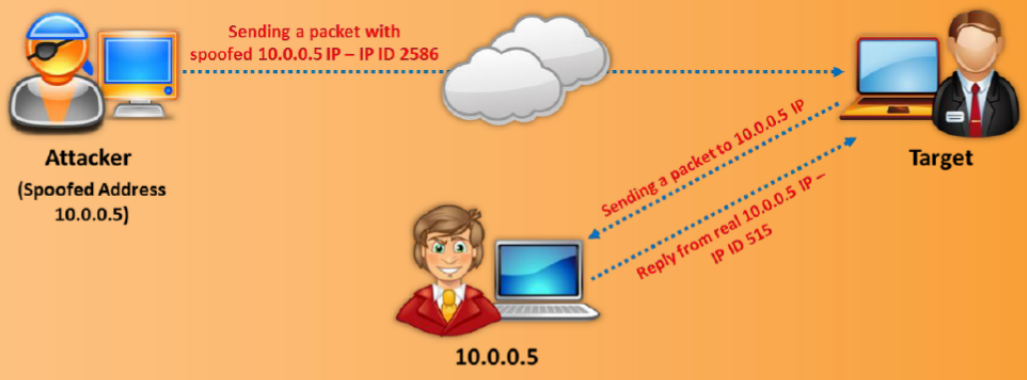
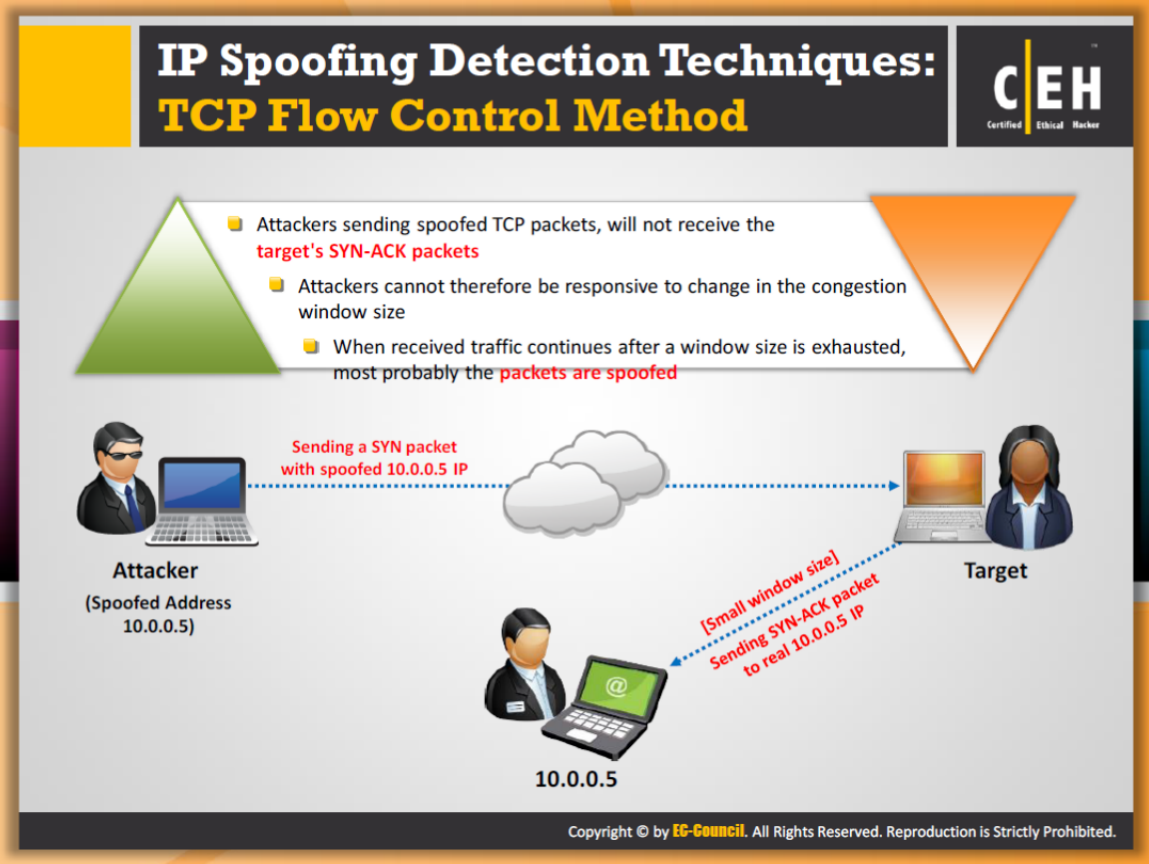


