



**Module 03**

## **Scanning Networks**

# Module Objectives



## Module Objectives

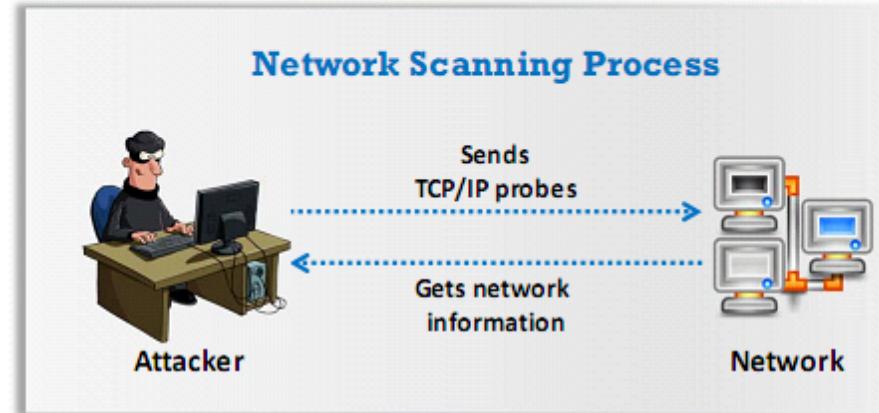
- Understanding Network Scanning Concepts
- Understanding various Scanning Tools
- Understanding various Scanning Techniques
- Understanding various Techniques to Scan Beyond IDS and Firewall
- Understanding Banner Grabbing
- Drawing Network Diagrams
- Overview of Scanning Pen Testing

# Module Flow

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

# Overview of Network Scanning

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



## Objectives of Network Scanning

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

# TCP Communication Flags

Data contained in the packet should be processed immediately

**URG**  
(Urgent)

There will be no further transmissions

**FIN**  
(Finish)

Resets a connection

**RST**  
(Reset)

**PSH**  
(Push)

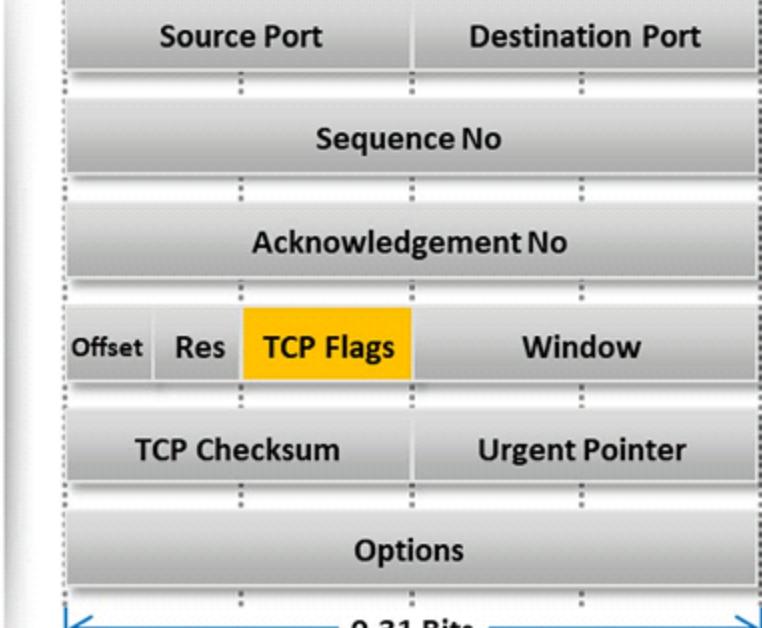
Sends all buffered data immediately

**ACK**  
(Acknowledgement)

Acknowledges the receipt of a packet

**SYN**  
(Synchronize)

Initiates a connection between hosts

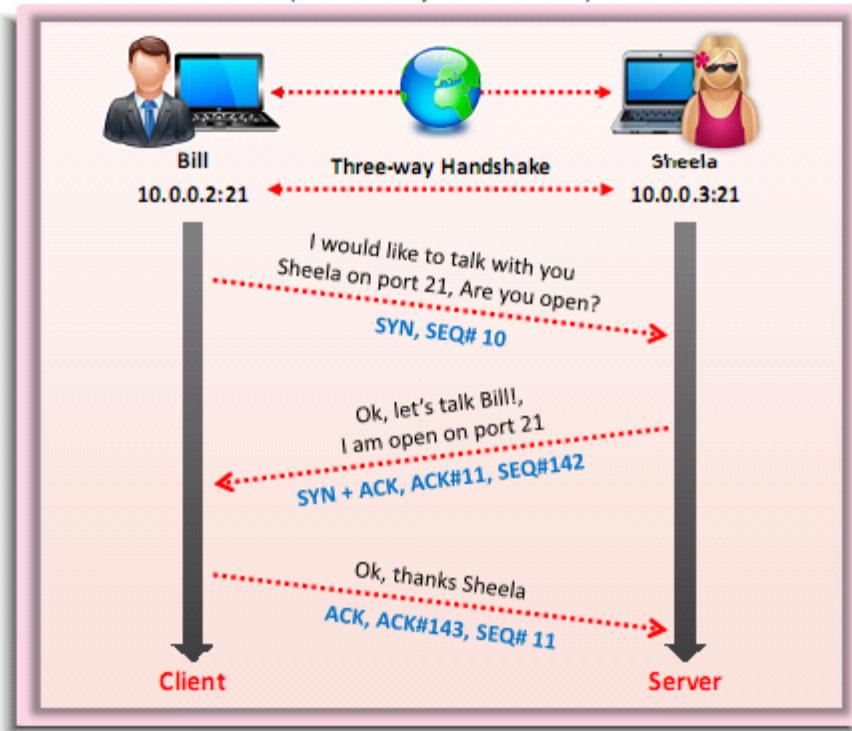


Standard TCP communications are controlled by flags in the TCP packet header

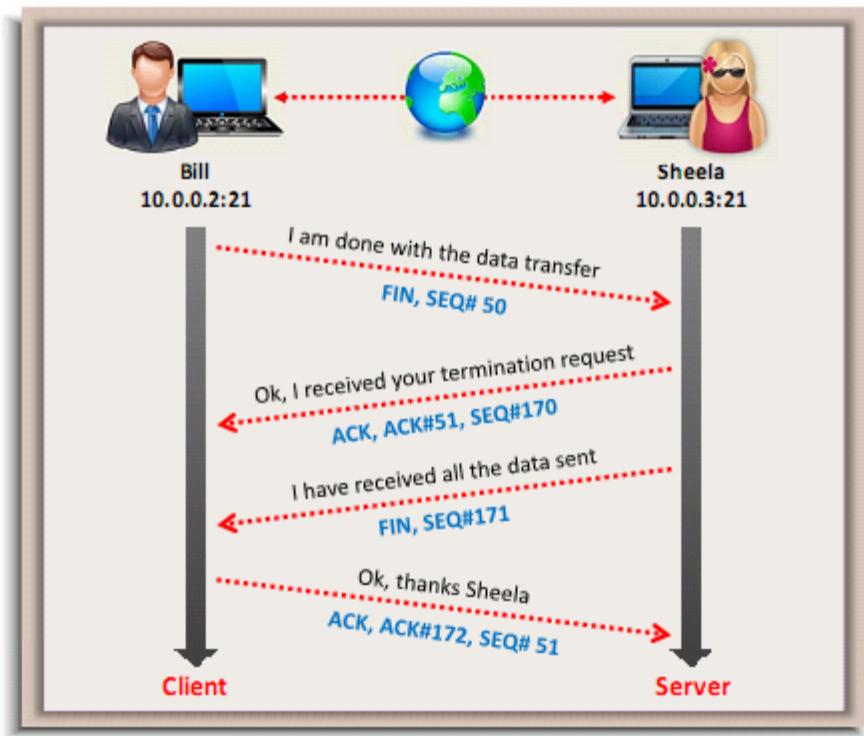
# TCP/IP Communication

## TCP Session Establishment

(Three-way Handshake)

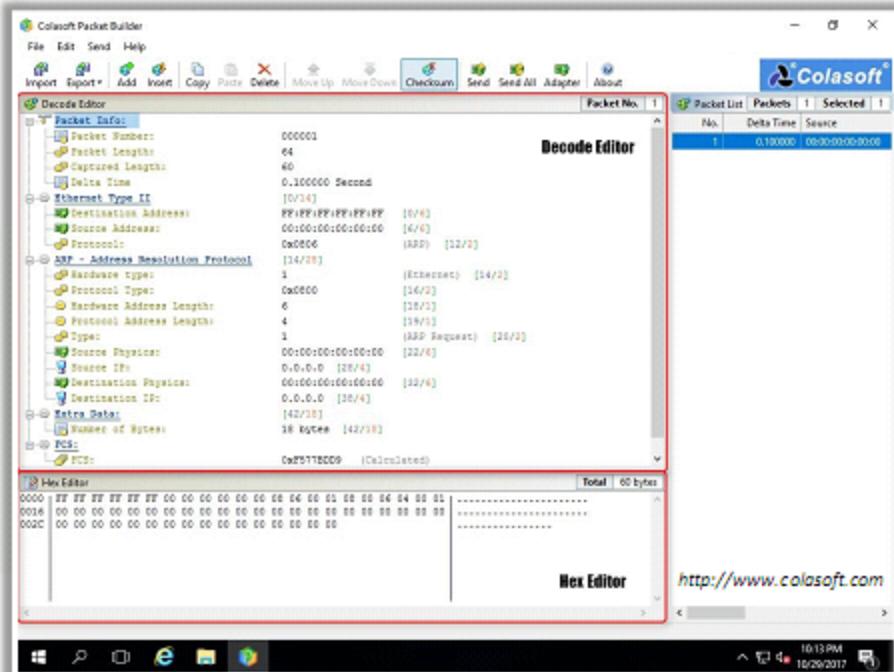


## TCP Session Termination



# Creating Custom Packet Using TCP Flags

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



## Packet Crafting Tools

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

# Scanning in IPv6 Networks



IPv6 increases the IP address size from **32 bits** to **128 bits**, to support more levels of addressing hierarchy



Traditional network scanning techniques will be **computationally less feasible** due to the larger search space (64 bits of host address space or  $2^{64}$  addresses) provided by the IPv6 in a subnet



Scanning in the IPv6 network is more difficult and complex when compared to the IPv4. Additionally, a number of scanning tools do not support ping sweeps on **IPv6 networks**



Attackers need to harvest IPv6 addresses from **network traffic**, **recorded logs**, or **Received from:** and other header lines in the archived email or Usenet news messages



