

C|PENT

Certified Penetration Testing Professional

Module 07

Network Penetration Testing - Perimeter Devices

MODULE OBJECTIVE

Learn how to implement a comprehensive penetration testing methodology for assessing security of network perimeter devices, such as Firewalls, IDSes, Routers and Switches.

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

Module Objective

This module discusses how to implement a comprehensive penetration testing methodology for assessing security of network perimeter devices, such as Firewalls, IDSes, Routers and Switches.



Assessing Firewall Security Implementation

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

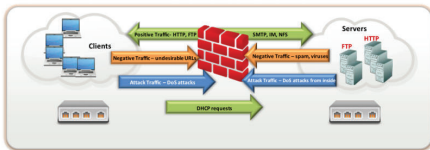
Assessing Firewall Security Implementation

A firewall is a software- or hardware-based system located at the network gateway that protects the resources of a private network from users on other networks. As a network penetration tester, it is essential for you to assess the security of the installed firewall in the organization.

Testing the Firewall from Both Sides



- Examine the firewall by simultaneously testing **both sides of the firewall**
- From **outside**, try to send packets to the firewall and check whether it allows them to pass against the configuration
- From **inside**, analyze the packets coming through the firewall and check whether the firewall allows them to pass against the configuration



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

Testing the Firewall from Both Sides (Cont'd)



The following are some of the activities to be performed for testing the firewall from outside:

- Identify the **firewall rules** by using appropriate **firewall tools** like firewalking
- Test whether unauthorized connections can be created to the **internal network from outside**
- Check for the reaction of the firewall to **fragmented and spoofed packets** that can be generated using a **packet generator**

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

Testing the Firewall from Both Sides (Cont'd)



The following are some of the activities to be performed for testing the firewall from **inside**:

- 1 Test (possibly using tunneled protocols) whether unauthorized connections from the internal network to the **Internet** can be created
- 2 Execute a **vulnerability scanner** on the hosts of the firewall system (i.e., firewall host, internal router, external router) from inside
- 3 Identify the **firewall rules** by using appropriate **firewall tools** (like firewalking from both sides)

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

Testing the Firewall from Both Sides

Examine the firewall by simultaneously testing from both sides of the firewall. In the test from outside, the penetration tester will try to send packets from outside of the firewall and check whether they pass through firewall against the firewall configuration. In the test from inside, penetration tester will analyze the packets that are coming through firewall and check whether the firewall allows packets to pass against the firewall configuration.

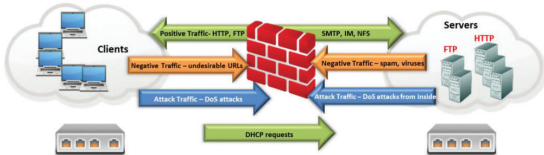


Figure 7.1: Example for testing the firewall from both sides

Following are some of the activities to be performed for testing the firewall from outside:

- Identify the firewall rules by using appropriate firewall tools such as firewalking.
- Test whether unauthorized connections can be created to the internal network from outside.
- Check for the reaction of the firewall to fragmented and spoofed packets that can be generated using a packet generator.

Following are some of the activities to be performed for testing the firewall from inside:

- Test (possibly using tunneled protocols) whether unauthorized connections from the internal network to the internet can be created.
- Execute a vulnerability scanner on the hosts of the firewall system (i.e., firewall host, internal router, external router) from inside.
- Identify the firewall rules by using appropriate firewall tools (such as firewalking from both sides).

Find Information about the Firewall



I

Check for information about the company's firewall in the **Engagement Letter/Pen Testing Contract** or search over **Internet** and **publicly available sources** (Newsgroup and blog postings made by employees, job postings, trade publications, etc.)

II

Send an email message to a non-existing user of the target organization and check **bounced mail header** for company's name server and topographic information

Bounced Mail Header

```
SMTP <Jhon@abc.com>
Please reply to Postmaster@123.abc.com if you feel this message to be in error.

Received: from 123.abc.com ([128.105.10.15]) by 123.abc.com

Netscape Mail Server v3.02) with ESMTD id BBB160 for <Jhon@abc.com>; Sun, 01
March 2015 13:57:40 -0600 Received: from abc.com (root@earth.abc.com
[128.164.10.16]) by abc.com (8.7.1/8.7.1) with ESMTD id CBB29812 for <
Jhon@abc.com>; Sun, 01 March 2015 13:57:40 -0600 (EST) Received: from
evil.mask.com (lucifer@evil [128.105.10.16]) by abc.com (8.7.1/8.7.1) with SMTP
id CBB29804 for <Jhon@abc.com>; Sun, 01 March 2015 13:57:40 -0600 (EST)
```

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

Find Information about the Firewall

To perform penetration test on any target firewall, it is important to find out the information about the firewall. Check the information about the company's firewall by searching over the internet and publicly available sources (newsgroup and blog postings made by employees, job postings, trade publications, etc.) or in the engagement letter/pen testing contract. You can use tools such as Nmap to collect additional information about the target firewall.

You can also send an email to a non-existing user of the target organization and check bounced mail header for company's name server and topographic information. An example of such bounced mail header is shown in the following screenshot.

Bounced Mail Header

```
SMTP <Jhon@abc.com>
Please reply to Postmaster@123.abc.com if you feel this message to be in error.

Received: from 123.abc.com ([128.105.10.15]) by 123.abc.com

Netscape Mail Server v3.02) with ESMTD id BBB160 for <Jhon@abc.com>; Sun, 01
March 2015 13:57:40 -0600 Received: from abc.com (root@earth.abc.com
[128.164.10.16]) by abc.com (8.7.1/8.7.1) with ESMTD id CBB29812 for <
Jhon@abc.com>; Sun, 01 March 2015 13:57:40 -0600 (EST) Received: from
evil.mask.com (lucifer@evil [128.105.10.16]) by abc.com (8.7.1/8.7.1) with SMTP
id CBB29804 for <Jhon@abc.com>; Sun, 01 March 2015 13:57:40 -0600 (EST)
```

Figure 7.2: Example for bounced SMTP mail header

Locate the Firewall by Conducting Traceroute



- Run a normal traceroute command to **find out the access control device, such as a firewall**
- "Request timed out" means the packet could not make it there and back. It is due to firewall or other security measures at the target system

```

root@kali:~# traceroute to certifiedhacker.com [192.168.236.11], 30 hops max, 60 byte packets
 0  192.168.0.2  (192.168.0.2)  0.304 ms  0.323 ms  0.202 ms
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  certifiedhacker.com [192.168.236.11]  0.304 ms  0.323 ms  0.202 ms

```

```

C:\Users\ADMIN\cmdtracert certifiedhacker.com
Tracing route to certifiedhacker.com [192.168.236.11]
over a maximum of 30 hops:
 0  <1 ms  <1 ms  <1 ms  [192.168.0.2]
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  certifiedhacker.com [192.168.236.11]  0.304 ms  0.323 ms  0.202 ms

```

Note: By default Windows tracert uses ICMP and both Mac OS X and Linux traceroute use UDP

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

Locate the Firewall by Conducting Traceroute

Most operating systems such as Windows, Unix, and so on come with a command-line utility called traceroute along with ping command. Traceroute traces the path or route through which the target host packets travel in the network. Tracert is the windows command-line utility for performing traceroute operation. By default, Windows tracert uses ICMP protocol, whereas Mac OS X and Linux traceroute use UDP protocol for performing the traceroute operation.

The Traceroute utility can detail the path IP packets travel between two systems. It can trace the number of routers the packets are traveling through, the round-trip time duration in transiting between two routers, and, if the routers have DNS entries, the names of the routers and their network affiliation. It can also trace the geographic location. It works by exploiting a feature of an internet protocol called time-to-live (TTL). The TTL field indicates the maximum number of routers a packet may transit. Each router that handles a packet decrements the TTL count field in the ICMP header by one. When the count reaches zero, the tool discards the packet and transmits an error message to the originator of the packet. The utility records the IP address and DNS name of that router, and sends out another packet with a TTL value of two. This packet makes it through the first router, then times-out at the next router in the path. This second router also sends an error message back to the originating host. Traceroute continues to do this, and records the IP address and name of each router until a packet finally reaches the target host or until it decides that the host is unreachable.

Finally, when it reaches the destination, the normal ICMP ping response will be sent to the sender. Thus, this utility helps to reveal the IP addresses of the intermediate hops in the route of the target host from the source.

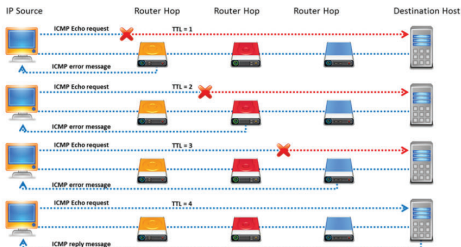


Figure 7.3: Schematic displaying process of ICMP request/response by traceroute

During the traceroute process, if you receive a message “Request timed out,” that means the packet could not make it there and back. This is due to firewall or other security measures at the target system.

The Traceroute utility helps to find the IP addresses of intermediate devices, such as routers, firewalls, and so on, present between a source and its destination. After running several traceroutes, you will be able to find the location of a hop in the target network.

Hence, by performing traceroute on the target, you can locate and identify the presence of any firewall established in the perimeter.

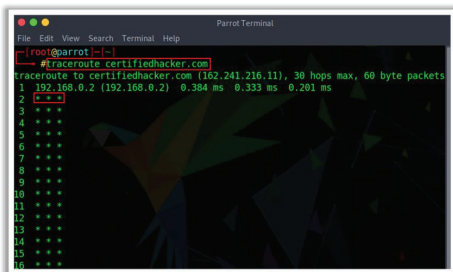
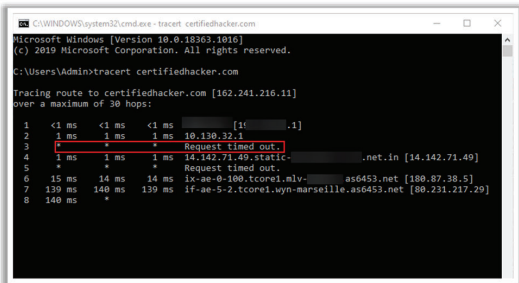


Figure 7.4: Screenshot displaying traceroute result on Linux machine—firewall blocking the packets



```
C:\WINDOWS\system32\cmd.exe - tracert certifiedhacker.com
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>tracert certifiedhacker.com

Tracing route to certifiedhacker.com [162.241.216.11]
over a maximum of 30 hops:

  0  <1 ms    <1 ms    <1 ms    [1' .1]
  1  1 ms     1 ms     1 ms     10.130.32.1
  2  *        *        *        Request timed out.
  3  *        *        *        Request timed out.
  4  1 ms     1 ms     1 ms     14.142.71.49.static-.net.in [14.142.71.49]
  5  *        *        *        Request timed out.
  6  15 ms    14 ms    14 ms    ix-ae-0-100.tcore1.mlv- as6453.net [180.87.38.5]
  7  139 ms   140 ms   139 ms   if-ae-5-2.tcore1.wyn-marseille.as6453.net [80.231.217.29]
  8  140 ms   *        *        *


```

Figure 7.5: Screenshot displaying traceroute result on Windows machine—firewall blocking the packets

Try to Pass through the Firewall using Hping



- Hping is tcp ping utility that allows you to pass through firewall even if they are blocked
- With the help of Hping, create **custom packets** to send towards the firewall. It will elicit unique responses from the **firewall**
- This can also be used to determine the **type of firewall**

- Craft a **SYN packet** using **Hping** or any other packet crafter and send it to the firewall
- Look for ICMP messages coming back from the firewall
- If you get an ICMP unreachable type 13 message (which is an admin prohibited packet) with a source IP address of the access control device, usually this is a **packet filter firewall**

Example 1:

```
[root@localhost]# hping 10.10.1.7 -c2 -S -p21 -n
# HPING 10.10.1.7 (eth0 10.10.1.1) : S set, 40 data bytes
60 bytes from 10.10.1.1: flags=SA seq=0 ttl=242 id=65121
win=64240
time=144.4 ms
```

Example 2:

```
[root@localhost]# hping 10.10.1.10 -c2 -S -p80 -n
#HPING 10.10.1.10 (eth0 10.10.1.1) : S set, 40 data bytes
ICMP Unreachable type 13 from 10.10.1.8
```

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

Try to Pass through the Firewall using Hping

Source: <http://www.hping.org>

Hping is a TCP ping utility that allows you to pass through a firewall even if they are blocked. With the help of Hping, you can create custom packets to send toward the firewall. It will elicit unique responses from the firewall. This can also be used to determine the type of firewall. Craft a SYN packet using Hping or any other packet crafter, and send it to the firewall. Look for ICMP messages coming back from the firewall. If you get an ICMP unreachable type 13 message (which is an admin-prohibited packet) with a source IP address of the access control device, usually this is a packet filter firewall.