FIGURE 3.81: Using IP Identification Number for IP Spoofing Detection



IP Spoofing Detection Techniques: TCP Flow Control Method

The TCP can optimize the **flow control** on both the **send** and the **receiver side** with its algorithm. The algorithm accomplishes the flow control based on the sliding window principle. The flow of IP packets can be **controlled 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 the sender can transmit without acknowledgement. Thus, this field helps us to control data flow. When the window size is set to **zero**, the sender should stop sending more data.

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** continues to send data to the victim. If the victim receives data packets beyond the window size, then the packets must be treated as spoofed. 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 difficult to build multiple spoofing replies with the correct sequence number. Therefore, the flow control spoofed packet detection must be applied 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, you should set the SYN-ACK to zero. If the sender sends an **ACK** with any data, then 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 but not ACK with data.

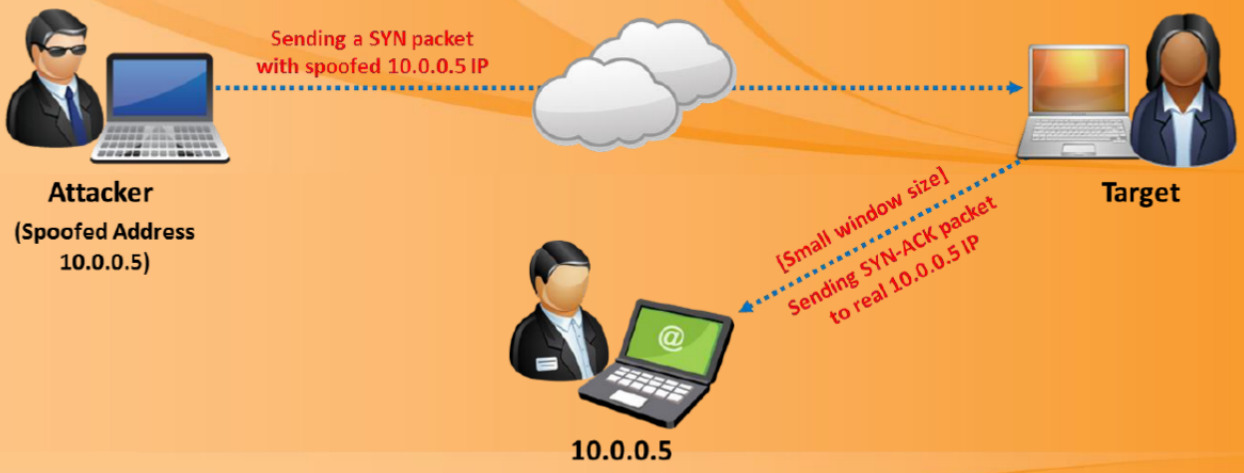


FIGURE 3.82: Using TCP Flow Control Method for IP Spoofing Detection

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 doesn't 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, in order 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

Attackers may spoof themselves as a trusted host and send **malicious packets** to you. If you accept those packets by considering that the packets are sent by your trusted host, then you may get infected. Therefore, it is advisable to test the 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

You should filter all the incoming and outgoing packets to avoid attacks and sensitive information loss. The incoming packets may be the malicious packets coming from the attacker.

If you do not employ any kind of incoming packet filtering mechanism such as a firewall, then the malicious packets may enter your private network and may cause severe loss. You can use access control lists (ACLs) to **block unauthorized access**. At the same time, there is also a possibility of insider attackers. These attackers may send sensitive information about your business to your competitors. This may also lead to great monetary loss or other issues. There is one more risk of outgoing packets, which is when an attacker succeeds 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 giving any notification. This can be figured out by filtering the outgoing packets. Therefore, you should give the same importance to the scanning of outgoing packets as that of the incoming packet data scanning.

Use random initial sequence numbers

Most of the devices chose 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 or she can make a malicious connection to the server and sniff your network traffic. To avoid this, risk you should use random initial sequence numbers.