Scanning the IPv6 network, however, offers a large number of hosts in a subnet. Once an attacker is able to compromise one host in the subnet, he or she can probe the "**all hosts**" and **link local multicast address**

# Module Flow

1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Scanning Techniques**

7

**Scanning Pen Testing**

4

**Scanning Beyond IDS and Firewall**

5

**Banner Grabbing**

6

**Draw Network Diagrams**

# Scanning Tool: Nmap

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



```
Zenmap
Scan Tools Profile Help
Target: 10.10.10.10 Profile: Intense scan, all TCP ports Scan Cancel
Command: nmap -p 1-65535 -T4 -A -v 10.10.10.10

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12
16:09 Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 16:09
Completed NSE at 16:09, 0.00s elapsed
Initiating NSE at 16:09
Completed NSE at 16:09, 0.00s elapsed
Initiating ARP Ping Scan at 16:09
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 16:09, 0.75s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:09
Completed Parallel DNS resolution of 1 host. at 16:09, 0.00s elapsed
Initiating SYN Stealth Scan at 16:09
Scanning 10.10.10.10 [65535 ports]
Discovered open port 443/tcp on 10.10.10.10
Discovered open port 135/tcp on 10.10.10.10
Discovered open port 139/tcp on 10.10.10.10
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 88/tcp on 10.10.10.10
Increasing send delay for 10.10.10.10 from 0 to 5 due to 38 out of 94 dropped probes since last increase.
Discovered open port 5357/tcp on 10.10.10.10
Discovered open port 7688/tcp on 10.10.10.10
SYN Stealth Scan Timing: About 6.71% done; ETC: 16:17
(0:07:11 remaining)
Discovered open port 49668/tcp on 10.10.10.10
Discovered open port 49870/tcp on 10.10.10.10
SYN Stealth Scan Timing: About 13.37% done; ETC: 16:18
(0:06:35 remaining)
SYN Stealth Scan Timing: About 19.96% done; ETC: 16:19
(0:06:05 remaining)

Filter Hosts
```

```
Zenmap
Scan Tools Profile Help
Target: 10.10.10.10 Profile: Intense scan, all TCP ports Scan Cancel
Command: nmap -p 1-65535 -T4 -A -v 10.10.10.10

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host 10.10.10.10
PORT STATE SERVICE VERSION
80/tcp open http Microsoft IIS httpd 10.0
| http-methods:
|_ Supported Methods: OPTIONS TRACE GET HEAD POST
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
443/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
|_ssl-cert: Subject: commonName=Client-01.CAST.com
|_Issuer: commonName=Client-01.CAST.com
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2016-11-20T13:51:11
| Not valid after: 2017-11-20T00:00:00
| MD5: 6a1a 0f01 b9d3 a5de 3850 2bc7 3676 2b30
|_SHA-1: 57c9 e713 3078 72d3 10c4 71d8 49ee 7467 cc49
|F12
|_ssl-date: 2018-01-12T10:47:50+00:00; 8s from scanner time.
445/tcp open microsoft-ds Windows 10 Enterprise
10586 microsoft-ds (workgroup: CAST)
5357/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
7688/tcp open pando-pub?
17500/tcp open ssl/db-lsp?
49664/tcp open msrpc Microsoft Windows RPC

Filter Hosts
```

<https://nmap.org>

# Scanning Tool: Hping2 / Hping3

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

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

<http://www.hping.org>

## ICMP Scanning

```
File Edit View Search Terminal Help
root@kali:~# hping3 -1 10.10.10.10
HPING 10.10.10.10 (eth0 10.10.10.10): icmp mode set, 28 headers + 0 data bytes
len=46 ip=10.10.10.10 ttl=128 id=2860 icmp_seq=0 rtt=5.5 ms
len=46 ip=10.10.10.10 ttl=128 id=2861 icmp_seq=1 rtt=6.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2862 icmp_seq=2 rtt=5.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2863 icmp_seq=3 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2864 icmp_seq=4 rtt=4.0 ms
len=46 ip=10.10.10.10 ttl=128 id=2865 icmp_seq=5 rtt=2.9 ms
len=46 ip=10.10.10.10 ttl=128 id=2866 icmp_seq=6 rtt=1.9 ms
len=46 ip=10.10.10.10 ttl=128 id=2867 icmp_seq=7 rtt=1.1 ms
len=46 ip=10.10.10.10 ttl=128 id=2868 icmp_seq=8 rtt=5.2 ms
len=46 ip=10.10.10.10 ttl=128 id=2869 icmp_seq=9 rtt=3.8 ms
len=46 ip=10.10.10.10 ttl=128 id=2870 icmp_seq=10 rtt=8.1 ms
len=46 ip=10.10.10.10 ttl=128 id=2871 icmp_seq=11 rtt=3.1 ms
^C
--- 10.10.10.10 hping statistic ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 1.1/4.2/8.1 ms
root@kali:~#
```