■ Example 1:

In this scenario, where there is no presence of a packet filter firewall, let us consider crafting and sending 2 SYN packets (-c2 -s) to destination port 21 (-p21), considering numeric don't resolve option (-n) to target IP address (10.10.1.7), and analyze the result.

Command:

```
[root@localhost]# hping 10.10.1.7 -c2 -S -p21 -n
```

Result:

```
# HPING 10.10.1.7 (eth0 10.10.1.1) : S set, 40 data bytes
60 bytes from 10.10.1.1: flags=SA seq=0 ttl=242 id=65121
win=64240 time=144.4 ms
```

From this result, we can analyze that TCP port 21 on 10.10.1.7 is open, given the fact that we received a packet with an SA flag set, which is basically a SYN/ACK packet. This concludes that a port is open, but we do not know if it is the firewall or not.

▪ **Example 2:**

In this scenario, where there is a presence of a packet filter firewall, let us consider crafting and sending 2 SYN packets (-c2 -S) to destination port 80 (-p80), considering numeric don't resolve option (-n) to target (10.10.1.10), and analyze the result.

Command:

```
[root@localhost]# hping 10.10.1.10 -c2 -S -p80 -n
```

Result:

```
# HPING 10.10.1.10 (eth0 10.10.1.1) : S set, 40 data bytes
ICMP Unreachable type 13 from 10.10.1.8
```

This result, since you got the response as ICMP Unreachable type 13 from IP address 10.10.1.8, demonstrates that ICMP administrator prohibited filter packet. By this you can confirm that there is a firewall present at IP address 10.10.1.8 and it is blocking port 80 on 10.10.1.10.

Enumerate Firewall Access Control List using Nmap



- ➡ Most firewall **implementations** have default ports in use for **remote** management purposes, for example, user authentication, management, VPN connections, etc.
- ➡ Nmap can be helpful to find out loopholes in firewall implementation
- ➡ With Nmap, you can find state of ports open, filtered, unfiltered and closed on the firewall

Nmap shows three states of ports:

- **Open** – Port is **listening**
- **Filtered** – Port is blocked by an **access control device** (router/firewall)
- **Unfiltered** – Traffic is passing from access control devices (firewall/router), but the port is not open

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

Enumerate Firewall Access Control List using Nmap (Cont'd)



SYN stealth scan for enumeration of Access Control list

SYN stealth scan **through Firewall** for enumeration of Access Control list

[illegible]

The screenshot shows a Parrot Terminal window with the following content:

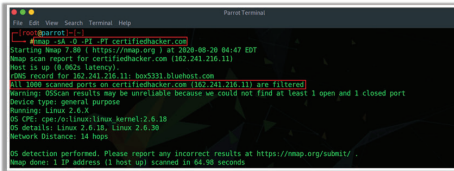
```
File Edit View Search Terminal Help
root@parrot:~# curl -s https://api.certnexus.com/v1/0
Starting Host 27071 https://certnexus.com/v1/0 2020-08-28 04:38 EDT
Map scan report for certnexus.com [192.241.216.11]
Host is up (0.019s latency).
SNI record for 162.241.216.11: bock3311.bluehost.com

PORT      STATE SERVICE
0/tcp    filtered http
```

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

Enumerate Firewall Access Control List using Nmap (Cont'd)

ACK scan for enumeration of Access Control list



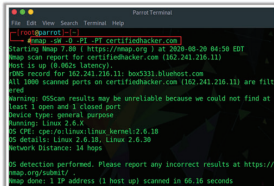
```
Parrot Terminal
File Edit View Search Terminal Help
~(root@parrot:~)
~$ nmap -sA -O -PI -PF certifiedhacker.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 04:47 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.002s latency).
DNS record for 162.241.216.11: box533l.bluehost.com
All 1000 scanned ports on certifiedhacker.com (162.241.216.11) are filtered
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Service type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux kernel:2.6.18
OS details: Linux 2.6.18, Linux 2.6.30
Network Distance: 14 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 64.98 seconds
```

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

Enumerate Firewall Access Control List using Nmap (Cont'd)

Win Packet Scan for Enumeration of Access Control list



```
Parrot Terminal
File Edit View Search Terminal Help
~(root@parrot:~)
~$ nmap -sW -O -PI -PF certifiedhacker.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 04:50 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.002s latency).
DNS record for 162.241.216.11: box533l.bluehost.com
All 1000 scanned ports on certifiedhacker.com (162.241.216.11) are filtered
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Service type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux kernel:2.6.18
OS details: Linux 2.6.30, Linux 2.6.30
Network Distance: 14 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 66.16 seconds
```

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

Enumerate Firewall Access Control List using Nmap

Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network. In the enumeration phase, you have to create active connections to the target system and perform directed queries to gain more information about the target. You can use the information collected by means of enumeration to identify the vulnerabilities or weak points in the system security, which helps you to exploit the target system. Enumeration techniques work in an intranet environment.

A firewall controls the traffic flow between the enterprise network, DMZ, and the internet. It primarily operates on security policy and uses Access Control Lists (ACLs; an implementation of a logic that can selectively permit or deny the traffic on an interface). A firewall examines the traffic and makes decision to what ensures that only permitted traffic will be allowed into the network, else it will be denied. In addition, you must understand the configuration of a firewall and the mapping it provides through the services and servers behind it.

Most firewall implementations have default ports in use for remote management purposes, for example, user authentication, management, VPN connections, and so on. You can use tools such as Nmap to find out loopholes in firewall implementation and conduct enumeration on the target firewall. With Nmap, you can also find state of ports open, filtered, unfiltered, and closed on the firewall.

Nmap shows three states of ports:

- Open—port is listening
- Filtered—port is blocked by an access control device (router/firewall)
- Unfiltered—traffic is passing from access control devices (firewall/router), but the port is not open

You can perform the following Nmap scanning techniques to enumerate the firewall Access Control Lists:

SYN stealth scan for enumeration of Access Control list

Syntax: `nmap -sS -O -PI -PT certifiedhacker.com`

This method uses a packet response to fingerprint the firewall. In the preceding syntax, `-sS` represents TCP SYN scan, `-O` enables OS detection and activates the remote host identification via TCP/IP fingerprinting, `-PI` represents true ping (ICMP echo request) packet where this option is used to find the hosts that are active and looks for subnet-directed broadcast address on the network, and `-PT` represents TCP ping request on target website (www.certifiedhacker.com).

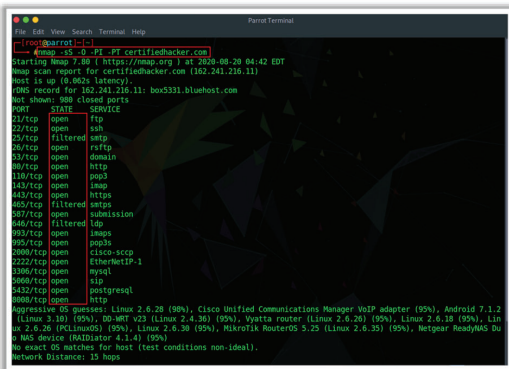


Figure 7.6: Screenshot displaying SYN stealth scan for enumeration of ACL

SYN stealth scan through Firewall for enumeration of Access Control list

Syntax: `nmap -sS -O -PI -PT certifiedhacker.com -p 80`

This scan determines the feasibility of SYN stealth scanning through the firewall for enumeration. In the preceding syntax, `-p` represents scan for only specified ports, that is, port number 80.

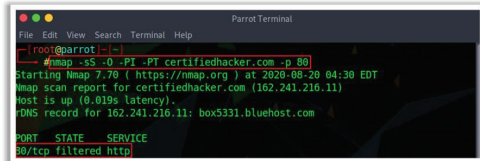


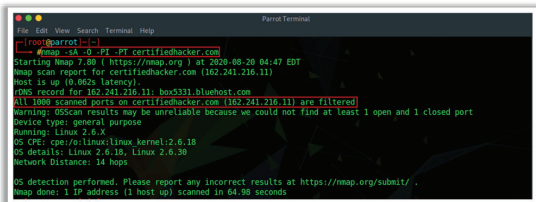
Figure 7.7: Screenshot displaying SYN stealth scan through firewall for enumeration of ACL

ACK scan for enumeration of Access Control list

Syntax: `nmap -sA -O -PI -PT certifiedhacker.com`

This scan is used to determine the firewall's ability to screen enumeration using the ACK packets. In the preceding syntax, **-sA** represents TCP ACK scan; this scan is different from other scan methods; it does not determine whether the ports are open or open|filtered. It is used to map out firewall rulesets, to determine whether it is stateful or not, and to find out which ports are filtered. ACK packet is sent along with random-looking acknowledgement/sequence numbers to the specified port. If an RST response comes then the port is classified as "unfiltered." If nothing is returned, then the port is classified as "filtered."

Note: Usually Nmap does not print "unfiltered" ports; that means all the probes got through and returned RSTs.



```
File Edit View Search Terminal Help
Parrot Terminal
[~]root@parrot[~]
-- [root@parrot:~]
-- nmap -sA -O -PI -PT certifiedhacker.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 04:47 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.062s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
All 1000 scanned ports on certifiedhacker.com (162.241.216.11) are filtered
Warning: OSscan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux kernel:2.6.18
OS details: Linux 2.6.18, Linux 2.6.30
Network Distance: 14 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 64.98 seconds
```

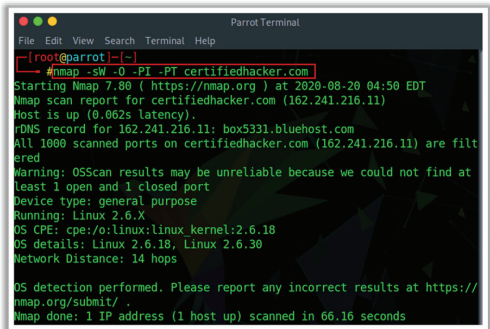
Figure 7.8: Screenshot displaying ACK scan for enumeration of ACL

Win Packet Scan for Enumeration of Access Control List

Syntax: **nmap -sW -O -PI -PT certifiedhacker.com**

This scan is used to determine the firewall's ability to screen enumeration using the WIN packets.