Ingress filtering

Prohibiting spoofed traffic from entering the Internet is the best way to block it. This can be achieved with the help of ingress filtering. Ingress filtering applied on routers enhances the functionality of the routers and blocks spoofed traffic. It can be implemented in many ways. Configuring and using **access control lists (ACLs)** that drop packets with source address outside the defined range is one way 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

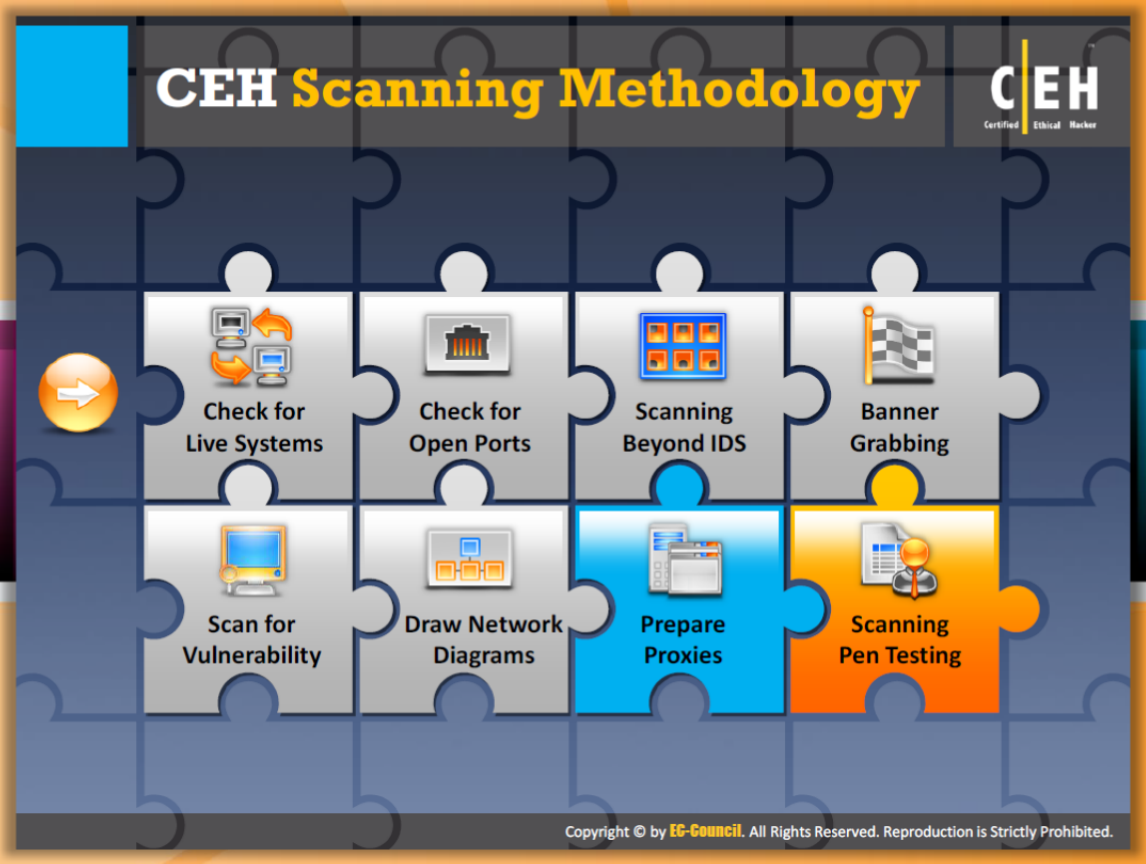
If you want to attain maximum network security, then use **strong encryption** for all the traffic placed onto the transmission media without considering its type and location. This is the best solution for IP spoofing attacks. Attackers usually tend to find targets that can be compromised easily. If an attacker wants to break into encrypted network, then he or she has to face a whole slew of encrypted packets, which is a difficult task. Therefore, the attacker may try to find another target that can be easily compromised or may attempt other techniques to break into the network. 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.