## ACK Scanning on port 80

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

# Hping 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
```



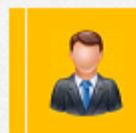
SYN scan on port 50-60

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



FIN, PUSH and URG scan on port 80

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



Scan entire subnet for live host

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



Intercept all traffic containing HTTP signature

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



SYN flooding a victim

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

# Scanning Tools

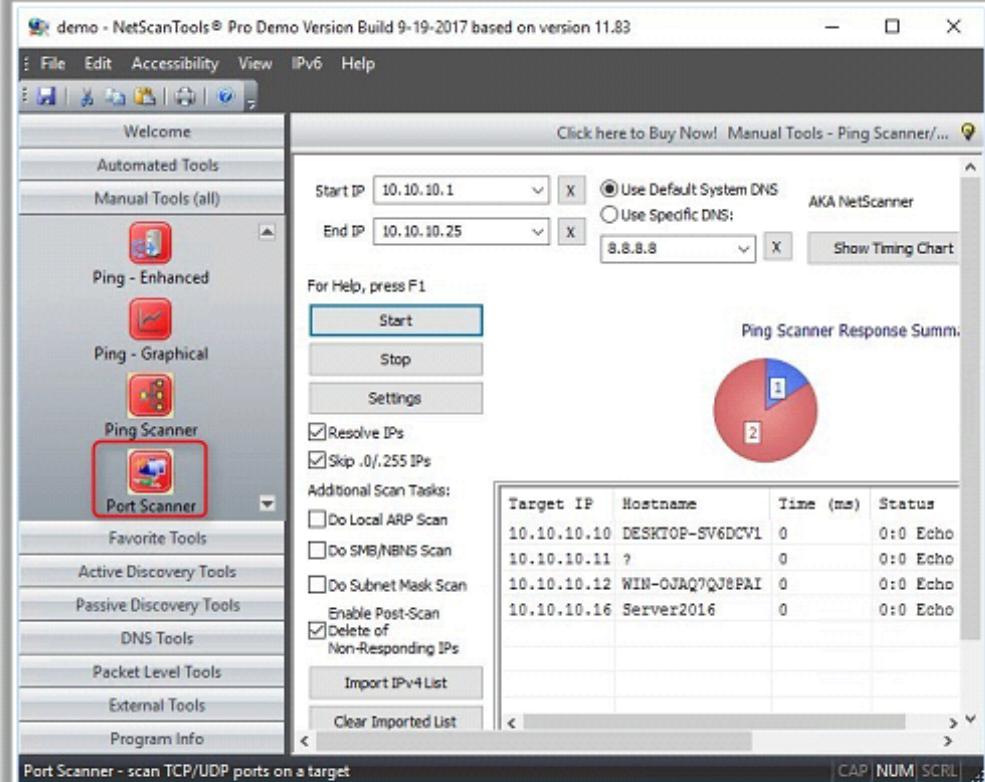
## NetScanTools Pro

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

## Scanning Tools

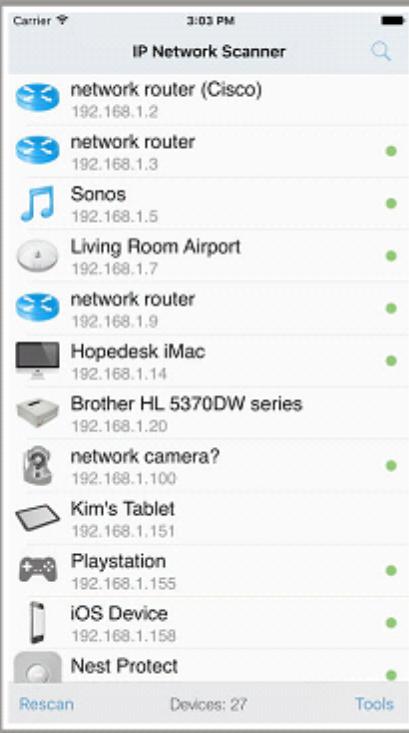
- SuperScan (<https://www.mcafee.com>)
- PRTG Network Monitor (<https://www.paessler.com>)
- OmniPeek (<https://www.savvius.com>)
- MiTeC Network Scanner (<http://www.mitec.cz>)
- NEWT Professional (<http://www.komodolabs.com>)
- MegaPing (<http://www.magnetosoft.com>)

## NetScanTools Pro


<http://www.netscantools.com>

# Scanning Tools for Mobile

## IP Scanner



## Fing



## Hackode

<https://play.google.com>



## zANTI

<https://www.zimperium.com>



## cSploit

<http://www.csploit.org>



## FaceNiff

<http://www.effecthacking.com>



## PortDroid Network Analysis

<https://play.google.com>



# Module Flow

1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Scanning Techniques**

7

**Scanning Pen Testing**

4

**Scanning Beyond IDS and Firewall**

5

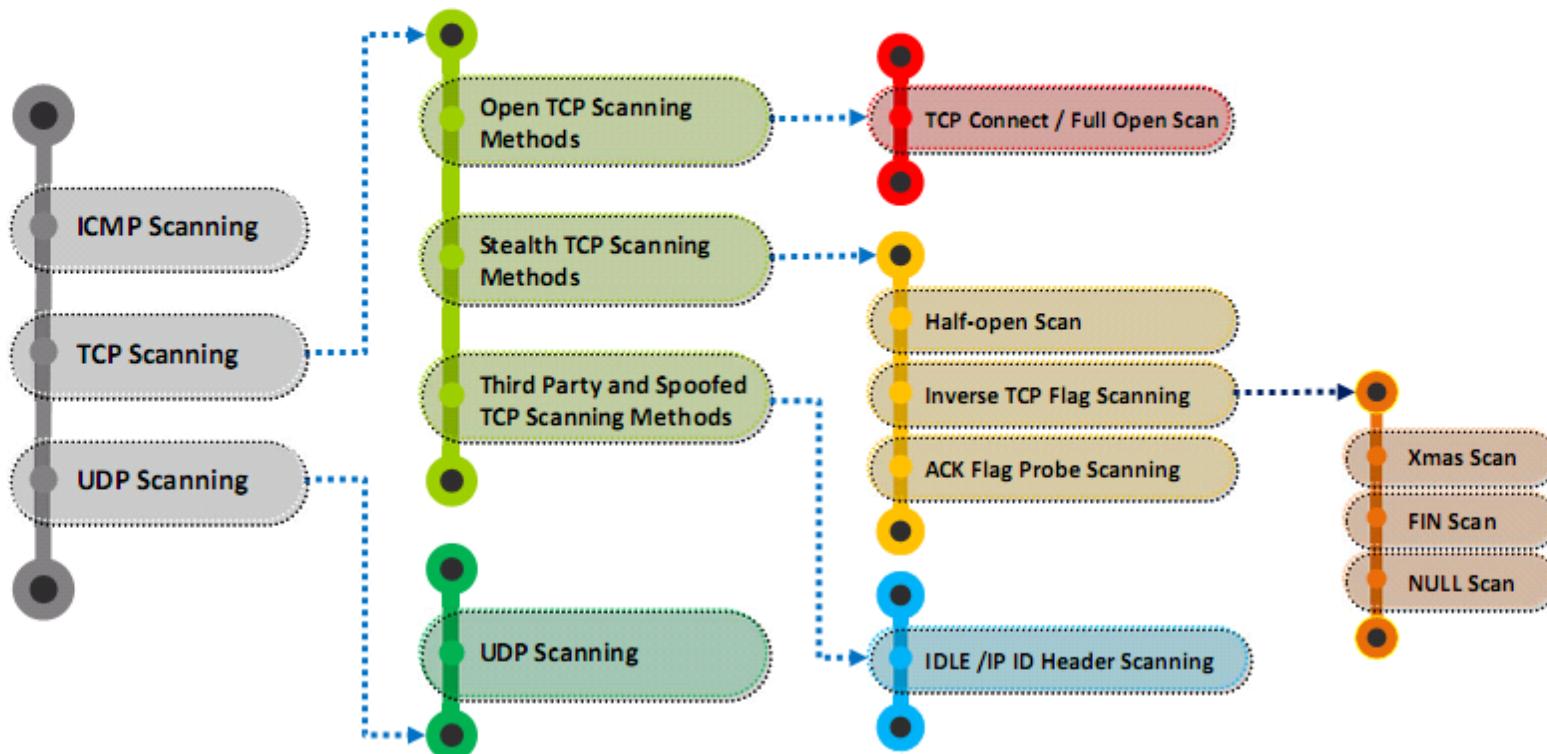
**Banner Grabbing**

6

**Draw Network Diagrams**

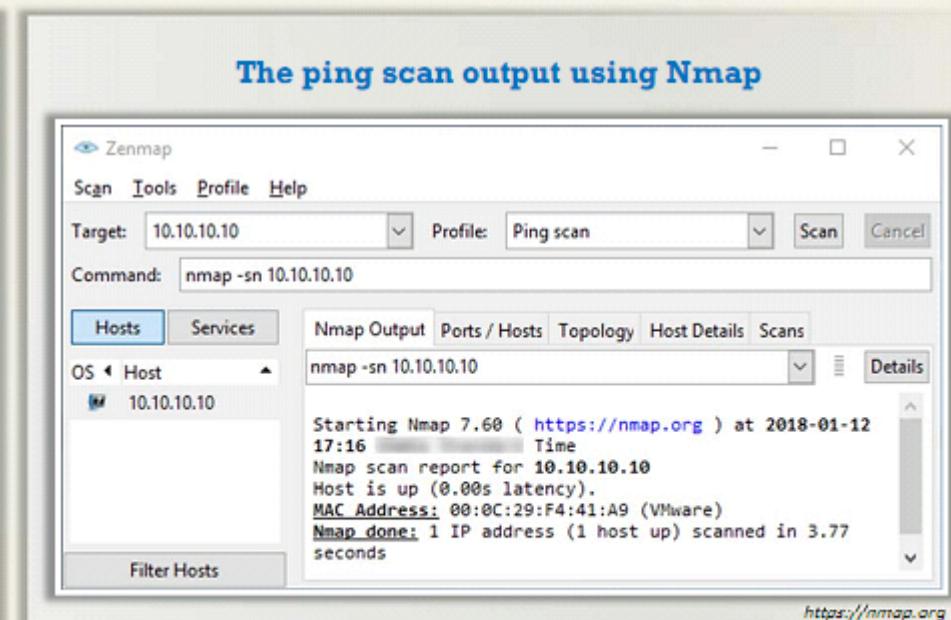
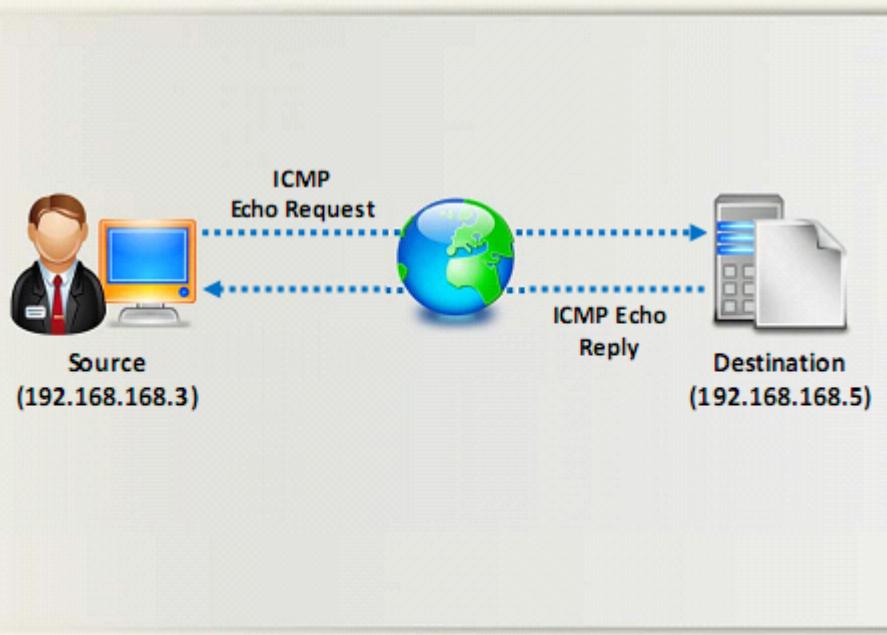
# Scanning Techniques

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



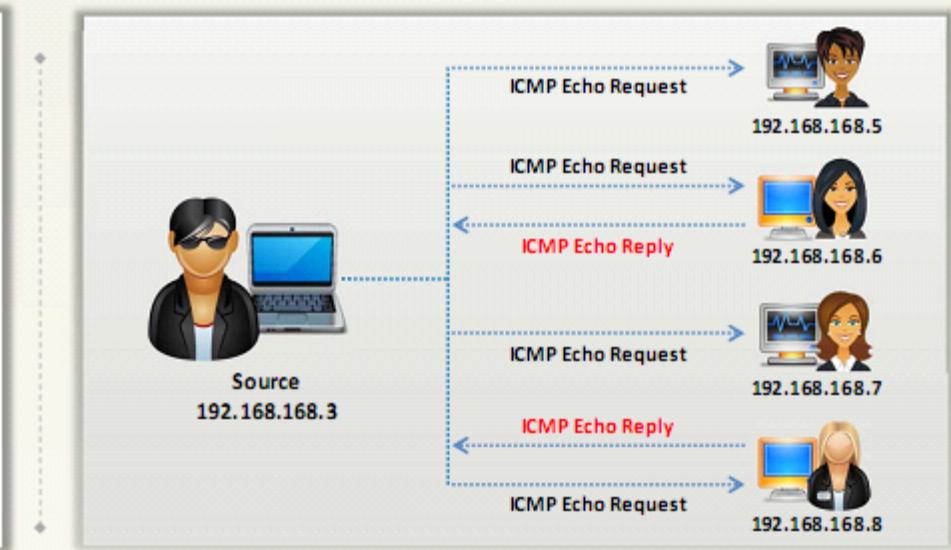
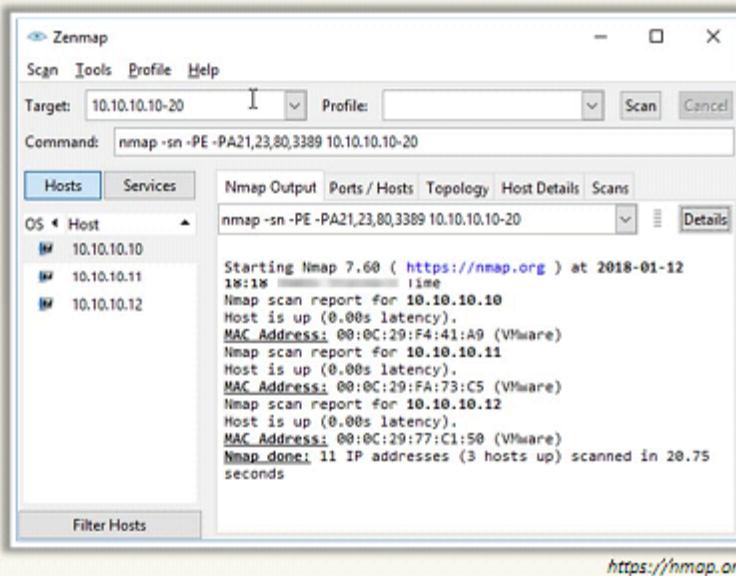
# ICMP Scanning - Checking for Live Systems

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



# Ping Sweep - Checking for Live Systems

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



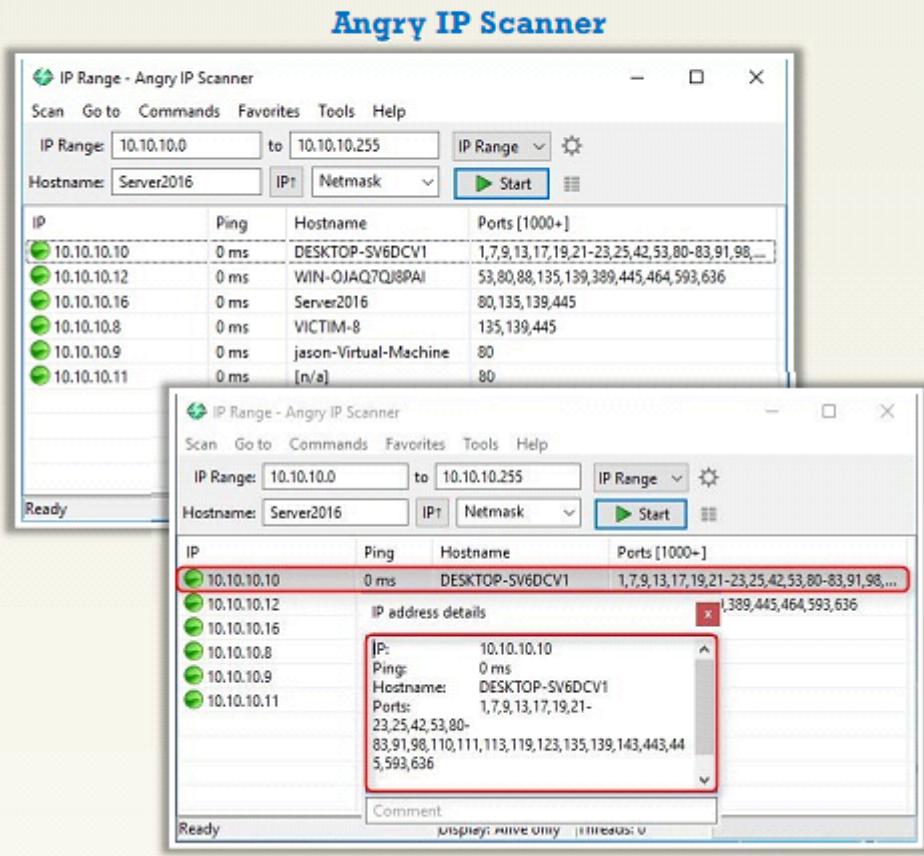
# Ping Sweep Tools

## Angry IP Scanner

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

## Ping Sweep Tools

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



# ICMP Echo Scanning

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

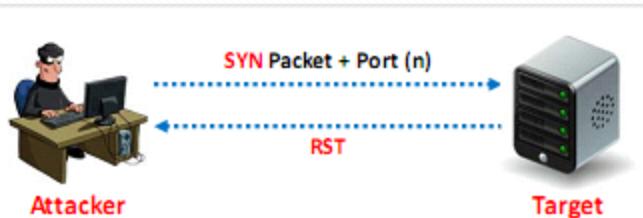
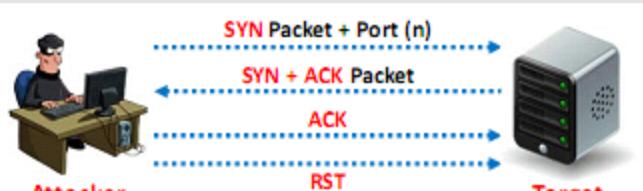


The screenshot shows the Zenmap interface with the title "Zenmap". The "Targets" section has "Target: 10.10.10.20" and "Profile: Ping scan". The "Command" field contains "nmap -sn 10.10.10.20". The main window has tabs for "Hosts" (selected), "Services", "Nmap Output", "Ports / Hosts", "Topology", "Host Details", and "Scans". The "Hosts" tab displays three hosts: 10.10.10.10, 10.10.10.11, and 10.10.10.12. The "Nmap Output" tab shows the following text:

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

# TCP Connect / Full Open Scan

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