In the preceding syntax, **-sW** represents TCP Window Scan, this scan is exactly same as an ACK scan, except it sometimes detects open ports and filtered/nonfiltered. In addition, it always prints unfiltered when it receives an RST response.



```
Parrot Terminal
File Edit View Search Terminal Help
-[root@parrot]-[~]
-> nmap -sW -O -PI -PT certifiedhacker.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 04:50 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.062s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
All 1000 scanned ports on certifiedhacker.com (162.241.216.11) are filtered
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.18
OS details: Linux 2.6.18, Linux 2.6.30
Network Distance: 14 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.16 seconds
```

Figure 7.9: Screenshot displaying WIN scan for enumeration of ACL

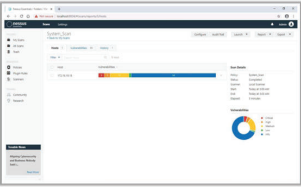
Scan the Firewall for Vulnerabilities



Run network vulnerability scanner, such as Nessus against your firewall to determine if there are any problems with it

Nessus includes a **whole category of vulnerability checks** just for firewalls

It can check the **firewall of a host and applications** vulnerabilities



Source: <https://www.tenable.com>

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

Scan the Firewall for Vulnerabilities

Setting up a firewall for any infrastructure is very important, and it is a great way of providing some basic security to the services. A small misconfiguration in a firewall can lead to vulnerabilities and can compromise the target system. Scanning the firewall plays a crucial role in identifying the vulnerabilities in firewall infrastructure. Run network vulnerability scanner such as Nessus against your firewall to determine if there are any problems with it. Nessus includes a whole category of vulnerability checks just for firewalls. It can check the firewall of an against host and applications vulnerabilities. A Nessus scanner can be placed behind the firewall to scan the large networks that are protected by a firewall. By default, the Nessus daemon listens to TCP port 1241. If you are connecting the scanner from the outside, then you need to connect to port 1241 on the real IP address of the deployed scanner.

- **Nessus**

Source: <https://www.tenable.com>

Nessus is a vulnerability, configuration, and compliance scanner. It features high-speed discovery, configuration auditing, asset profiling, malware detection, sensitive-data discovery, patch management integration, and vulnerability analysis of a wireless network.

Features:

- Provides the real-time values, and there is no need to wait for the scanning to be completed to view the results.
- Provides the same user interface for all operating systems, including Mac, Windows, and Linux.

- Continues scanning in the server even if the UI is disconnected.
- Provides a scan template that creates scan policies for auditing the network.

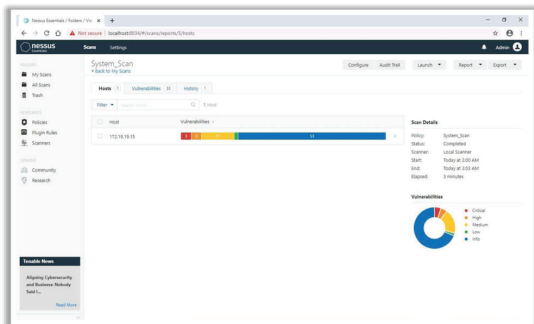


Figure 7.10: Screenshot of Nessus Vulnerability Scanner

Map Firewall Make and Version with Associated Vulnerabilities



- Firewalls have specific **vulnerabilities**
- If a firewall is not patched up, then it is **vulnerable to attacks**
- Search the security vulnerabilities associated with specific version, make, model of firewall on the Internet and find out related exploits in Google hacking databases
- Send product-specific exploits against **firewall vulnerabilities** and test for responses

Google

cisco asa 5520 firewall vulnerabilities

about 1,100 results (0.42 seconds)

All - www.cisco.com - Cisco Small Business Security - Next-Gen...
Cisco ASA 5520 - List of security vulnerabilities
Security vulnerabilities of Cisco ASA 5520 - List of all known CVEs, security advisories and links to full CVE details
www.cisco.com - products - Cisco ASA 5520 - Cisco ASA 5520 - CVE security vulnerabilities
Cisco ASA 5520 security vulnerabilities - Cisco ASA 5520 and list of versions

CVE Details

Search + Details Security Vulnerabilities

Search for a CVE by ID, name, or keyword. You can also search by product, vendor, or CVSS score. Results are sorted by relevance.

CVE ID	Product	Severity	CVSS	Exploit	Platform
CVE-2017-15354	Cisco ASA 5520	High	9.8	Yes	Windows
CVE-2017-15353	Cisco ASA 5520	High	9.8	Yes	Windows
CVE-2017-15352	Cisco ASA 5520	High	9.8	Yes	Windows
CVE-2017-15351	Cisco ASA 5520	High	9.8	Yes	Windows
CVE-2017-15350	Cisco ASA 5520	High	9.8	Yes	Windows

Cisco ASA 8.x - EXTRABACON Authentication Bypass

CVE	Author	Type	Platform	Date
CVE-2017-15354	Exploit-DB	Exploit	Windows	2017-08-14

Exploit: 1 / 1

Vulnerable App:

Exploit Title: Cisco ASA 8.x Authentication Bypass

Author: 0x00sec

Product: Cisco ASA 8.x

Vulnerable App: Cisco ASA

Vulnerable Version: 8.4

Vulnerable Vendor: Cisco

CVE ID: CVE-2017-15354

CVSS: 9.8 (critical) using NVD

Exploit: 1 / 1

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

Map Firewall Make and Version with Associated vulnerabilities

In this step, you need to search for firewall-specific vulnerabilities. If a firewall is not properly patched, then it is vulnerable to various attacks. You can identify the product-specific vulnerabilities manually by searching the specific version, make, model of firewall on the internet or in the CVE details and find out related exploits in Google hacking databases.

CVE details is a free CVE security vulnerability database that is taken from National Vulnerability Database xml feeds provided by National Institute of Standards and Technology. You can view vulnerability details, references, exploits, Metasploit modules, CVSS score reports, full list of vulnerable products, and vulnerability trends. You can directly go to the CVE details website and search for any product-specific vulnerabilities, for example, Cisco ASA 5520. The details of bugs can be displayed as shown in the screenshots.



Cisco ASA 8.x - 'EXTRABACON' Authentication Bypass

EDB-ID: 40236	CVE: 2018-6386	Author: SHADOW BROKERS	Type: REMOTE	Platform: m: HARDWARE	Date: 2018-09-18
EDB Verified: ✖		Exploit: 📄 / {}		Vulnerable App:	

← →

```
# Exploit Title: Cisco ASA 8.X Authentication Bypass
# Date: 17-08-2018
# Exploit Author: Equation Group
# Vendor Homepage: Cisco
# Software Link: Cisco
# Version: Cisco ASA 8.X
# Tested on: Cisco ASA 8.4.2
# CVE : Not sure

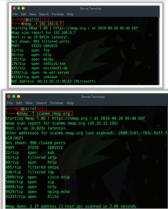
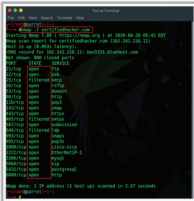
Requirements:
* SNMP read (public) string
* Access to SNMP service
```

Figure 7.13: Screenshot displaying Exploit database search result for Cisco ASA 5520 firewall vulnerabilities

Try to Bypass the Firewall using Fragmented Packets



- Use **nmap -f option** to send the fragmented packets; this may result in bypassing the packet inspection at the firewalls

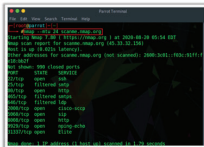
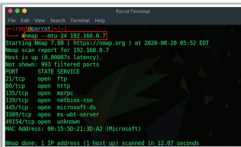


Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Try to Bypass the Firewall using Fragmented Packets (Cont'd)



- You can also send fragmented packets by setting a **specific MTU** (Maximum Transmission Unit) in your Nmap scan
- Use **nmap --mtu [packet size]** to create packets with size based on the number that we will give in [packet size]
- MTU number that you specify must be a multiple of 8 (8,16,24,32 etc.)



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

Try to Bypass the Firewall Using Fragmented Packets

During penetration test, you may come across the systems that are behind the firewalls and may be unable to get the information you want. As a tester, you must know how to avoid the firewall rules that are in place, and gather the information about the host. You have to apply various methods in order to bypass a firewall. Using fragmented packets can sometimes allow you to bypass the firewall and gather the information that is beyond the firewall. Fragmented packets are used when the data of the network layer is too big to be transmitted over the data

link layer in a single packet. The data of the network layer is split into several pieces (fragments) and transferred over the network.

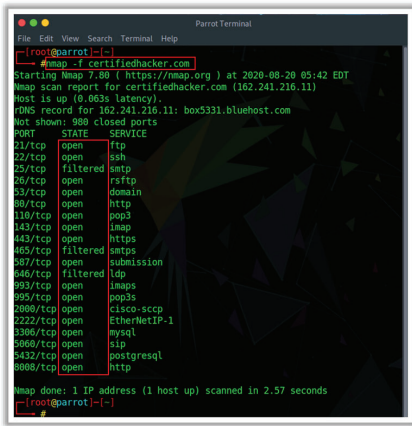
- **Nmap**

Source: <https://nmap.org>

You can use Nmap tool to send the fragmented the packets, thus bypass the packet inspection of firewalls.

Syntax: `nmap -f certifiedhacker.com`

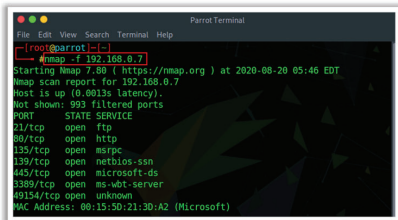
`-f` represents fragment packets. This option requests the scan to use tiny fragmented IP packets. It divides the TCP header into several packets, which makes it harder for intrusion-detection systems, packet filters, and annoyances to detect what actually the tester is doing. When the option is specified once, it will split the packet into 8 bytes or less after the IP header. Hence, for 20 bytes of TCP header, it will be split into three packets, for 8 bytes it will split into two and for 4 bytes it will splits into one. If `“-f”` option is specified again, it uses 16 bytes per fragment. An example of the fragmented packets command along with the result is displayed in the following screenshots.



```
Parrot Terminal
File Edit View Search Terminal Help
[~root@parrot]-[~]
# nmap -f certifiedhacker.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 05:42 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.063s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 980 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    filtered smtp
26/tcp    open  rsftp
53/tcp    open  domain
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
443/tcp   open  https
465/tcp   filtered smtps
587/tcp   open  submission
646/tcp   filtered ldap
993/tcp   open  imaps
995/tcp   open  pop3s
2000/tcp  open  cisco-sccp
2222/tcp  open  EtherNet/IP-1
3306/tcp  open  mysql
5060/tcp  open  sip
5432/tcp  open  postgresql
8008/tcp  open  http

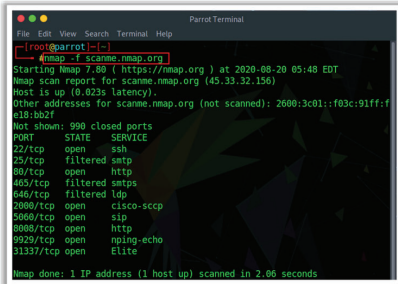
Nmap done: 1 IP address (1 host up) scanned in 2.57 seconds
[~root@parrot]-[~]
#
```

Figure 7.14: Screenshot displaying Nmap -f scan result for certifiedhacker.com



```
Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot:[~]
[~]nmap -f 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 05:46 EDT
Nmap scan report for 192.168.0.7
Host is up (0.0013s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49154/tcp open  unknown
MAC Address: 08:15:50:21:30:A2 (Microsoft)
```

Figure 7.15: Screenshot displaying Nmap -f scan result for local machine



```
Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot:[~]
[~]nmap -f scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 05:48 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.023s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:f
e18:bb2f
Not shown: 990 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    filtered smtp
80/tcp    open  http
465/tcp   filtered smtps
646/tcp   filtered ldp
2000/tcp  open  cisco-sccp
5060/tcp  open  sip
8008/tcp  open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite
Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
```

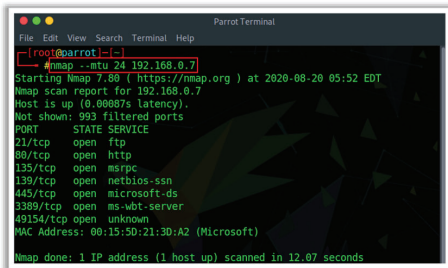
Figure 7.16: Screenshot displaying Nmap -f scan result for scanme.nmap.org

You can also send fragmented packets by setting a specific MTU (maximum transmission unit) in your nmap scan.