Besides these basic countermeasures, you can perform the following to avoid IP spoofing attacks:

- You should limit the access to configuration information on a machine
- You should always disable commands like **ping**
- You should reduce **TTL fields** in TCP/IP requests
- You should use multilayered firewalls

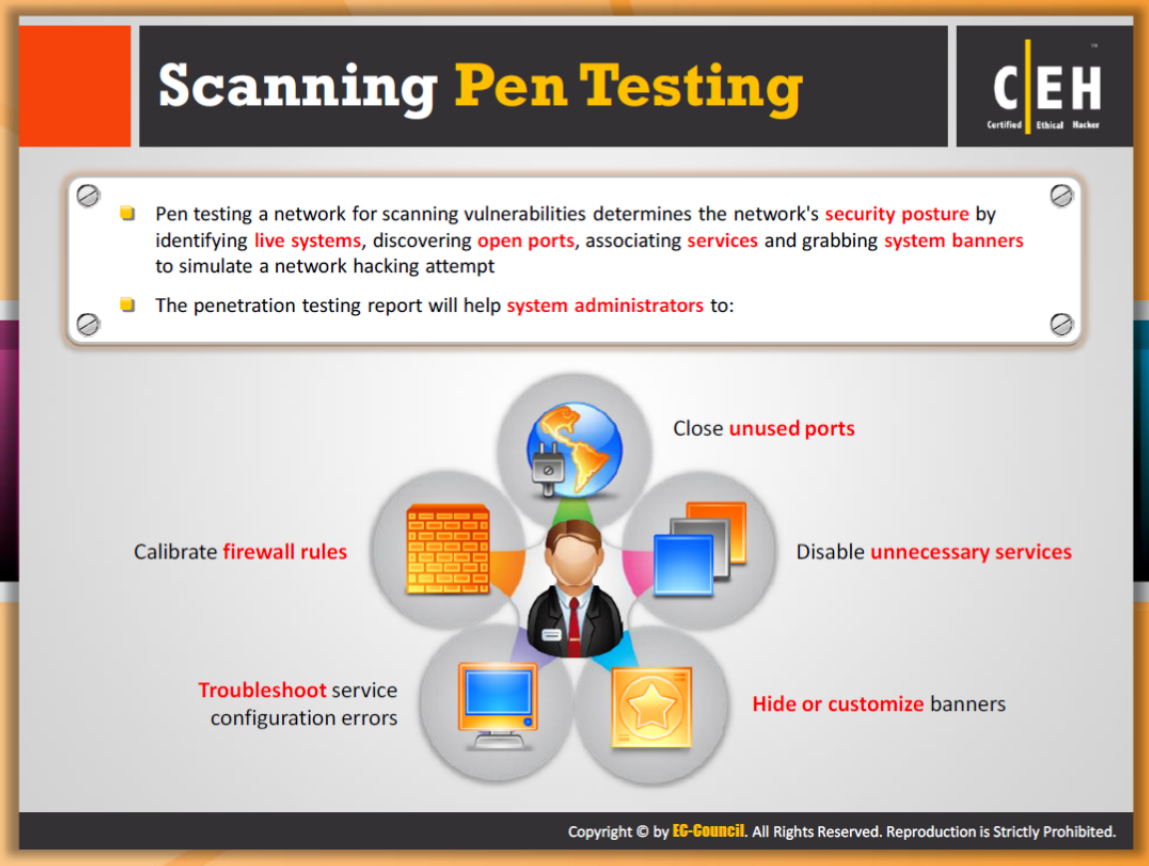


CEH Scanning Methodology

So far, we have discussed concepts such as what to scan, how to scan, how to detect vulnerabilities, and the respective countermeasures that are necessary to perform scanning pen testing. Now we will begin the action of **scanning pen testing**.

Check for Live Systems	Scan for Vulnerability
Check for Open Ports	Draw Network Diagrams
Scanning Beyond IDS	Prepare Proxies
Banner Grabbing	Scanning Pen Testing

This section highlights the need to scan pen testing and the steps to be followed for effective pen testing.