```
Zenmap
Scan Tools Profile Help
Target: nmap 10.10.10.10 Profile:
Command: # -sT -v nmap 10.10.10.10
Hosts Services
OS Host
10.10.10.10
10.10.10.11
Nmap Output Ports / Hosts Topology Host Details Scans
# -sT -v nmap 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12 18:44
Time
Initiating ARP Ping Scan at 18:44
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 18:44, 0.74s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:44
Completed Parallel DNS resolution of 1 host. at 18:44, 0.00s
elapsed
Initiating Connect Scan at 18:44
Scanning 10.10.10.10 [1000 ports]
Discovered open port 139/tcp on 10.10.10.10
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 135/tcp on 10.10.10.10
Discovered open port 80/tcp on 10.10.10.10
Discovered open port 443/tcp on 10.10.10.10
Connect Scan Timing: About 72.93% done; ETC: 18:47 (0:00:56 remaining)
Discovered open port 5357/tcp on 10.10.10.10
Completed Connect Scan at 18:47, 212.18s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Failed to resolve "nmap".
Host is up (1.0s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
MAC Address: 00:0C:29:F4:41:A9 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 218.30 seconds
Raw packets sent: 1 (288) | Rcvd: 1 (288)
```

# Stealth Scan (Half-open Scan)

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

## Stealth Scan Process

The client sends a single **SYN** packet to the server to the appropriate port

**01**

If the port is open, subsequently, the server will respond with an **SYN/ACK** packet

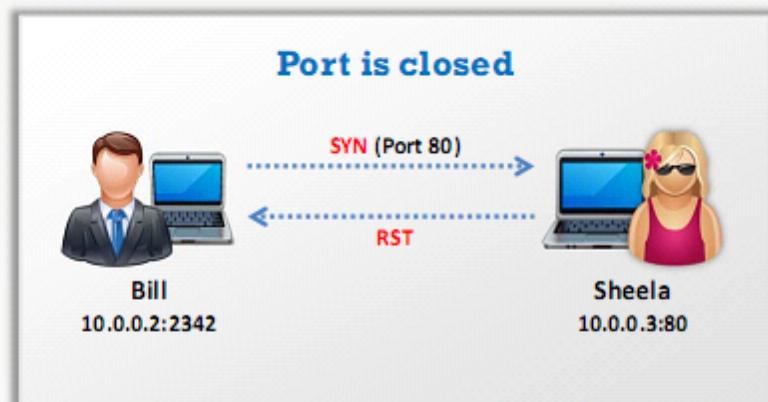
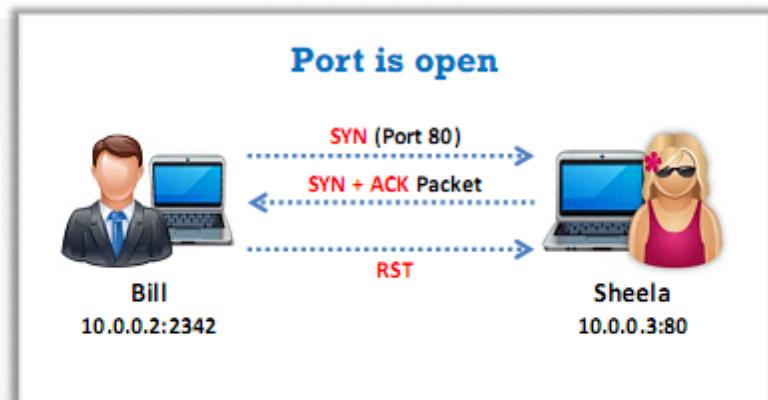
**02**

If the server responds with an **RST** packet, then the remote port is in the "closed" state

**03**

The client sends the **RST** packet to close the initiation before a connection can ever be established

**04**



# Inverse TCP Flag Scanning

01

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

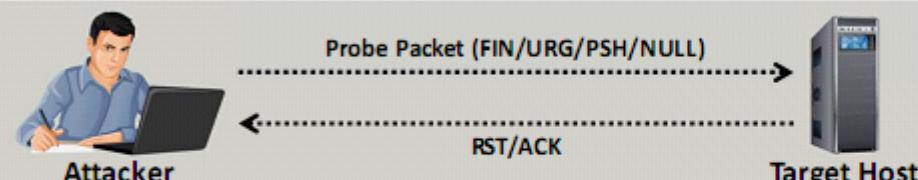
02

Port is open



03

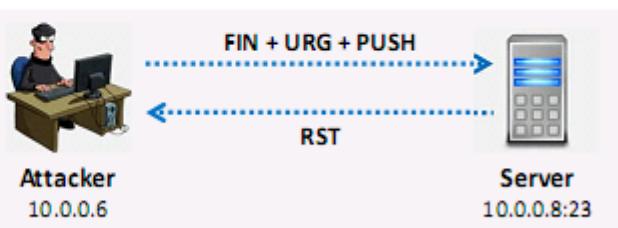
Port is closed



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

# Xmas Scan

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



**Xmas Scan Using Nmap**

```

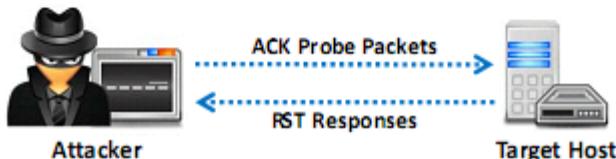
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-12 19:28
Standard Time
Initiating ARP Ping Scan at 19:28
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 19:28, 0.75s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:28
Completed Parallel DNS resolution of 1 host.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

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

# ACK Flag Probe Scanning

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

TTL based ACK flag probe scanning



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

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

WINDOW based ACK flag probe scanning

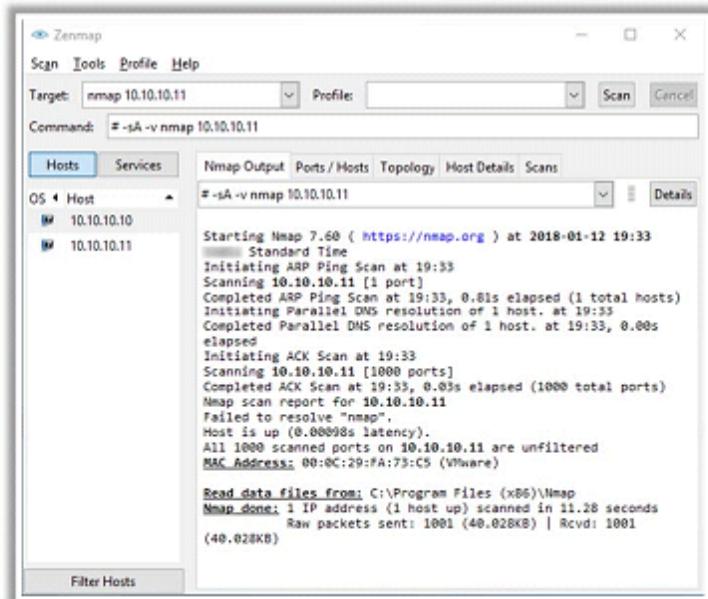
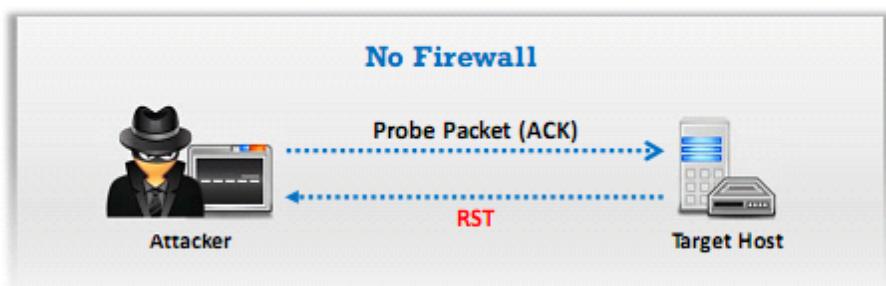
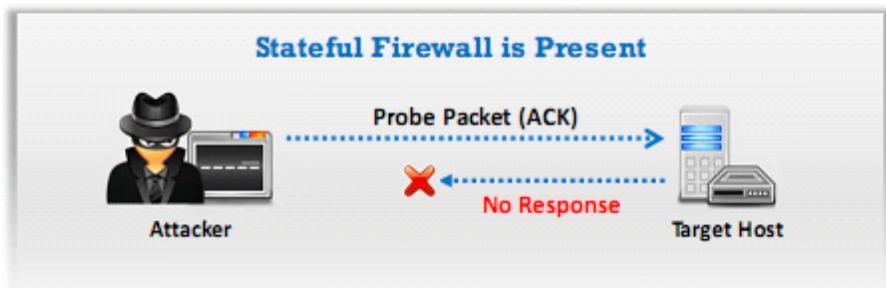


```
1: host 10.2.2.12 port 20: F;RST -> ttl: 64 win: 0
2: host 10.2.2.12 port 21: F;RST -> ttl: 64 win: 0
3: host 10.2.2.12 port 22: F;RST -> ttl: 64 win: 512
4: host 10.2.2.12 port 23: F;RST -> ttl: 64 win: 0
```