Syntax: **nmap --mtu [packet size] [target]**

Use **--mtu** option to create packets with size based on the number that we will give in [packet size]. MTU number that you specify must be a multiple of 8 (8, 16, 24, 32, etc.)

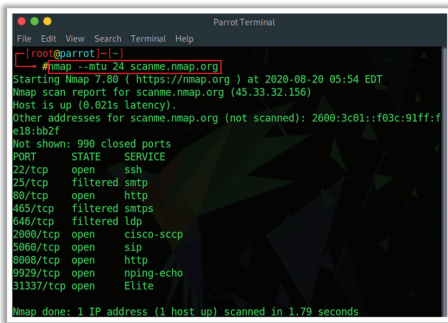
Examples of the mtu-based fragment packets along with the results are displayed in the following screenshots.



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot:~]
#nmap --mtu 24 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 05:52 EDT
Nmap scan report for 192.168.0.7
Host is up (0.00087s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49154/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 12.07 seconds
```

Figure 7.17: Screenshot displaying Nmap --mtu 24 scan result for local machine



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot:~]
#nmap --mtu 24 scanme.nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 05:54 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.021s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:f
e18:bb2f
Not shown: 990 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    filtered smtp
80/tcp    open  http
465/tcp   filtered smtps
646/tcp   filtered ldaps
2000/tcp  open  cisco-sccp
5060/tcp  open  sip
8008/tcp  open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
```

Figure 7.18: Screenshot displaying Nmap --mtu 24 scan result for scanme.nmap.org

Try to Bypass the Firewall by Spoofing Packets



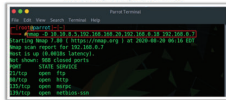
- Instruct **Nmap** to spoof packets from other hosts, making it hard for firewall to determine from which system the scan actually started

- Use following nmap commands:

- **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)

Note: The decoys you specify should be online in order for this technique to work. You might accidentally SYN flood your target if the decoys are offline and it will also be easy for the target to determine which host is scanning.

Traffic Originating from Decoys



No.	Time	Source	Destination	Protocol	Length	Info
8100	11.21.01.01	192.168.0.7	192.168.100.10	TCP	60	54383 → 51581 [RST, ACK]
8107	11.21.01.02	192.168.0.7	192.168.100.10	TCP	60	54386 → 51581 [RST, ACK]
8109	11.21.03.77	10.0.0.8	192.168.0.7	TCP	60	54383 → 541 [RST, SEQ]
8100	11.21.03.77	192.168.0.14	192.168.0.7	TCP	60	54383 → 541 [RST, SEQ]
8100	11.21.03.77	192.168.0.10	192.168.0.7	TCP	60	54383 → 541 [RST, SEQ]
8100	11.21.03.12	192.168.0.7	10.0.0.8	TCP	60	54381 → 51581 [RST, ACK]
8102	11.21.03.96	192.168.0.7	192.168.0.10	TCP	60	54381 → 51581 [RST, ACK]
8100	11.21.04.00	192.168.0.7	192.168.100.10	TCP	60	54381 → 51581 [RST, ACK]
8100	11.21.04.00	192.168.0.18	192.168.0.7	TCP	60	54381 → 51581 [RST, ACK]
8100	11.21.04.04	192.168.0.7	192.168.0.18	TCP	60	54381 → 51581 [RST, ACK]
8100	11.57.00.00	192.168.0.7	230.255.255.250	SIGP	210	*SIGNAL* & HTTP/3.1

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

Try to Bypass the Firewall by Spoofing Packets

If you want to hide your identity and avoid the detection while performing the scan with Nmap, then you can use Decoy scan option in the Nmap. It scans the target with multiple fake and spoofed IP addresses, making it literally undetectable for the victim to identify the actual source of packets. This way the firewall can be easily bypassed as it cannot detect the scan and thinks it is coming from multiple resources or IP addresses. The decoys, you specified must be online so that this technique can work. With this decoy option, it appears to the remote host that the host(s) you specify as decoys are scanning the target network too. Thus, the target firewall might report 5–10 port scans from unique IP address, but they cannot identify the particular IP address from which the IP scanning is performed.

Syntax 1: `nmap -D <decoy1>, <decoy2>, <decoy3> ME` (Manually specify the IP addresses of the decoys)

Each decoy host is separated with commas, and option “**ME**” can be placed optionally as one of the decoys to represent the real IP address.

Syntax 2: `nmap -D RND: 10 [target]` (Generates a random number of decoys)

If "ME" is not used, then Nmap will put the real IP address randomly or with "RND" option, and a non-reserved IP address can be generated, RND:<number> to generate <number> addresses.

Decoys are used both at initial and actual port scanning phases. They are also used during the OS detection. Ensure that you use limited number of decoys since usage of too many decoys may result in slowing down the scan and potentially make the result less accurate.

Note: The decoys you specify should be online in order for this technique to work. You might accidentally SYN flood your target if the decoys are offline and it will also be easy for the target to determine which host is scanning.

The decoy scan and its respective captured logs are displayed in the following screenshots:

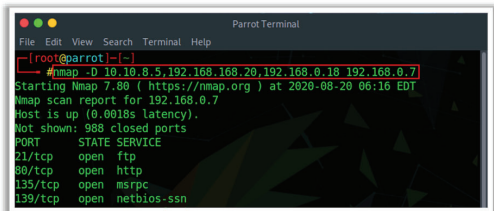


Figure 7.19: Screenshot displaying Nmap decoy scan with multiple decoys on target

Traffic Originating from Decoys

No.	Time	Source	Destination	Protocol	Length	Info
8196	11.283671	192.168.0.7	192.168.168.20	TCP	54	9666 → 58388 [RST, ACK]
8197	11.283674	192.168.0.7	192.168.0.18	TCP	54	9666 → 58388 [RST, ACK]
8198	11.283777	10.10.8.5	192.168.0.7	TCP	58	58388 → 981 [SYN] Seq=
8199	11.283777	192.168.0.18	192.168.0.7	TCP	58	58388 → 981 [SYN] Seq=
8200	11.283777	192.168.168.20	192.168.0.7	TCP	58	58388 → 981 [SYN] Seq=
8201	11.283981	192.168.0.7	10.10.8.5	TCP	54	981 → 58388 [RST, ACK]
8202	11.283986	192.168.0.7	192.168.0.18	TCP	54	981 → 58388 [RST, ACK]
8203	11.288400	192.168.0.7	192.168.168.20	TCP	54	981 → 58388 [RST, ACK]
8204	11.288404	192.168.0.18	192.168.0.7	TCP	58	[TCP Retransmission] Seq=
8205	11.288469	192.168.0.7	192.168.0.18	TCP	54	981 → 58388 [RST, ACK]
8206	11.576098	192.168.0.7	239.255.255.250	SSDP	216	Ⓜ→SEARCH Ⓜ HTTP/1.1

Figure 7.20: Screenshot displaying Wireshark result capturing scan on target from multiple decoys

Try to Bypass the Firewall by Spoofed Source Port



If the firewall is configured to allow all incoming traffic through a specific port number, you can exploit this configuration by **sending Nmap scan request** with specific source port number

- `Nmap --source-port 53 [target]`
- `Nmap -g 53 [target]`

```

Terminal
File Edit View Search Terminal Help
root@kali:~# nmap -sP 192.168.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 00:21 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0000s latency).
Not shown: 599 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
4880/tcp  open  RDP
5132/tcp  open  unknown
5133/tcp  open  unknown
5134/tcp  open  unknown
5135/tcp  open  unknown
5136/tcp  open  unknown
5137/tcp  open  unknown
MAC Address: 08:00:27:1B:42 (Microsoft)

Terminal
File Edit View Search Terminal Help
root@kali:~# nmap -sP 192.168.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 00:20 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0000s latency).
Not shown: 599 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
4880/tcp  open  RDP
5132/tcp  open  unknown
5133/tcp  open  unknown
5134/tcp  open  unknown
5135/tcp  open  unknown
5136/tcp  open  unknown
5137/tcp  open  unknown
MAC Address: 08:00:27:1B:42 (Microsoft)
    
```

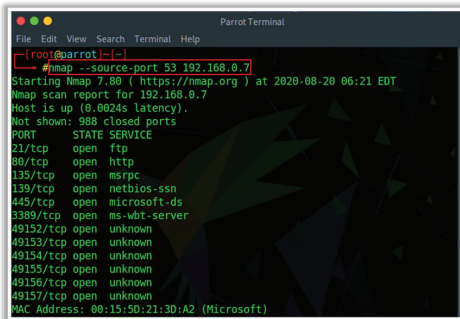
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Try to Bypass the Firewall by Spoofed Source Port

Most of the administrators commit the mistake by setting up firewall rule to allow all incoming traffic from specific common ports such as 20 (FTP), 53 (DNS), and 67 (DHCP). To probe the target firewall, Nmap will randomly select an available outgoing source port, whereas `--source-port` option allows the Nmap tool to choose the specified port as the source for all packets. This option allows us to identify the weakness in the firewall that are configured improperly, or those that blindly accept incoming traffic based on a specified port number.

The following commands are used in Nmap to bypass firewall by spoofing the source port:

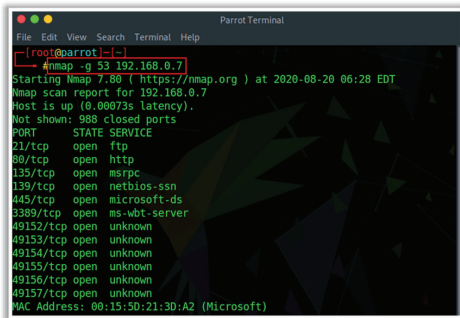
Syntax: `Nmap --source-port 53 [target]`



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot:~]
#nmap --source-port 53 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 06:21 EDT
Nmap scan report for 192.168.0.7
Host is up (0.0024s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)
```

Figure 7.21: Screenshot displaying result of --source-port scan for port 53

Syntax: **Nmap -g 53 [target]**



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot:~]
#nmap -g 53 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 06:28 EDT
Nmap scan report for 192.168.0.7
Host is up (0.00073s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)
```

Figure 7.22: Screenshot displaying result of -g scan for port 53

Try to Bypass Firewall by MAC Address Spoofing



- When the firewall allows all the traffic based on the MAC address, you can exploit this firewall configuration by spoofing mac address
- Use **nmap --spoof-mac** option to spoof mac address of specific vendors or generate random mac address or own MAC address
 - Specify MAC address from a Vendor → **--spoof-mac Dell/Apple/3Com**
 - Generate a random MAC address → **--spoof-mac 0**
 - Specify your own MAC address → **--spoof-mac 00:01:02:25:56:AE**

```

root@kali:~# nmap -sT -Pn --spoof-mac Dell 192.168.0.7
Starting Nmap 7.80 (https://nmap.org) at 2025-08-28 06:31 EDT
Spoofing MAC address 08:00:97:0A:3A:80 (Dell EMC)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 192.168.0.7
Host is up (0.002s latency).
Not shown: 593 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
59152/tcp open  unknown
60153/tcp open  unknown
60154/tcp open  unknown
60155/tcp open  unknown
60156/tcp open  unknown
60157/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
root@kali:~#
    
```

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

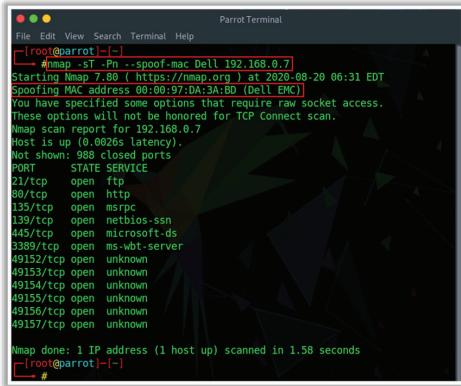
Try to Bypass Firewall by MAC Address Spoofing

MAC address spoofing is a simple identity-hiding technique, but it creates a difficult scenario for the victim to identify the host from where the incoming requests are originated. This technique will be more useful when there is a MAC filtering rule to filter traffic from certain MAC addresses, so you need to discover from which MAC address you need to obtain the results. When the firewall allows all the traffic based on the MAC address, you can exploit this firewall configuration by spoofing mac address.