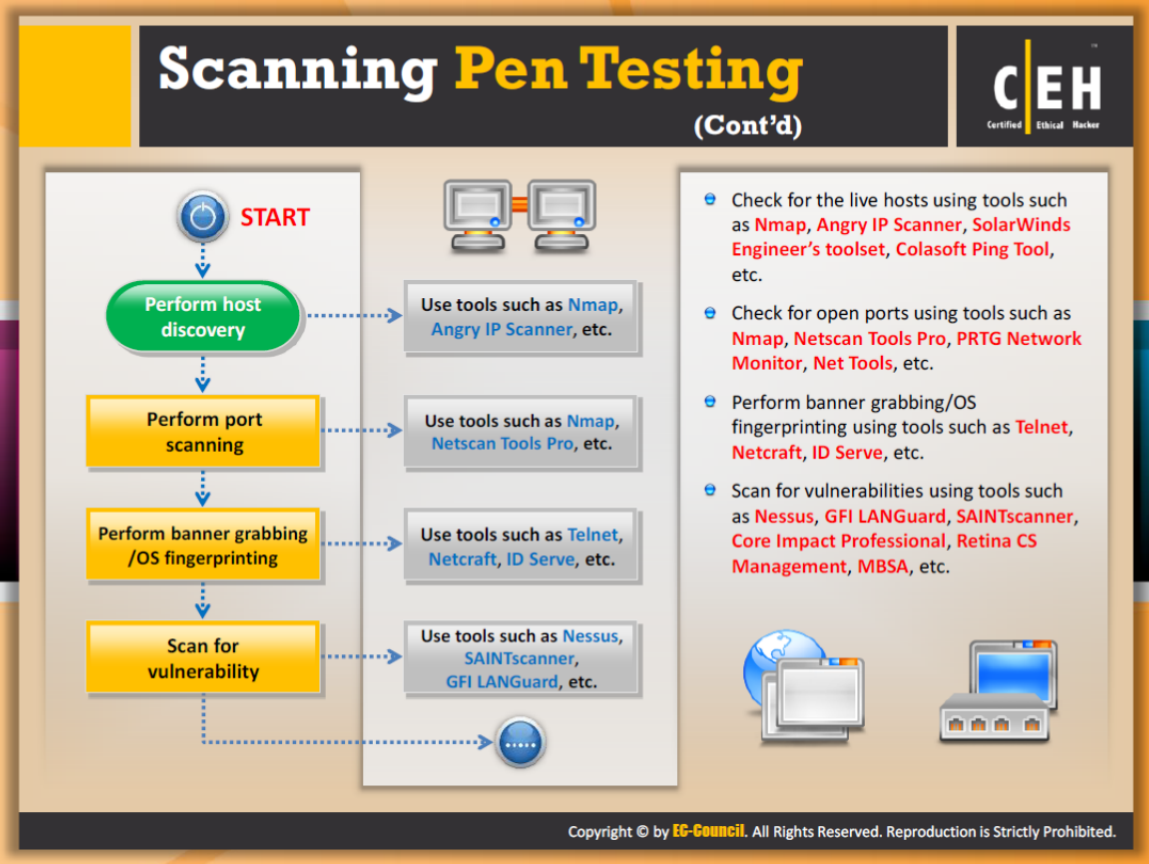


Scanning Pen Testing

The network scanning penetration test helps you to determine the **network security** posture by identifying live systems, discovering open ports and associated services, and grabbing system banners from a remote location, simulating a **network hacking attempt**. You should scan or test the network in all possible ways to ensure that no security loophole is overlooked.

Once you are done with the penetration testing, you should **document** all the **findings** obtained at every stage of testing so that it helps system administrators to:

- **Close unused ports** if not necessary/unknown open ports found
- Disable unnecessary **services**
- **Hide** or customize **banners**
- **Troubleshoot** service configuration **errors**
- **Calibrate firewall rules** to impose more restriction



Scanning Pen Testing (Cont'd)

Let's see step by step how a penetration test is conducted on the target network.

Step1: 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 host, i.e., **accessible hosts** in the target network, using **network scanning tools** such as **Angry IP Scanner**, **Nmap**, **Netscan**, etc. It is difficult to detect live hosts behind the firewall.

Step 2: Port Scanning

Perform port scanning using tools such as **Nmap**, **Netscan Tools Pro**, **PRTG Network Monitor**, **Net Tools**, etc. These tools will help you to probe a server or host on the target network for open ports. **Open ports** are the **doorways** for attackers to install **malware** on a system. Therefore, you should check for open ports and close them if not necessary.