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

# ACK Flag Probe Scanning (Cont'd)

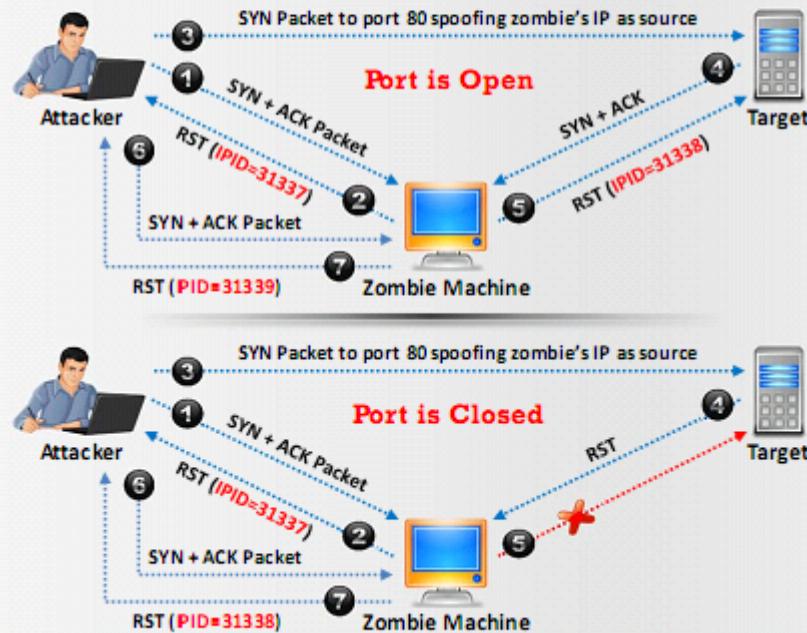
- ACK flag probe scanning can also be used to **check the filtering system of target**
- Attackers send an **ACK probe packet** with a random sequence number, no response implies that the **port is filtered** (stateful firewall is present) and RST response means that the **port is not filtered**



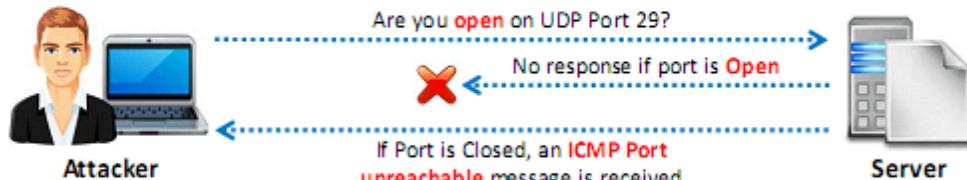
# IDLE/IPID Header Scan

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

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



# UDP Scanning



## UDP Port Open

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

## UDP Port Closed

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

```

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-13 10:51
Standard time
Initiating ARP Ping Scan at 10:51
Scanning 10.10.10.11 [1 port]
Completed ARP Ping Scan at 10:51, 0.76s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:51
Completed Parallel DNS resolution of 1 host. at 10:51, 0.00s elapsed
Initiating UDP Scan at 10:51
Scanning 10.10.10.11 [1000 ports]
Increasing send delay for 10.10.10.11 from 0 to 50 due to max_successful_tryms increase to 4
Increasing send delay for 10.10.10.11 from 50 to 100 due to 11 out of 24 dropped probes since last increase.
UDP Scan Timing: About 11.38% done; ETC: 10:56 (0:04:01 remaining)
Completed UDP Scan at 11:08. 995.27s elapsed (1000 total ports)
Nmap scan report for 10.10.10.11
Failed to resolve "nmap".
Host is up (0.000072s latency).
All 1000 scanned ports on 10.10.10.11 are closed (921) or open|filtered (79)
MAC Address: 00:0C:29:FA:73:CS (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1002.37 seconds
Raw packets sent: 1846 (53.832KB) | Rcvd: 1903 (107.852KB)

```

# SSDP and List Scanning

## SSDP Scanning

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

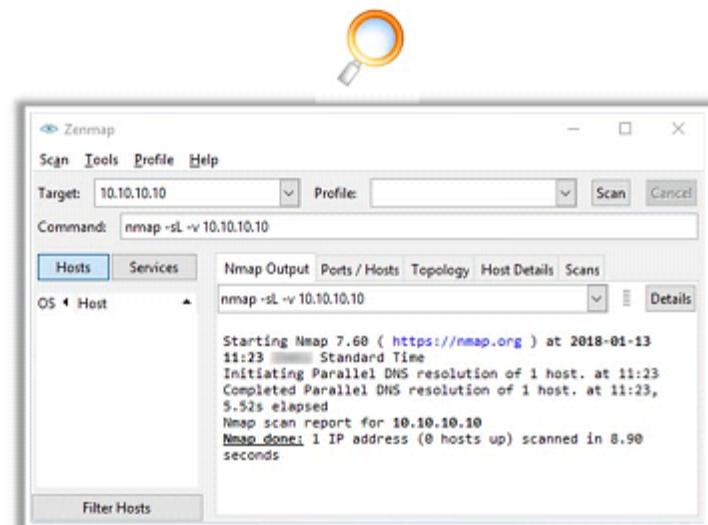
```
root@kali: ~
File Edit View Search Terminal Help
nse > use auxiliary/scanner/upnp/ssdp_search
nse auxiliary(ssdp_search) > set RHOSTS 192.168.0.17
RHOSTS => 192.168.0.17
nse auxiliary(ssdp_search) > show options

Module options (auxiliary/scanner/upnp/ssdp_search):
Name      Current Setting  Required  Description
----      -----          -----    -----
BATCHSIZE        256          yes       The number of hosts to probe in each set
CHOST           no           no        The local client address
REPORT_LOCATION  false        yes       This determines whether to report the UPnP e
RHOSTS          192.168.0.17   yes       The target address range or CIDR identifier
REPORT          1900         yes       The target port
THREADS          1            yes       The number of concurrent threads

nse auxiliary(ssdp_search) > exploit
[*] The quicker you become the more you are able to hear
[*] Sending UPnP SSDP probes to 192.168.0.17->192.168.0.17 (1 hosts)
[*] No SSDP endpoints found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
nse auxiliary(ssdp_search) >
```

## List Scanning

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



# Port Scanning Countermeasures

**01**

Configure **firewall** and **IDS rules** to detect and block probes

**05**

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

**02**

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

**06**

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

**03**

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

**07**

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

**04**

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

**08**

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

# Module Flow

1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Scanning Techniques**

7

**Scanning Pen Testing**

4

**Scanning Beyond IDS and Firewall**

5

**Banner Grabbing**

6

**Draw Network Diagrams**

# IDS/Firewall Evasion Techniques

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

# Packet Fragmentation

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

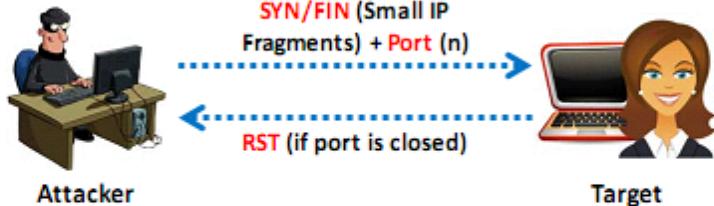


- The **TCP header** is split into several packets so that the packet filters are not able to detect what the packets intends to do

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

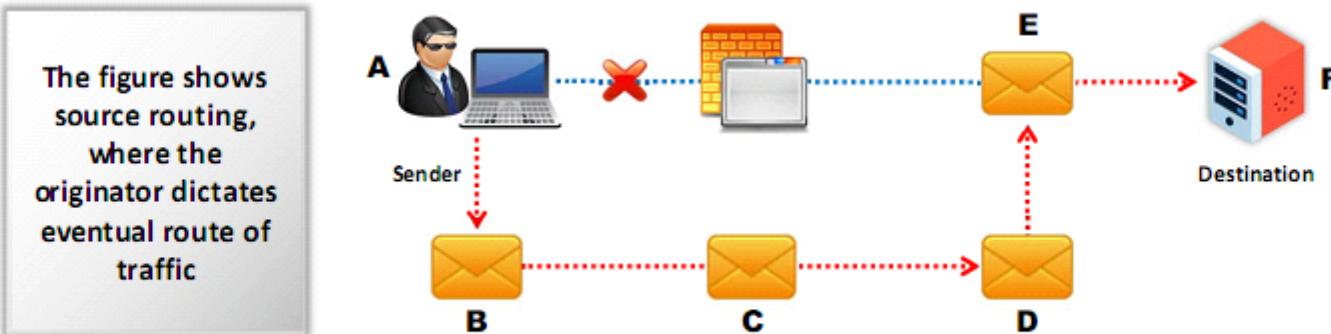
Starting Nmap 7.60 ( http://nmap.org ) at
2017-02-10 11:03 EDT
Initiating SYN Stealth Scan at 11:03
Scanning 192.168.168.5 [1000 ports]
Discovered open port 139/tcp on 192.168.168.5
Discovered open port 445/tcp on 192.168.168.5
Discovered open port 135/tcp on 192.168.168.5
Discovered open port 912/tcp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