With the help of the Nmap tool you can spoof the MAC address by using the option “**--spoof-mac**” It will choose a MAC address from a specific vendor. Then on the basis of the vendor name, Nmap will choose completely random MAC address for every scan. Another way is to manually specify the MAC address, so that you can spoof the address of a computer within the same network. Nmap maintains the nmap-mac-prefixes database. Whenever we specify the vendor name, Nmap will consider that database to find the suitable MAC address.

Following are different ways of performing MAC address spoofing commands:

- Specify MAC address from a vendor → **--spoof-mac Dell/Apple/3Com**
- Generate a random MAC address → **--spoof-mac 0**
- Specify your own MAC address → **--spoof-mac 00:01:02:25:56:AE**



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~#
[root@parrot]~# nmap -sT -Pn --spoof-mac Dell 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 06:31 EDT
Spoofing MAC address 00:00:97:DA:3A:BD (Dell EMC)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 192.168.0.7
Host is up (0.0026s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
[root@parrot]~#
```

Figure 7.23: Screenshot displaying spoofed MAC address scan using Nmap

Try to Bypass the Firewall by IP Address Spoofing



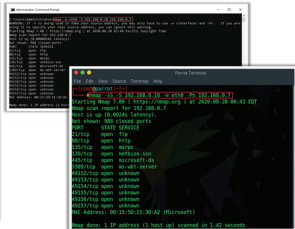
- When the firewall allows all the traffic based on the IP address, you can exploit this firewall configuration by spoofing IP address
- Use `nmap -S <spoofed ip> -e [interface]` option to spoof IP address to go undetected

For Example,

```
nmap -e eth0 -S 192.168.1.100
192.168.1.109
```

```
-e = interface
-S = Specify source (+ address)
```

- It will use the eth0 interface and spoof a source IP of 192.168.1.100 while scanning 192.168.1.109



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

Try to Bypass the Firewall by IP Address Spoofing

Most of the firewalls filter the packets based on the source IP address. These firewalls examine the source IP address and decide whether the packet is coming from legitimate whitelisted source or illegitimate source. When the firewall allows all the traffic based on the IP address, you can exploit this firewall configuration by spoofing the IP address. To spoof the IP address, you can use the Nmap tool.

IP address spoofing or IP spoofing is a technique in which you can make use of someone's IP address to hide their identity. In IP spoofing, you can create IP packets by using a forged IP address and gain access over the system or network without authorization.

In some cases, the Nmap tool may not determine the source address, in such scenario use **“-s”** option with the IP address of an interface through which you wish to send the packets. Another usage of this flag is to spoof the scan to make the victim think that someone else is scanning them. For this kind of scan options **“-e”** and **“-pn”** are generally used.

Option **-e** states which interface needs to send and receive packets. Here the Nmap tool should either automatically detect this or tell that it cannot detect it.

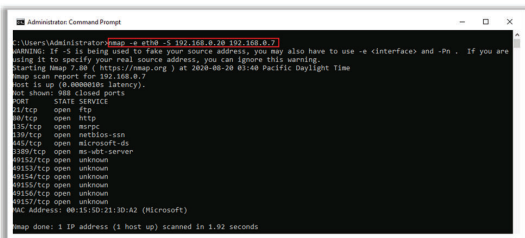
Note: Nmap does not produce a useful report, because the replay packets won't be received since they are addressed to the spoofed IP address.

Syntax: `nmap -S <spoofed ip> -e [interface]` (use this option to spoof IP address to go undetected)

Example:

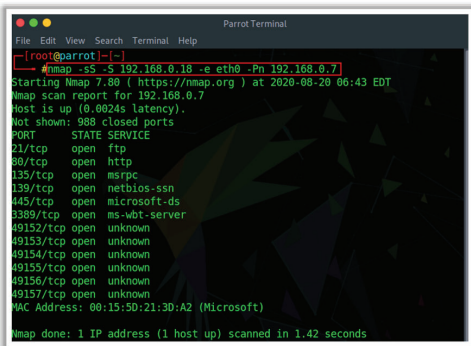
```
nmap -e eth0 -S 192.168.1.100 192.168.1.109
```


where `-e` specifies interface and `-S` specifies source IP address. It will use the `eth0` interface and spoof a source IP of `192.168.1.100`, while scanning `192.168.1.109`.



```
Administrator: Command Prompt
C:\Users\Administrator> nmap -e eth0 -S 192.168.0.20 192.168.0.7
WARNING: If -S is being used to fake your source address, you may also have to use -e <interface> and -Pn . If you are
using it to specify your real source address, you can ignore this warning.
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 03:40 Pacific Daylight Time
Nmap scan report for 192.168.0.7
Host is up (0.0000010s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 1.92 seconds
```

Figure 7.24: Screenshot displaying spoofed IP address scan using Nmap on windows machine



```
Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~# nmap -sS -S 192.168.0.18 -e eth0 -Pn 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 06:43 EDT
Nmap scan report for 192.168.0.7
Host is up (0.0024s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds
```

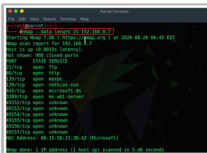
Figure 7.25: Screenshot displaying spoofed IP address scan using Nmap on Linux machine

Try to Bypass the Firewall by Varying Packet Size



Many firewalls are configured to **detect port scan** attempts by **inspecting** size of packets as the ports scanners, generally, send packets that have a specific size

To avoid detection of your port scan attempt, use **nmap --data-length** option to send the packets with different size than the default value



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

Try to Bypass the Firewall by Varying Packet Size

Most of the firewalls/IDSs either block or log the packets by looking at their data length to identify the potential port scan. This is because most of the scanners send the packets with default data length from Nmap. To avoid such detection, the better way to send nmap packets is to set the data length option "**--data-length,**" in order to bypass filters and evade IDS systems. The value of data length for the packets can be specified in the arguments as **--data-length 25** as shown in the following screenshot.

Assume that, nmap sends packet to the target that is 58 bytes and the following screenshot shows the captured packets with data length 58.

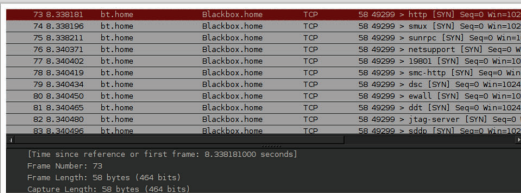


Figure 7.26: Screenshot displaying captured data length on target machine using Wireshark

In order to change the above default value, use the command:

nmap --data-length 25 [target ip address] as shown in the following screenshot.

```

Parrot Terminal
File Edit View Search Terminal Help
[~root@parrot]~
# nmap --data-length 25 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 06:45 EDT
Nmap scan report for 192.168.0.7
Host is up (0.0015s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:15:5D:21:3D:A2 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 5.46 seconds
  
```

Figure 7.27: Screenshot displaying usage of data length scan in Nmap

Now, recapture the packets on the target machine and you can see that the packet data length is changed to the value 83 as shown in the following screenshot. This method can avoid being discovered by firewalls which will check for the default packet size that nmap generates.

No.	Time	Source	Destination	Protocol	Length	Info
2007	24.367124	bt.home	Blackbox.home	TCP	83	44474 → 1ad2 [SYN] Seq=0 Win=1024
2008	24.371390	bt.home	Blackbox.home	ISAKMP	83	[Malformed Packet]
2009	24.373446	bt.home	Blackbox.home	TCP	83	44474 → msft-gc-ssl [SYN] Seq=0 Win=1024
2010	24.373459	bt.home	Blackbox.home	TCP	83	44474 → apple-sasl [SYN] Seq=0 Win=1024
2011	24.373466	bt.home	Blackbox.home	TCP	83	44474 → mcads [SYN] Seq=0 Win=1024
2012	24.373472	bt.home	Blackbox.home	TCP	83	44474 → 7512 [SYN] Seq=0 Win=1024
2013	24.373478	bt.home	Blackbox.home	TCP	83	44474 → excv [SYN] Seq=0 Win=1024
2014	24.373484	bt.home	Blackbox.home	TCP	83	44474 → msn-sql [SYN] Seq=0 Win=1024
2015	24.375592	bt.home	Blackbox.home	TCP	83	44474 → x11-3 [SYN] Seq=0 Win=1024
2016	24.375606	bt.home	Blackbox.home	TCP	83	44474 → eppc [SYN] Seq=0 Win=1024
2017	24.375613	bt.home	Blackbox.home	OMAPI	83	Unknown opcode (0x9c1c4e4d)[Malformed Packet]

Frame 2007: 83 bytes captured on interface eth0, Time 24.367124000 seconds	
Time since reference or first frame:	24.367124000 seconds
Frame Number:	2007
Frame Length:	83 bytes (664 bits)
Capture Length:	83 bytes (664 bits)

Figure 7.28: Screenshot displaying captured data length on target machine using Wireshark

Try to Bypass the Firewall by Sending Bad Checksums



Send the packets with **bad checksum** in an attempt to avoid firewall detection. This will **reveal** information from systems when it is not properly configured



If you don't get any information, it indicates that the **system** is well **configured**

```
Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot[~]
# nmap --badsum 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 07:19 EDT
Nmap scan report for 192.168.0.7
Host is up (0.00055s latency).
All 1000 scanned ports on 192.168.0.7 are filtered
MAC Address: 00:15:5D:29:A0:2B (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 21.25 seconds
```

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

Try to Bypass the Firewall by Sending Bad Checksums

Checksum is used to check the integrity of the data. It will check the packet's checksum against the known record ensuring the integrity of the content is correct and undamaged. Bad checksum is used to ensure network security and to evade firewalls; however, only poorly designed systems will give a response to a packet with a bad checksum. A bad checksum error is the result when the data is damaged. You can send packets with an incorrect checksum to discover whether the information in the system is properly configured or not and whether you are able to avoid firewall detection. The command "**--badsum**" is used to send packets with bad checksums. If you don't get any result by using this command, then the target system is well configured.

Syntax: **nmap --badsum [target]**