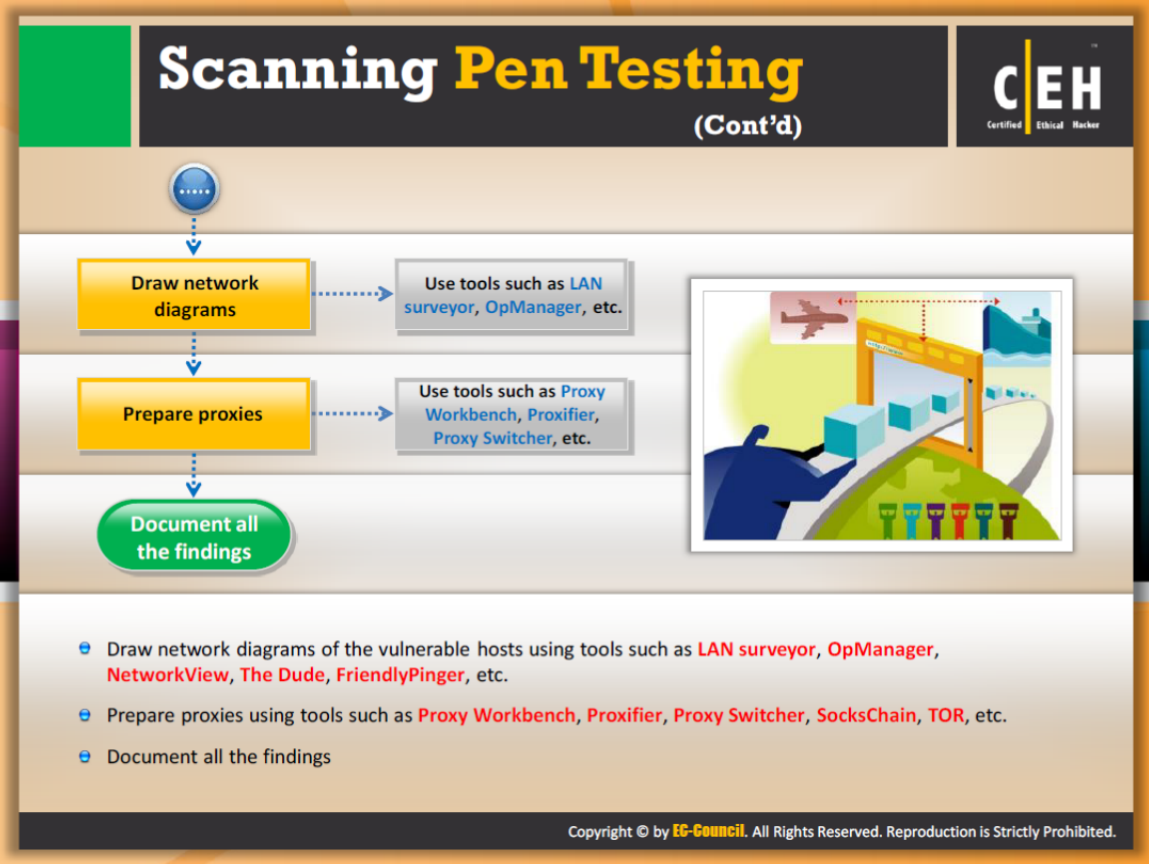
Step 3: Banner Grabbing or OS Finger Printing

Perform banner **grabbing/OS** fingerprinting using tools such as Telnet, Netcraft, ID Serve, Netcat, etc. This determines the **operating system** running on the target host of a network and its **version**. Once you know the version and 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 4: Scan for Vulnerabilities

Scan the network for vulnerabilities using **network vulnerability scanning tools** such as **Nessus**, **GFI LANGuard**, **SAINT**, **Core Impact Professional**, **Ratina CS**, **MBSA**, etc. These tools help you to find the vulnerabilities present in the target network. In this step, you will able to determine the **security weaknesses**/**loopholes** of the target system or network.