## SYN/FIN Scanning Using IP Fragments



# Source Routing

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



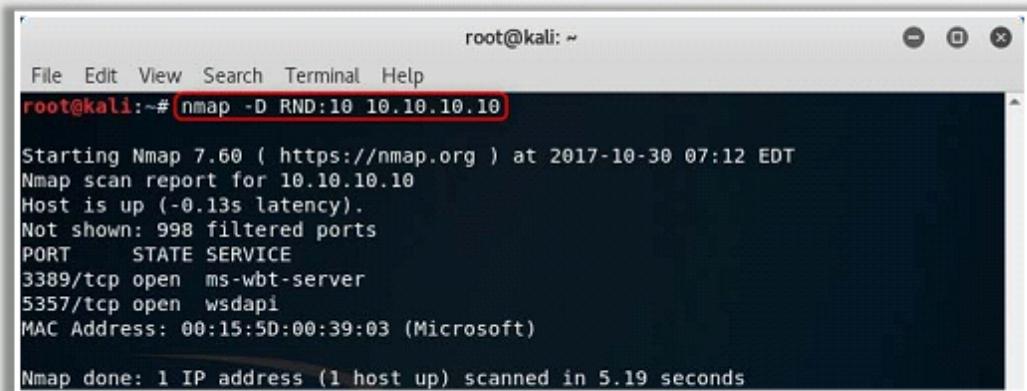
# IP Address Decoy

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

## Decoy Scanning using Nmap

Nmap has two options for decoy scan:

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



```
root@kali:~# nmap -D RND:10 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2017-10-30 07:12 EDT
Nmap scan report for 10.10.10.10
Host is up (-0.13s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

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

# IP Address Spoofing

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



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

## IP Spoofing Detection Techniques: Direct TTL Probes

01

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

02

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



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

# IP Spoofing Detection Techniques: IP Identification Number

01

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

02

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

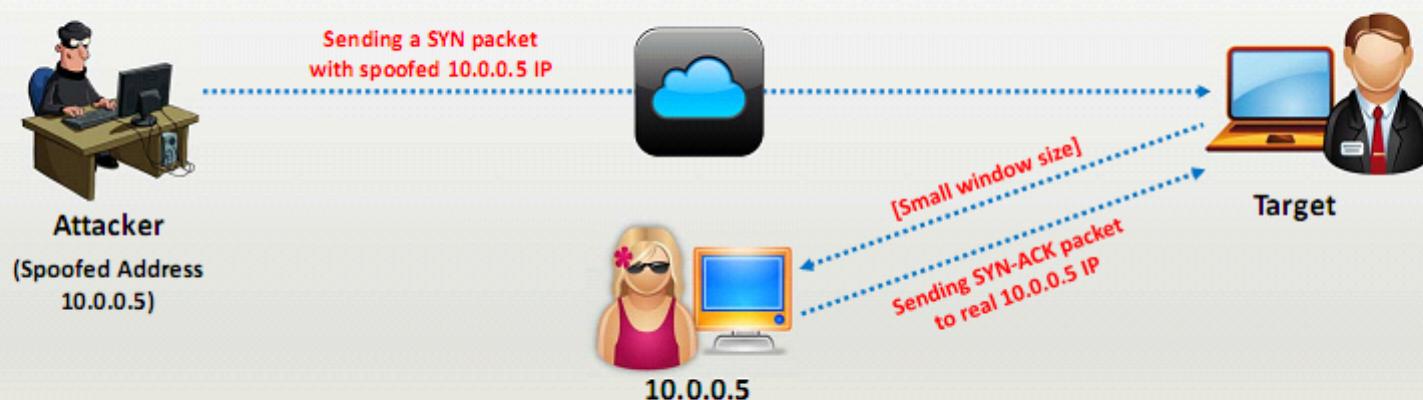
03

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



## IP Spoofing Detection Techniques: TCP Flow Control Method

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



# IP Spoofing Countermeasures

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

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

**Use multiple firewalls** providing multi-layered depth of protection

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

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

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

# Proxy Servers

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

## Why Attackers Use Proxy Servers?

To hide the actual source of a scan and **evade certain IDS/firewall restrictions**



To **mask the actual source** of the attack by impersonating a fake source address of the proxy



To **remotely access intranets and other website resources** that are normally off limits



To **interrupt all the requests sent by a user** and transmit them to a third destination, hence victims will only be able to identify the proxy server address



To chain **multiple proxy servers** to avoid detection



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

# Proxy Chaining

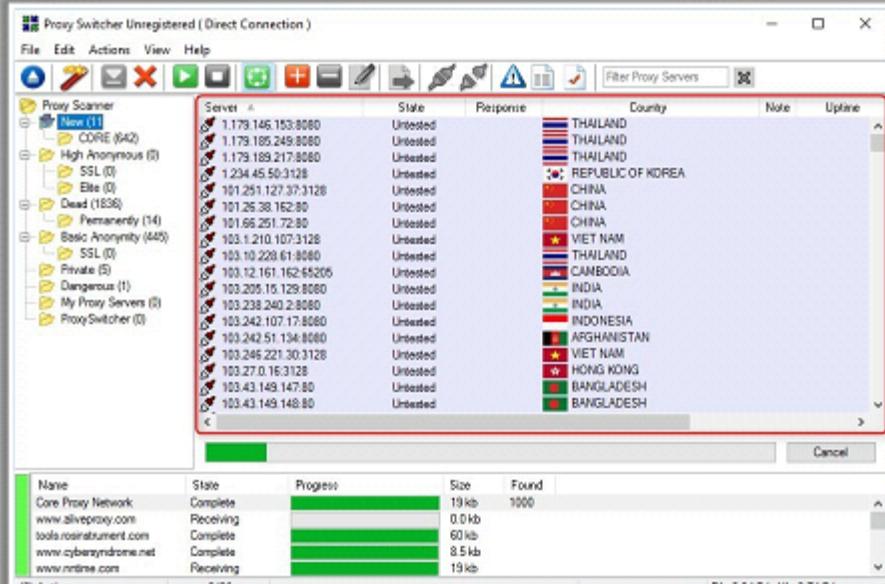
- 01 User **requests a resource** from the destination
- 02 Proxy client at the user's system connects to a **proxy server** and passes the request to proxy server
- 03 The proxy server **strips the user's identification information** and passes the request to next proxy server
- 04 This process is repeated by all the proxy servers in the **chain**
- 05 At the end, the **unencrypted request** is passed to the web server



# Proxy Tools: Proxy Switcher and Proxy Workbench

## Proxy Switcher

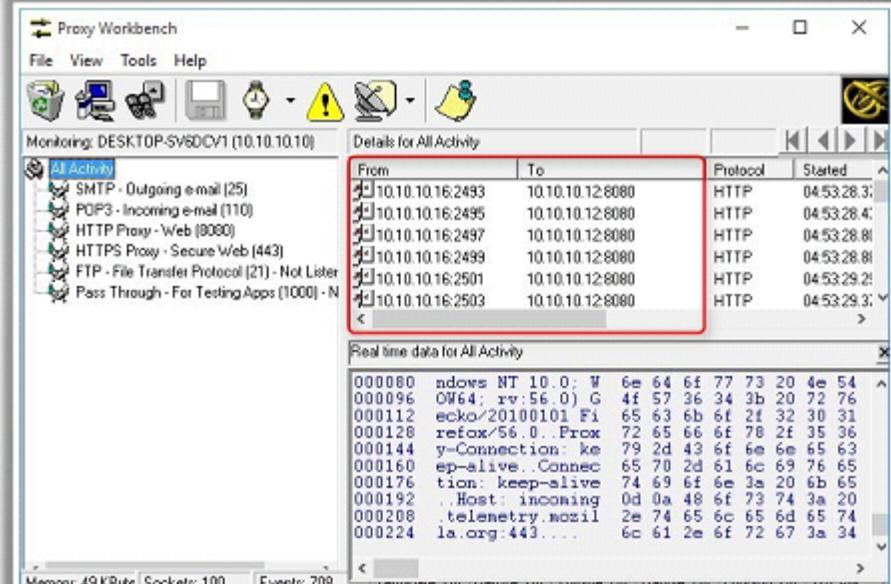
- Proxy Switcher allows you to **surf anonymously on the Internet** without disclosing your IP address



<http://www.proxyswitcher.com>

## Proxy Workbench

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

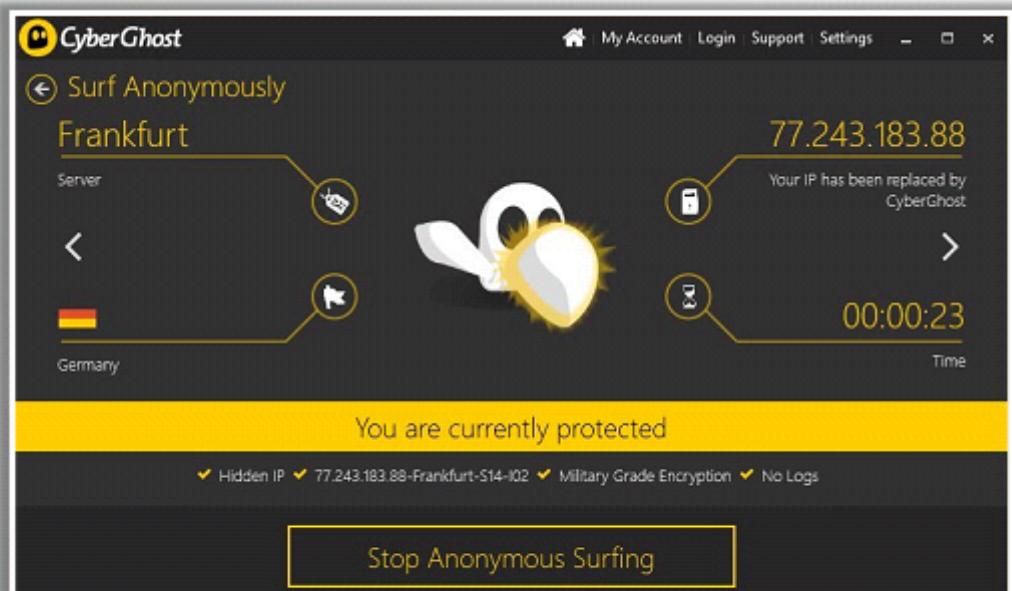


<http://proxyworkbench.com>

# Proxy Tools

## CyberGhost VPN

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



<https://www.cyberghostvpn.com>



### Tor

<https://www.torproject.org>



### Burp Suite

<https://www.portswigger.net>



### Hotspot Shield

<https://www.hotspotshield.com>



### Proxifier

<https://www.proxifier.com>

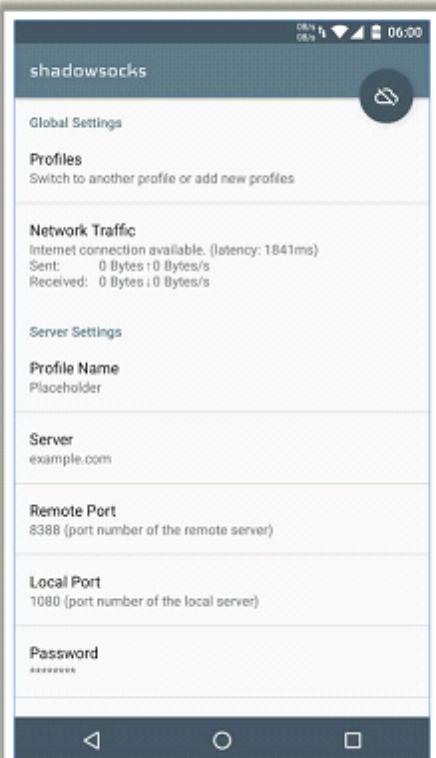


### Charles

<https://www.charlesproxy.com>

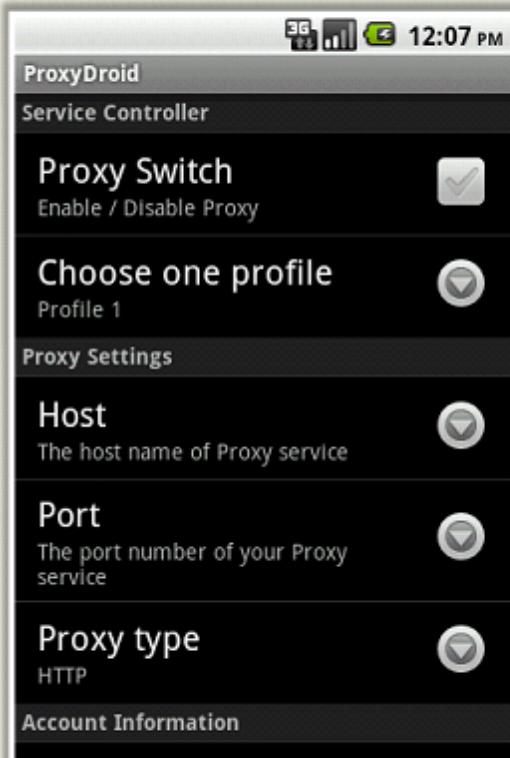
# Proxy Tools for Mobile

## Shadowsocks



<https://shadowsocks.org>

## ProxyDroid



<https://github.com>

## CyberGhost VPN

<https://www.cyberghostvpn.com>



## Servers Ultimate

<http://www.icecoldapps.com>



## Hotspot Shield

<https://www.hotspotshield.com>



## NetShade

<http://www.raynersw.com>



## Proxy Manager

<https://play.google.com>



# Anonymizers

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

Anonymizers make activity on the Internet untraceable

Anonymizers allow you to bypass Internet censors

## Why use Anonymizer?

Privacy and anonymity

Protects from online attacks



Access restricted content

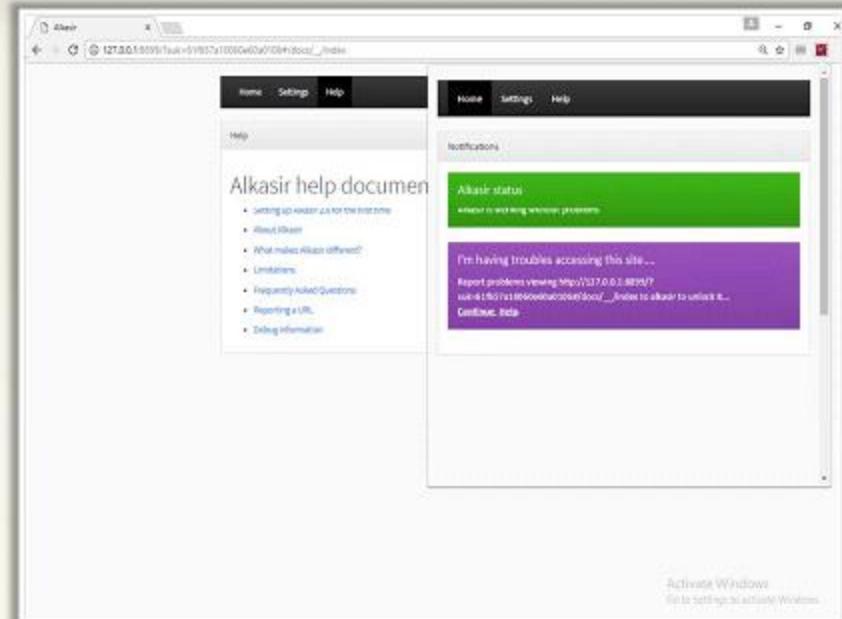
Bypass IDS and Firewall rules