```
Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot[~]
# nmap --badsum 192.168.0.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 07:19 EDT
Nmap scan report for 192.168.0.7
Host is up (0.00055s latency).
All 1000 scanned ports on 192.168.0.7 are filtered
MAC Address: 00:15:5D:29:A0:2B (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 21.25 seconds
```

Figure 7.29: Screenshot displaying Nmap badsum scan result

Try to Bypass the Firewall using Port Redirection



- If you cannot get direct access to a port, then use **port redirection**
- It is used to **bypass** port filtering
- Install a **port redirector** and make it listen on a selected port number
- Packets received on the **listening port number** are forwarded to the desired port on the remote host
- Tools:
 - `fpipe -l 80 -r 139 192.168.10.40`
 - `datapipe 80 139 192.168.10.40`

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

Try to Bypass the Firewall using Port Redirection

When a port cannot be accessed directly, port redirection helps gain access to those ports. Port redirection is often used to bypass port filtering. You can compromise a target system that can help bypass the firewall and then exploit it with the use of port redirection. Once the port redirector is installed, it listens to certain port numbers and redirects all packets received on the listening port to a remote host.

The following tools can be used to perform port redirection:

- **Fpipe**

Source: <https://fpipe.apponic.com>

Fpipe is a source port forwarder/redirector. It can create a TCP or UDP stream with a source port of your choice. This is useful for getting past firewalls that allow traffic with source ports of say 23, to connect with internal servers.

Syntax: **fpipe -l 80 -r 139 192.168.10.40**

- **Datapipe**

Source: <https://github.com>

Datapipe is one of the most popular redirectors that work on Unix-based platforms. It is very time-consuming, because it must run on both the attacker's system and the compromised host.

Syntax: **datapipe 80 139 192.168.10.40**

Try to Bypass the Firewall using IP Address in Place of URL

- Use online services, such as **Host2ip**, that convert host/domain names to an IP address to find the IP address of the blocked website
- In order to bypass the firewall, type the **IP address directly into the browser's address bar** in place of typing the blocked website's domain name
 - For example, to access Facebook, type its **IP address (173.252.120.6)** instead of typing domain name

The diagram illustrates a bypass technique. On the left, a 'Tester' icon is shown with a browser window. A dashed line with a green checkmark connects the browser to a brick wall with flames representing a 'Firewall'. The IP address '173.252.120.6' is written above this line. Below the firewall, a trash can icon indicates that the original domain 'www.facebook.com' is blocked. A second dashed line connects the firewall to a 'Facebook Login' page icon on the right, showing the bypass is successful.

Note: This method fails if the blocking software tracks the IP address sent to the web server

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

Try to Bypass the Firewall Using IP Address in Place of URL

This method involves typing the IP address directly in browser's address bar in place of typing the blocked website's domain name in order to bypass the firewall. Since most of the organizations block certain websites, such as social networking, shopping, movies, music, and so on, even Internet Service Providers (ISPs) block some websites. They use firewalls to block websites using their DNS servers while some of the firewalls use IP-based blocking. An advanced firewall can identify the keywords in the URL and content to block them dynamically based on the rule.

To bypass the firewall and to access the blocked websites, there is a simple method to access them by using their IP address. To know the IP address of the blocked website, simply run CMD on your machine, type "**ping www.facebook.com,**" and press enter. This will return the IP address of the URL. Then simply, type the IP address directly into the browser's address bar in place of typing the blocked website's domain name.

For example, to access Facebook, type its IP address (173.252.120.6) instead of typing domain name.

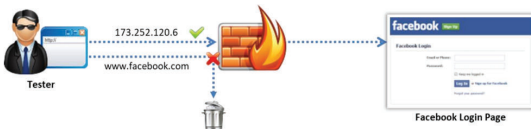
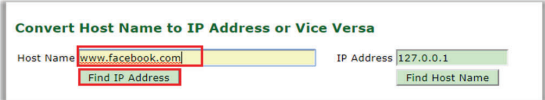


Figure 7.30: Schematic displaying bypassing firewall using IP address in place of URL

Use online services, such as Host2ip, that convert host/domain names to an IP address to find the IP address of the blocked website. It will get the IPv4 address associated to the given host name or website. Simply visit the URL <http://www.hcidata.info/host2ip.cgi> and, for example, type the **Host Name** as *www.facebook.com*, and click on **Find IP Address** as shown in the following screenshot.



The screenshot shows a web interface titled "Convert Host Name to IP Address or Vice Versa". It features two input fields: "Host Name" and "IP Address". The "Host Name" field contains "www.facebook.com" and has a red box around it with a "Find IP Address" button below it. The "IP Address" field contains "127.0.0.1" and has a "Find Host Name" button below it.

Convert Host Name to IP Address or Vice Versa	
Host Name	IP Address
www.facebook.com	127.0.0.1
Find IP Address	Find Host Name

Figure 7.31: Screenshot displaying convert host name to IP address using Host2ip

The result will be displayed as shown in the following screenshot and the IP address of Facebook is 157.240.1.35.



The screenshot shows a web interface titled "Details of www.facebook.com". It displays the results of the conversion: Host Name, IP Address, and Location.

Details of www.facebook.com	
Host Name	: www.facebook.com
IP Address	: 157.240.1.35
Location	: United States (accuracy)

Figure 7.32: Screenshot displaying the results in Host2ip

Note: This method fails if the blocking software tracks the IP address sent to the web server.

Try to Bypass the Firewall using Anonymous Website Surfing Sites



- Search over the Internet for anonymous website surfing sites that provide options to **encrypt the URLs** of the websites
- These websites **hide the actual IP address** and show another IP address, which could prevent the website from being blocked, thus allowing access to them

Anonymous Website Surfing Sites

1	http://anonymouse.org	5	http://proxify.com
2	http://www.anonymizer.com	6	http://www.spysurfing.com
3	http://www.webproxyserver.net	7	http://zendproxy.com
4	http://kproxy.com	8	http://anype.com

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

Try to Bypass the Firewall using Anonymous Website Surfing Sites

Search over the internet for anonymous website surfing sites that provide options to encrypt the URLs of the websites. These websites hide the actual IP address and show another IP address, which could prevent the website from being blocked, thus allowing access to them. Anonymous web-surfing sites and proxies help you to browse the internet anonymously without reflecting your IP address and unblock the blocked websites (i.e., evade Firewall restrictions). There are number of anonymous web-surfing sites available online, some of which provide options to encrypt website URLs.

Here is a list of proxy servers that can help you to access blocked websites:

- **Anonymouse**

Source: <http://anonymouse.org>

This service allows you to surf the web without revealing any personal information.

- **Anonymizer Universal**

Source: <https://www.anonymizer.com>

Anonymizer Universal keeps your online activities safe, private, and secure.

- **Webproxyserver**

Source: <http://www.webproxyserver.net>

Webproxyserver.net is an SSL (or secured proxy server) that helps a user to change their IP address online and protect their identity. It has the capability to bypass restrictions, thus giving more advantage to the user.

- **Proxify**

Source: <https://proxify.com>

Proxify is an anonymous proxy service that allows anyone to surf the web privately and securely. Through Proxify, you can use websites, but they cannot uniquely identify or track you. Proxify hides your IP address and its encrypted connection prevents monitoring of your network traffic.

- **Spysurfing**

Source: <http://www.spysurfing.com>

Spysurfing is a free anonymous web-based proxy service. It ensures your privacy by letting you browse the web as an anonymous user. Websites often track, log, and analyze your IP address, geographical information, web browser, and other personal information. Spysurfing helps you avoid giving out this information by hiding your personal information, such as your IP address, from websites.

- **Zendproxy**

Source: <https://zendproxy.com>

It is a web proxy for unblocking. An online web proxy is a computer system or website that acts as the middle person between the user and the internet. A web proxy allows users to surf the internet anonymously; the online anonymizer hides the IP address of the user. By hiding the IP address of the user, the online anonymous proxy strips the IP address from the user's request to view that content, and is then able to view it without restrictions.

- **kproxy**

Source: <https://kproxy.com>

It is an online anonymous website surfer which can bypass online blocks to access foreign content like a local. It also allows you to surf privately by hiding your IP address (your location and personal information) online and protect your data from snooping by your internet service provider.

Try to Bypass the Firewall using a Proxy Server



Find an appropriate proxy server	In the Port box, type the port number that is used by the proxy server for client connections (by default, 8080)
On the Tools menu of any Internet browser, go to LAN of Network Connections tab, and then click LAN/Network Settings	Click to select the bypass proxy server for local addresses check box, if you do not want the proxy server computer to be used when connected to a computer on the local network
Under the proxy server settings; select the use a proxy server for LAN	Click OK to close the LAN Settings dialog box
In the Address box, type the IP address of the proxy server	Click OK again to close the Internet Options dialog box

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

Try to Bypass the Firewall using a Proxy Server

Proxy server acts as an intermediate between two end-point devices such as a computer, and other server from which a client or user is requesting for a service. The proxy server may reside in the same server as a firewall or on a separate server that accepts the requests through the firewall. The proxy server can be used for both legal and illegal purposes. In an organization, it is used to simplify the security and administrative control. In a personal context it is used for anonymous and privacy purpose. Illegally you can use a proxy server to monitor the traffic and weaken the user privacy.

Following are the steps used to bypass a firewall using a proxy server:

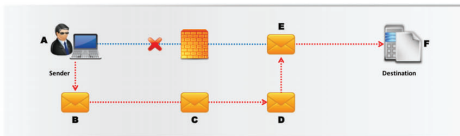
- Find an **appropriate** proxy server
- On the **Tools** menu of any internet browser, go to **LAN of Network Connections** tab, and then click **LAN/Network Settings**
- Under Proxy server settings, select the option: **Use a proxy server for LAN**
- In the **Address** box, type the IP address of the proxy server
- In the **Port** box, type the port number that is used by the proxy server for client connections (by default, 8080)
- Click to select the bypass proxy server for local addresses; check the box if you do not want the proxy-server computer to be used when connected to a computer on the local network
- Click **OK** to close the **LAN Settings** dialog box
- Click **OK** again to close the **Internet Options** dialog box

Try to Bypass the Firewall using Source Routing



Source routing is a technique in which the sender of a packet partially or completely **specifies the route** the packet takes through the network

Modify the **addressing information** in the IP packet header and the source address bits field in order to bypass the firewall



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

Try to Bypass the Firewall using Source Routing

Using this technique, you can partially or completely designate the route that a packet should take through the network, in such a way that the designated route should bypass the firewall node. Using this technique, you can evade firewall restrictions by modifying the addressing information in the IP packet header and the source address bits field. When these packets travel through the network nodes, each router will check the IP address of the destination and choose the next node to which to forward them. In source routing, you can make some or all of these decisions on the router. Source routing can be performed by two approaches: loose source routing and strict source routing. In a loose source routing, you have to specify one or more stages from which the packet must go through, whereas in strict source routing, you have to specify the exact route through which the packet must go through.

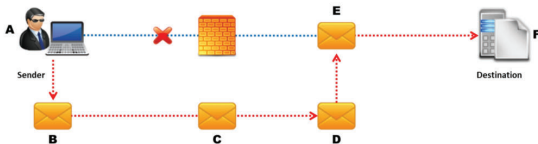


Figure 7.33: Schematic diagram displaying the example of source routing

Try to Bypass the Firewall using HTTP Tunneling Method

This method can be implemented if the target company has a **public web server with port 80** used for HTTP traffic, that is unfiltered on its firewall

Many firewalls **do not examine the payload of an HTTP packet** to confirm that it is legitimate HTTP traffic; thus, it is possible to tunnel traffic inside TCP port 80 because it is already allowed

Tools, such as **Super Network Tunnel**, **HTTPTunnel**, **HTTPORT**, and **HTTHOST** use this technique of tunneling traffic across **TCP port 80**