Scanning Pen Testing (Cont'd)

Step 5: Draw Network Diagrams

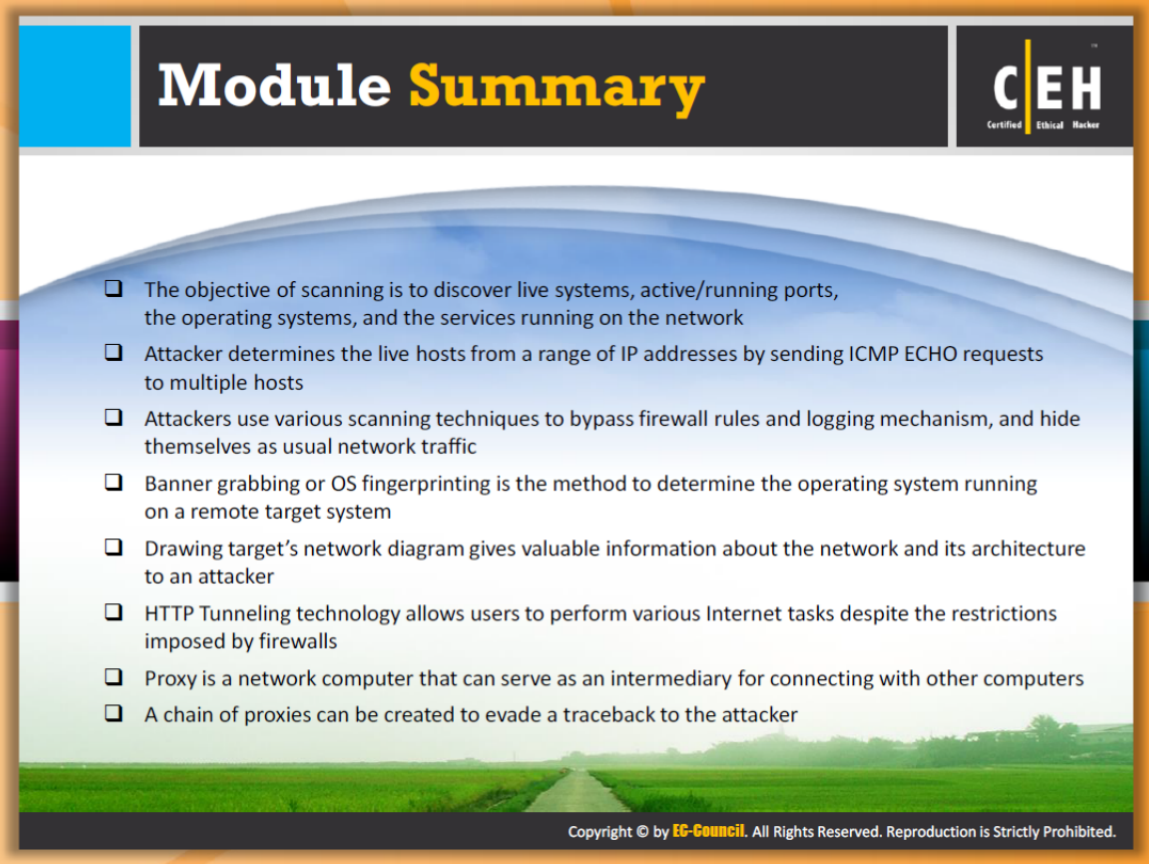
Draw a network diagram of the target organization that helps you to understand the **logical connection** and **path** to the **target host** in the network. The network diagram can be drawn with the help of tools such as **LAN surveyor**, **OpManager**, **LANState**, **FriendlyPinger**, etc. The network diagrams provide valuable information about the network and its architecture.

Step 6: Prepare Proxies

Prepare proxies using tools such as Proxifier, SocksChain, SSL Proxy, Proxy+, Gproxy, ProxyFinder, etc. to hide yourself from being caught.

Step 7: Document all Findings

The last but the most important step in **scanning penetration testing** is **preserving** all outcomes of tests conducted in previous steps in a document. This document will assist you in finding **potential vulnerabilities** in your network. Once you determine the potential vulnerabilities, you can plan the **counteractions** accordingly. Thus, penetration testing helps in assessing your network before it gets into real trouble that may cause severe loss in terms of value and finance.



Module Summary

Let's take a look at what you have learned in this module:

- 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, logging mechanism, and hide themselves as usual network traffic.
- **Banner Grabbing** or **OS fingerprinting** is the method to determine the operating system running on a remote target system.
- Drawing target's network diagram gives valuable information about the network and its **architecture** to an attacker.
- **HTTP Tunneling** technology allows users to perform various Internet tasks despite the restrictions imposed by firewalls.
- **Proxy** is a network computer that can serve as an intermediary for connecting with other computers.
- A chain of proxies can be created to evade a traceback to the attacker.