# Censorship Circumvention Tools: Alkasir and Tails

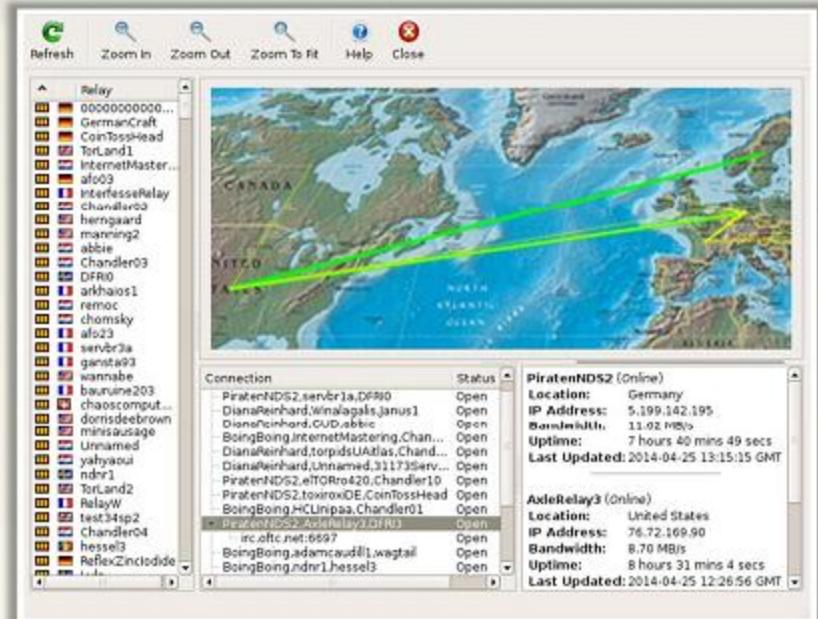
## Alkasir

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



## Tails

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



# Anonymizers

Whonix

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

<https://www.whonix.org>



## TunnelBear



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



**JonDo**  
<https://anonymous-proxy-servers.net>



Proxify  
<https://proxify.com>



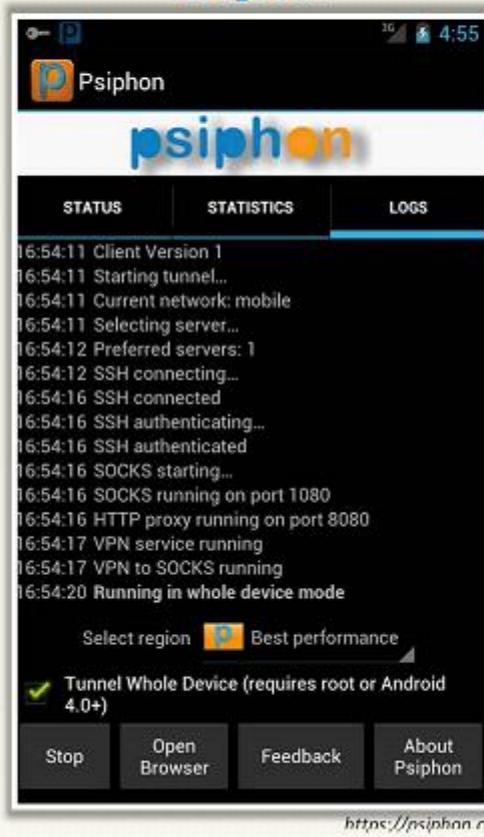
[Psiphon](https://psiphon.com)

# Anonymizers for Mobile

Orbot



Psiphon



OpenDoor



# Module Flow

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

# Banner Grabbing

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

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

## Active Banner Grabbing

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



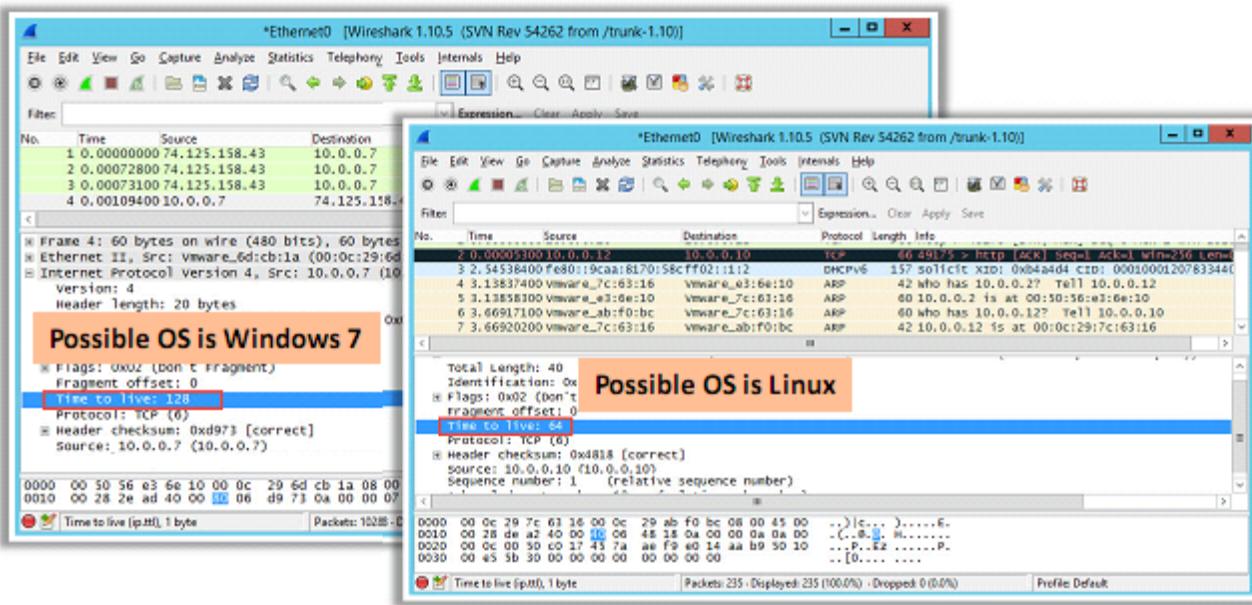
## Passive Banner Grabbing

- **Banner grabbing from error messages**  
Error messages provide information such as the type of server, type of OS, and SSL tool used by the target remote system.
- **Sniffing the network traffic**  
Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
- **Banner grabbing from page extensions**  
Looking for an extension in the URL may assist in determining the application's version.  
**Example:** .aspx => IIS server and Windows platform

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

# How to Identify Target System OS

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



## Values for the Operating Systems

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

# Banner Grabbing Countermeasures

## Disabling or Changing Banner

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

## Hiding File Extensions from Web Pages

- File extensions reveal information about the underlying server technology that an attacker can utilize to launch attacks
  - Hide file extensions to **mask the web technology**
  - Change **application mappings** such as .asp with .htm or .foo, etc. to disguise the identity of the servers
  - Apache users can use **mod\_negotiation** directives
  - IIS users use tools such as **PageXchanger** to manage the file extensions
- It is better if the file extensions are not used at all

# Module Flow

1

Network Scanning Concepts

4

Scanning Beyond IDS and Firewall

2

Scanning Tools

5

Banner Grabbing

3

Scanning Techniques

6

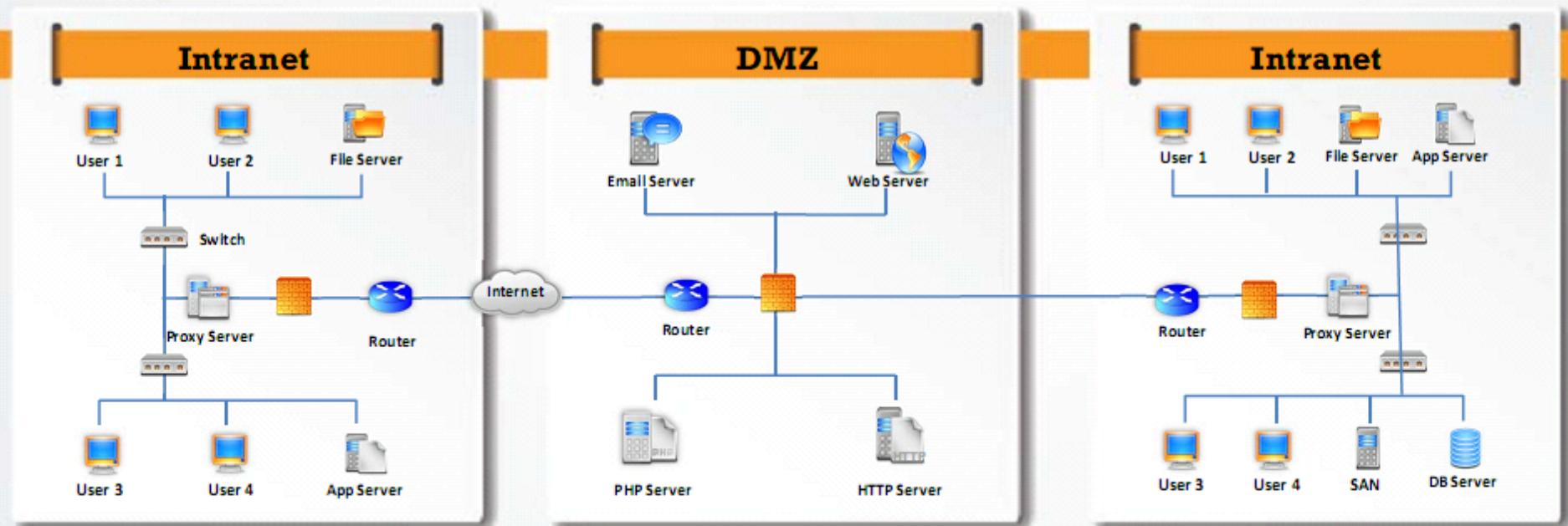
Draw Network Diagrams

7

Scanning Pen Testing

# Drawing Network Diagrams

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



Scanning Networks

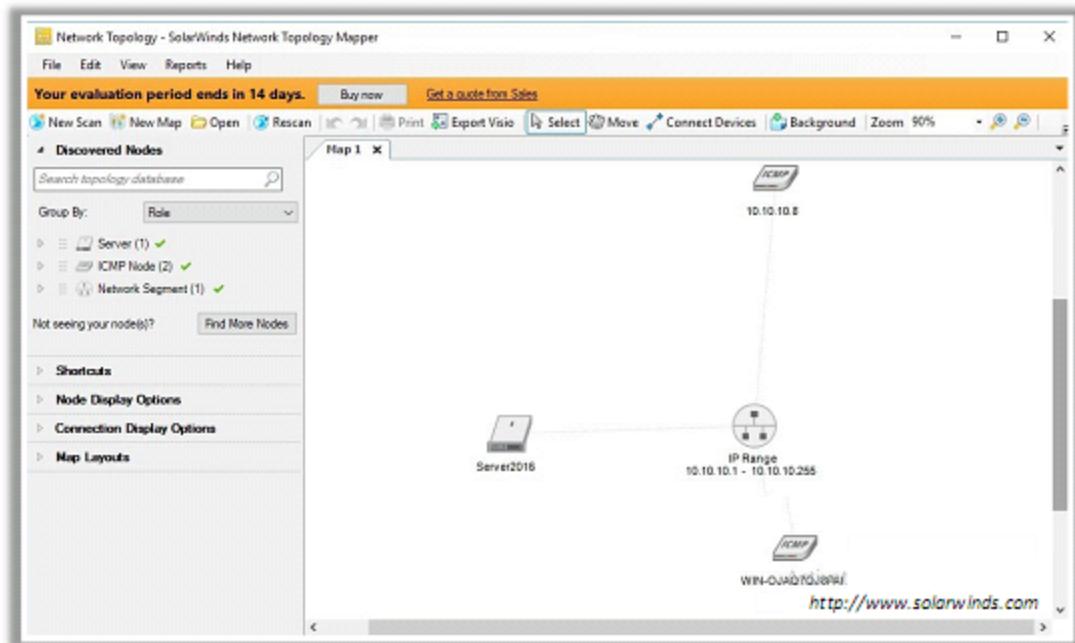
Draw Network Diagrams

# Network Discovery and Mapping Tools

C|EH  
Certified Ethical Hacker

## Network Topology Mapper

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



**OpManager**  
<https://www.manageengine.com>



**The Dude**  
<https://mikrotik.com>



**NetSurveyor**  
<http://nusaboutnets.com>



**NetBrain**  
<https://www.netbraintech.com>



**Spiceworks Inventory**  
<https://www.spiceworks.com>

# Network Discovery Tools for Mobile

## Scany



## Network "Swiss-Army-Knife"



**PortDroid Network Analysis**  
<https://play.google.com>



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



**Network Mapper**  
<https://play.google.com>



**Fing - Network Tools**  
<https://www.fing.io>



**ezNetScan**  
<https://play.google.com>

# Module Flow

1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Scanning Techniques**

4

**Scanning Beyond IDS and Firewall**

5

**Banner Grabbing**

6

**Draw Network Diagrams**

7

**Scanning Pen Testing**

# Scanning Pen Testing

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

Close **unused ports**



Disable **unnecessary services**



**Hide or customize banners**



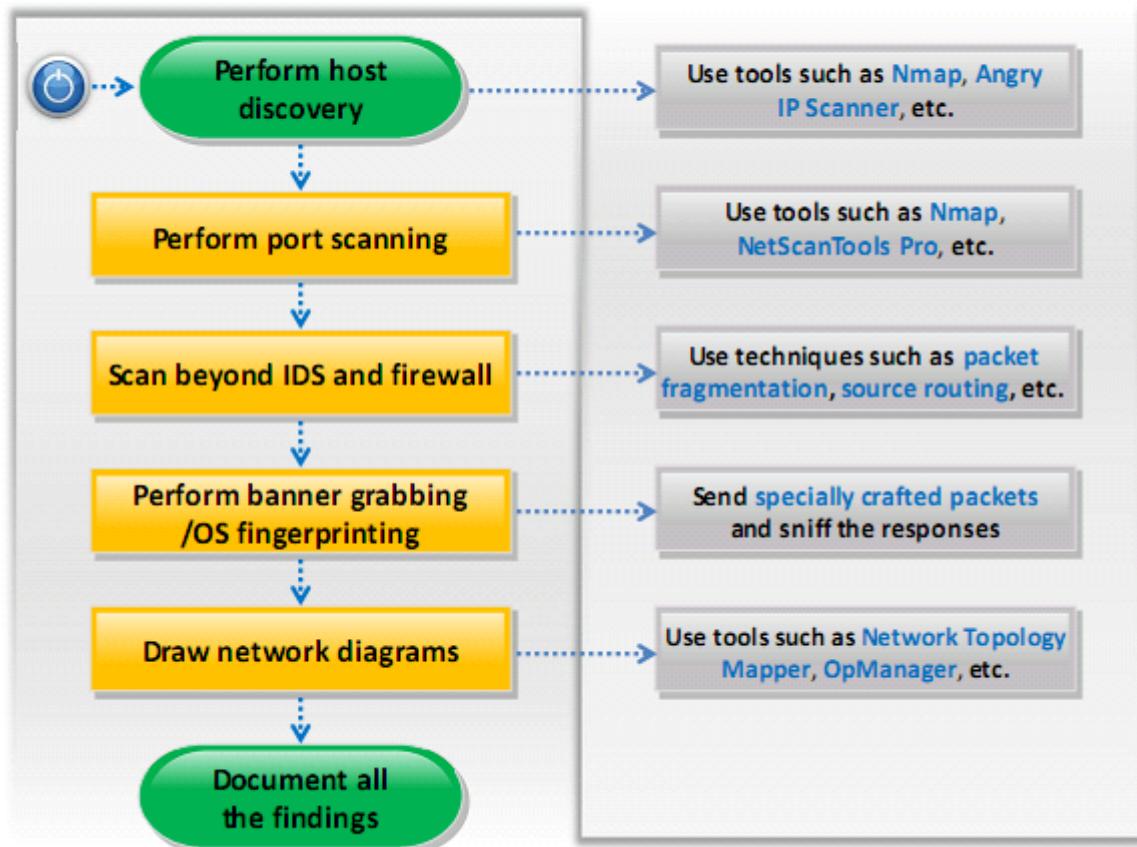
**Troubleshoot**  
service configuration errors



Calibrate **firewall rules**



# Scanning Pen Testing (Cont'd)



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

# Module Summary

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