Upload the **server onto the target system** and tell it which port is to be redirected through **TCP port 80**

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

Try to Bypass the Firewall using HTTP Tunneling Method

This method can be implemented if the target company has a public web server with port 80 used for HTTP traffic, that is unfiltered on its firewall. Many firewalls do not examine the payload of an HTTP packet to confirm that it is legitimate HTTP traffic, thus it is possible to tunnel traffic inside TCP port 80, because it is already allowed.



Figure 7.34: Schematic diagram displaying firewall bypass using HTTP tunneling method

HTTP tunneling grants access to the network users to achieve restricted connectivity through a firewall or proxy. In such conditions, some applications may lack native communications support. These restrictions include the following:

- Blocking of TCP/IP ports, traffic initiated from outside the network, and, network protocols except a few commonly used protocols, and so on
- Access to surf on denied websites
- Post in forums anonymously by hiding the IP address
- To use application such as chatting through ICQ or IRC, instant messengers, games, browsers, and so on
- Sharing of confidential resource over HTTP securely
- Downloading files with filtered extensions and/or with malicious code

For instance, consider that organization firewalls restrict users to access all ports except 80 and 443, and a user may want to use FTP. HTTP tunneling enables FTP use via HTTP protocol. HTTP tunnel creates a bidirectional virtual data connection tunneled in HTTP traffic. It works with the help of FTP client software to perform protocol encapsulation by enclosing data packets of one protocol such as SOAP or JRMP within HTTP packets on, for example, local port 80. Then these packets are sent through the firewall or proxy server as normal internet traffic, which is then directed to HTTP tunneling server software located outside the network. Upon receiving the packets, this server then unwraps FTP data and redirects the packet to the remote FTP server.

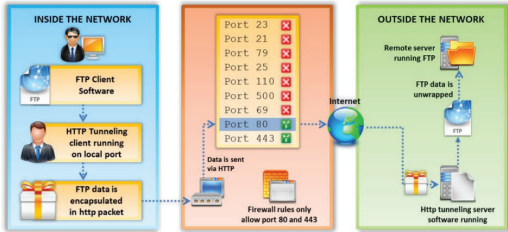


Figure 7.35: Schematic diagram displaying the working of HTTP tunneling method

You can use tools such as Super Network Tunnel, HTTPtunnel, HTTPORT, and HTTPHOST to tunnel traffic across TCP port 80. Upload the server onto the target system and tell it which port is to be redirected through TCP port 80.

- **HTTPort and HTTPHost**

Source: <https://www.targeted.org>

HTTPort allows you to bypass your HTTP proxy that is blocking you from the internet. With HTTPort you may use various internet software from behind the proxy, for example, email, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, and so on. The basic idea is that you set up your internet software in such a manner that it connects to your own local PC as if it was the remote server it needed. HTTPort then intercepts that connection and runs it through a tunnel through the proxy.

HTTPort doesn't really care for the proxy as such; it works perfectly with firewalls, transparent accelerators, NATs, and basically anything that lets HTTP protocol through. HTTP is the basis for web surfing, so if you can freely surf the web from where you are, HTTPort will bring you the rest of the internet applications. The remote host method is capable of tunneling through any proxy.

HTTPort uses a special server software called HTTPHost, which is installed outside the proxy-blocked network. It is a web server, and thus when HTTPort is tunneling, it sends

a series of HTTP requests to the HTTPHost. The proxy responds as if the user is surfing a website, and thus allows the user to do so. HTTPHost in turn performs half of the tunneling and communicates with the target servers. This mode is much slower, but works in most cases and features strong data encryption that makes proxy logging useless.

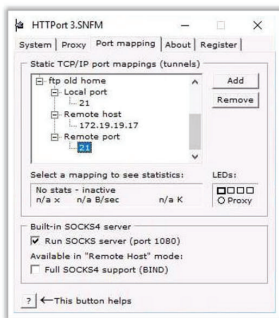


Figure 7.36: Screenshot displaying the HTTPort port mapping



Figure 7.37: Screenshot displaying the HTTPHost application log

■ Super Network Tunnel

Source: <http://www.networktunnel.net>

Super Network Tunnel is professional http tunneling software that includes http tunnel client and server software.

It like a secure VPN software that allows you to access your internet programs without being monitored at work, school, or the government, and gives you an extra layer of protection against hackers, spyware, or ID theft. It can easy bypass any firewall, for example, - Surf, IM, P2P, Game, and so on. It is easier than VPN, no complicated settings, portable, plug and play. It can also support http proxy, and normally completes a tunnel config within 10 minutes.

Despite other http tunnel software, Super Network Tunnel is really a two-way http tunnel software. The client computer can access the server computer's network and control the server's computer. It can run any game/IM/P2P/Stock software via server's network. Also, the server computer can access the client computer's network and remotely control the client's computer visa.

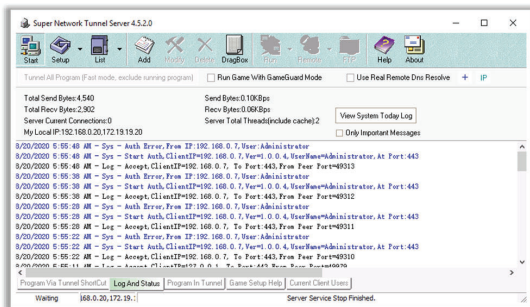


Figure 7.38: Screenshot displaying the results in Super Network Tunnel

■ HTTP-Tunnel

Source: <http://www.httptunnelclient.com>

HTTP-Tunnel technology allows users to perform various internet tasks despite the restrictions imposed by firewalls. This is made possible by sending data through HTTP (port 80). Additionally, HTTP-Tunnel technology is very secure, making it indispensable for both average and business communications.

Following are various applications utilized to implement HTTP-Tunnel technology:

HTTP-Tunnel Client: An application that runs in the system tray acting as a SOCKS server, managing all data transmissions between the computer and the network.

HTTP-Tunnel Server: A customizable server software solution for both personal and corporate networks.

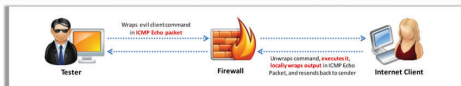
HTTP-Tunnel ActiveX Control: It allows developers to incorporate HTTP-Tunnel technology into their software applications.

Corporate Messenger (VCM): It is a secure and serverless instant messaging application, suitable for corporate intranets.

Try to Bypass the Firewall using ICMP Tunneling Method



- It allows tunneling a backdoor shell in the **data portion of ICMP Echo packets**
- RFC 792**, which **delineates ICMP operation**, does not define what should go in the data portion
- The **payload portion is arbitrary** and not examined by most firewalls; thus, any data can be inserted into the payload portion of the ICMP packet, including a backdoor application
- Some administrators keep **ICMP open on their firewall** because it is useful for tools like **ping** and **traceroute**
- Assuming that ICMP is allowed through a firewall, use **Loki ICMP tunneling** to execute commands of choice by **tunneling them inside the payload of ICMP echo packets**



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

Try to Bypass the Firewall using ICMP Tunneling Method

ICMP protocol is used to send error message to the client. As this service is required for network communication, users often enable this service on their networks. This provides you with full access to target networks, allowing them to tunnel a backdoor shell in the data portion of ICMP Echo packets.

RFC 792 outlines the ICMP operation; it does not define what should go in the data portion. The payload portion is arbitrary and not examined by most firewalls, thus any data can be inserted into the payload portion of the ICMP packet, including a backdoor application. Some administrators keep ICMP open on their firewall, because it is useful for tools, such as ping and traceroute.

Moreover, it does not cause a big threat from the security perspective. You can take advantage of enabled ICMP protocol on the network and perform ICMP tunneling to send your malicious data into the target network. Therefore, you can try to bypass firewall by inserting any data, including a backdoor application, into the payload portion of the ICMP packet. Assuming that ICMP is allowed through a firewall, use Loki ICMP tunneling to execute commands of choice by tunneling them inside the payload of ICMP echo packets.

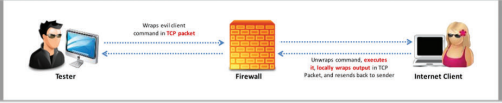


Figure 7.39: Schematic diagram displaying working of ICMP tunneling method

Try to Bypass the Firewall using ACK Tunneling Method



- 1 It allows tunneling a backdoor application with **TCP packets with the ACK bit set**
- 2 The ACK bit is used to **acknowledge receipt of a packet**
- 3 Some firewalls **do not check packets with the ACK bit set** because ACK bits are supposed to be used in response to legitimate traffic that is already being allowed through
- 4 Perform ACK tunneling using tools, such as **AckCmd** (<http://ntsecurity.nu>)



The diagram illustrates the ACK tunneling process. On the left, a **Tester** (represented by a person at a computer) sends a packet to a **Firewall** (represented by a brick wall). The packet is labeled "Wraps evil client command in TCP packet". The packet passes through the firewall. On the right, an **Internet Client** (represented by a person at a laptop) receives the packet. The client's actions are described as "Unwraps command, executes it, locally wraps output in TCP Packet, and resends back to sender".

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

Try to Bypass the Firewall using ACK Tunneling Method

In TCP protocol communication, a session is established when the client sends a SYN segment, the server responds with SYN/ACK segment, and the client confirms with a ACK segment. The traffic in the current session consists of ACK segments.

Ordinary packet-filtering firewalls define their rule sets based on when the SYN packet in TCP-level communication is established. This is because such a firewall assumes that only the SYN packet is coming from the client and thus possibly contains malicious code. This kind of firewall ignores the possibility that an attacker can also inject malicious code in the ACK packet. Since ACK packets are sent after establishing a session, ACK traffic is considered legitimate. Another reason why filtering of ACK packets is ignored is to lessen the workload of firewalls, as there can be many ACK packets for one SYN packet.

ACK tunneling allows tunneling a backdoor application with TCP packets with the ACK bit set. The ACK bit acknowledges the receipt of a packet. As stated earlier, some firewalls do not check packets with the ACK bit set because ACK bits are supposed to be used in response to legitimate traffic that has already been allowed to pass through. You can use tools such as AckCmd (<http://ntsecurity.nu>) to perform ACK tunneling.

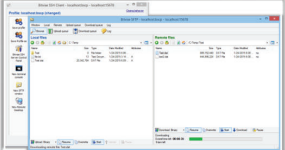


Figure 7.40: Schematic diagram displaying working of ACK tunneling method

Try to Bypass the Firewall using SSH Tunneling Method



- SSH tunneling method allows you to connect to **external SSH server**
- It creates an SSH tunnel and connects **local machine port** to the remote port 80 on remote web server in order to **bypass firewall restrictions**
- Use an SSH connectivity tool, such as **OpenSSH, Bitvise**, etc., to encrypt and tunnel all the traffic (including passwords) from a local machine to a remote machine to avoid detection by the target organization's perimeter security controls



Source: <http://www.bitvise.com>

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

Try to Bypass the Firewall using SSH Tunneling Method

SSH tunneling method allows you to connect to external SSH server. It creates an SSH tunnel and connects a local machine port to the remote port 80 on remote web server in order to bypass firewall restrictions. It involves sending unencrypted network traffic through an SSH tunnel. For example, suppose you want to transfer files on an unencrypted FTP protocol, but the FTP protocol is blocked on the target firewall, the unencrypted data can be sent over an encrypted SSH protocol using SSH tunneling. You can make use of this technique to bypass firewall restrictions. They connect to external SSH servers and create SSH tunnels to port 80 on the remote server, thereby bypassing firewall restrictions. You can use SSH connectivity tools such as OpenSSH, Bitvise, and so on to encrypt and tunnel all the traffic (including passwords) from a local machine to a remote machine to avoid detection by the target organization's perimeter security controls.

- **Bitvise**

Source: <https://www.bitvise.com>

Bitvise SSH Server provides secure remote login capabilities to Windows workstations and servers by encrypting data during transmission. It is ideal for remote administration of Windows servers; for advanced users who wish to access their home machine from work, or their work machine from home; and for a wide spectrum of advanced tasks, such as establishing a VPN using the SSH TCP/IP tunneling feature or providing a secure file depository using SFTP.

Bitvise SSH Client for Windows includes terminal emulation, graphical as well as command-line SFTP support, an FTP-to-SFTP bridge, tunneling features—including dynamic port forwarding through integrated proxy—and remote administration for SSH server.

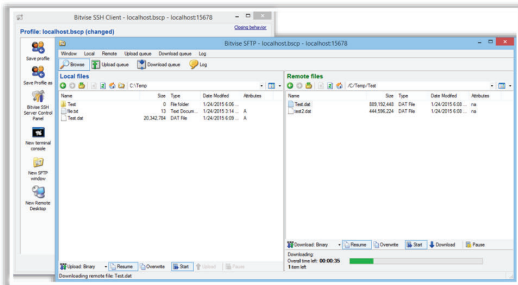


Figure 7.41: Screenshot displaying Bitvise SSH server Control Panel

- **OpenSSH**

Source: <https://www.openssh.com>

OpenSSH is the premier connectivity tool for remote login with the SSH protocol. Make use of OpenSSH (OpenBSD Secure Shell) to encrypt and tunnel all traffic from a local machine to a remote machine to avoid detection by perimeter security controls. OpenSSH is a set of computer programs that provides encrypted communication sessions over a computer network using the SSH protocol.

Example:

```
Syntax: ssh -f user@certifiedhacker.com -L  
5000:certifiedhacker.com:25 -N
```

In the preceding syntax, **-f** refers to background mode, **user** is the username, and **@certifiedhacker.com** is the server that you are logging into.

-L 5000:certifiedhacker.com:25 refers to **local-port:host:remote-port** respectively.

-N option in the syntax refers to “Do not execute the command on the remote system.”

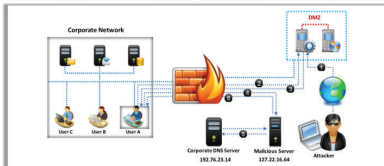
The preceding syntax forwards the local port 5000 to port 25 on certifiedhacker.com encrypted. Simply point your email client to use **localhost:5000** as the SMTP server.

Try to Bypass the Firewall through MITM Attack



DNS Server Poisoning Process

1. User A requests **WWW.example.com** from the **corporate DNS server**
2. Corporate DNS server sends the **IP address (127.22.16.64) of the attacker**
3. User A accesses the **attacker's malicious server**
4. Attacker connects with the real host and **tunnels the user's HTTP traffic**
5. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine



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

Try to Bypass the Firewall through MITM Attack

Most security administrators concentrate on the possibility of an external or internal network bypassing their firewall while ignoring the fact that firewalls can be bypassed using MITM attacks on DNS servers. In MITM attacks, you need to make use of DNS servers and routing techniques to bypass firewall restrictions. They may either take over the corporate DNS server or spoof DNS responses to perform the MITM firewall attack.

Following are the steps for DNS server poisoning process:

1. User A requests **www.example.com** from the corporate DNS server.
2. Corporate DNS server sends the IP address (127.22.16.64) of the attacker.
3. User A accesses the attacker's malicious server.
4. Attacker connects with the real host and tunnels the user's HTTP traffic.
5. The malicious codes embedded in the attacker's web page are downloaded and executed on the user's machine.

The schematic diagram for this process is displayed in the following figure:

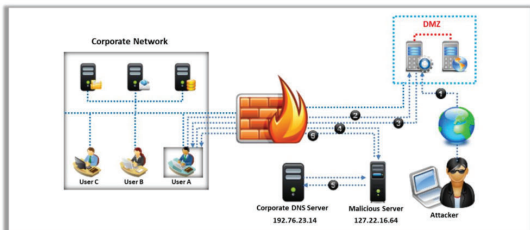


Figure 7.42: Diagram demonstrating steps for DNS server poisoning process

Try to Bypass the Firewall using Malicious Contents



- Create a malicious file using **Trojan Horse** Construction Kit
- Embed a **malicious file** in software installation files, mobile phone software, or text, multimedia, and graphics files to carry malicious content
- Send the contents containing **malicious code** to the user and trick him/her to open it so that the malicious code can be executed

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

Try to Bypass the Firewall using Malicious Contents

Create a malicious file using Trojan Horse Construction Kit. Embed a malicious file in software installation files, mobile phone software, or text, multimedia, and graphics files, to carry malicious content. Send the content containing malicious code to the user and trick him/her to open it, so that the malicious code can be executed. Apart from Trojan Horse Construction Kit, you can use tools such as Batch Trojan Generator, Umbra Loader—Botnet Trojan Maker, DELmE's Batch Virus Maker, JPS Virus Maker, and so on to create a malicious payload to compromise the target firewall.



Assessing IDS Security Implementation

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

Assessing IDS Security Implementation

To protect the networks, organizations use various network security measures such as firewalls, intrusion detection system (IDS), intrusion prevention system (IPS), and so on.

In this section, we will discuss the penetration steps for assessing the IDS. This section also discusses about the threats associated with IDS and how a well-planned penetration testing helps in protecting it.

Why IDS Penetration Testing?



- ➡ To check if IDS properly enforces an **organization's IDS policy**
- ➡ To check if the **IDS** enforces organization's network security policies
- ➡ To check if the IDS is good enough to **prevent external attacks**
- ➡ To check the effectiveness of the **network's security perimeter**
- ➡ To check the **amount of network information accessible** to an intruder
- ➡ To check the IDS for **potential breaches of security** that can be exploited
- ➡ To verify whether the **security policy is correctly enforced** by a sequence of IDS rules

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

Why IDS Penetration Testing?

The most preferred and trustful defensive measure that most of the organizations follow these days is implementing an IDS that can help to detect and alert their security team on finding any trace of vulnerability. IDS can either be a software or hardware that identifies, logs, and reports any suspected, malicious, and inappropriate activity. The main feature of the IDS is to monitor network activity on the host network/workstation and generate alerts when there is an intrusion. However, these so-called "protective walls of the system" may also give rise to various vulnerabilities and attacks that can lead to unrecoverable damage to the organization. It is the responsibility of the firm to appoint a penetration tester in order to find and fix the loopholes in their intrusion system. Because, their inattention may allow the attackers to change the settings of alarm in IDS system and thereby access the organization's resources. IDS penetration testing also helps in evaluating the IDS for ingress and egress traffic-filtering capabilities.

IDS penetration testing can be performed to do the following:

- Check if IDS properly enforces an organization's IDS policy
- Check if the IDS enforces organization's network security policies
- Check if the IDS is good enough to prevent external attacks
- Check the effectiveness of the network's security perimeter
- Check the amount of network information accessible to an intruder
- Check the IDS for potential breaches of security that can be exploited
- Verify whether the security policy is correctly enforced by a sequence of IDS rules

Common Techniques Used to Evade IDS Systems



- I** Try the **pattern matching approach** to identify potential attacks within the exploit code
- II** Use the **Unicode Evasion method**, which allows for viewing files on the IIS server
- III** Search for the **central log server's IP address** and crash the system using a DoS attack
- IV** Send specially **crafted packets** in order to trigger alerts and breed a large number of false reports
- V** **Flood the network** with noise traffic to exhaust its resources examining risk-free traffic

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

Common Techniques Used to Evade IDS Systems

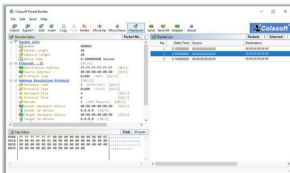
Following are some of the common techniques that are used to evade IDS system:

- Try the pattern matching approach to identify potential attacks within the exploit code. In this approach, depending on the regularly used strings in the code and the known attacks, potential attacks are recognized. However, this approach has some problems as the input, which is going to trigger a certain vulnerability must be the same, thus small change in input makes it difficult to create a useful pattern.
- Use the Unicode Evasion method that allows viewing files on the IIS server. Every character has a unique identifier for all written languages that provides uniform presentation of every character of each language. Single character may have too many representations, which is difficult to match with the signature.
- Many organizations use central logging server to accumulate IDS alert logs. The main function of the central logging server is to alert the log data. Search for central log server's IP address and crash the system using a DoS attack.
- Another similar technique to the DoS method is to send specially crafted packets in order to trigger alerts and breed a large number of false reports.
- An IDS primarily depends on its resources, such as memory, power of the processor, and analyzing traffic. Flood the network with noise traffic to exhaust its resources examining risk-free traffic.

Test for Resource Exhaustion



- Every IDS system has **memory**, **CPU**, and **bandwidth** limitations and is prone to resource exhaustion attacks
- IDS performance might degrade or fail if these resources are **exhausted**
- Use tools, such as **Colasoft Packet Builder**, **Network Traffic Generator and Monitor**, etc. to generate the traffic
- Test by sending large amounts of **traffic** to the IDS



Source: <https://www.colasoft.com>

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

Test for Resource Exhaustion

Every IDS system has memory, CPU, and bandwidth limitations, and is prone to resource exhaustion attacks. A resource exhaustion attack is a form of DoS attack where the available network resources are exhausted, resulting in a system failure or crash. IDS performance might degrade or fail if these resources are exhausted. Test by sending large amounts of traffic to the IDS. You can use tools such as Colasoft Packet Builder, Network Traffic Generator and Monitor, and so on to generate the traffic.

- **Colasoft Packet Builder**

Source: <https://www.colasoft.com>

Colasoft Packet Builder enables creating custom network packets. As a pen tester, you can use this tool to check the network protection against intrusion detection attacks and intruders.

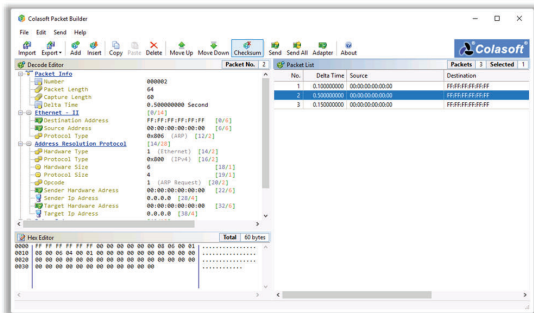


Figure 7.43: Screenshot of Colasoft Packet Builder tool

Test the IDS by Sending an ARP Flood



- Flood the network by sending **ARP packets**
- Use tools such as **NetScanTools Pro** to generate the ARP packets
- See the **IDS response** and how it **reacts** to this attack



Other ARP Packet Generator Tools

- Arping (<http://www.habets.pp.se>)
- arp-scan (<https://github.com>)



Source: <https://www.netscantools.com>

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

Test the IDS by Sending an ARP Flood

The ARP is part of the internet protocol (IP) that maps a computer's IP address with its MAC address. Create a large number of random ARP request packets, attempt to flood the network, and analyze the IDS response and see how it reacts to this ARP flood attack. You can use tools such as NetScanTools Pro to generate the ARP packets and perform ARP flood attack on IDS.

NetScanTools Pro

Source: <https://www.netscantools.com>

NetScanTools Pro packet generator allows you to construct or build a TCP, UDP, ICMP, CDP (Cisco Discovery Protocol), ARP/RARP, or RAW packet, and sends one or more packets to a target IPv4 address. This tool helps you to bypass IDS rules and generates ARP packets.

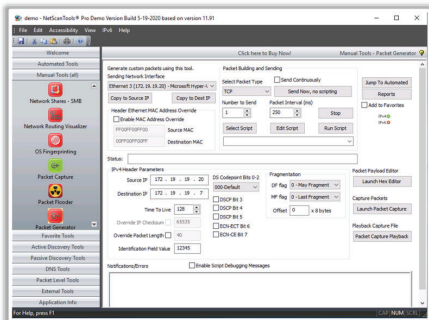


Figure 7.44: Screenshot displaying ARP Packet generation in NetScanTools Pro tool

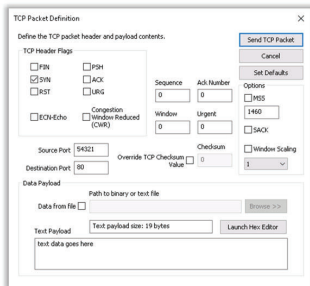


Figure 7.45: Screenshot showing TCP Packet Definition in NetScanTools Pro

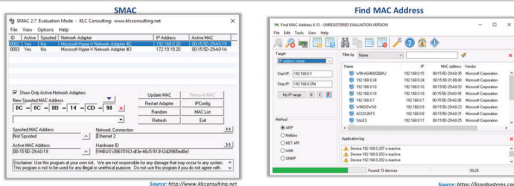
Some of the additional ARP packet generator tools are as follow:

- Arping (<http://www.habets.pp.se>)
- arp-scan (<https://github.com>)

Test the IDS by MAC Spoofing



- Traffic can be disrupted on a network if two **Ethernet adapters** have exactly the same hardware (or **MAC** – Media Access Control) addresses
- Use tools such as **SMAC**, **macof**, etc. to generate spoofed MAC addresses
- Test the IDS by sending spoofed **MAC addresses**



Test the IDS by MAC Spoofing

Every hardware device is assigned with a unique MAC address, and in a network, MAC address of various devices is hard-coded in a network interface controller (NIC). It is very important to note that two network devices should not have same MAC Address. If it happens then the traffic in the network may be disrupted. MAC spoofing is a technique of manipulating or masking the MAC address of any network device with a random MAC address. By performing MAC spoofing attack, you can bypass the Access Control Lists on servers, firewalls, IDS, routers, and so on. Test the IDS by sending spoofed MAC addresses and analyze the results. You can use tools such as SMAC, Find MAC Address, macof and so on to generate spoofed MAC addresses.

■ SMAC

Source: <http://www.klconsulting.net>

SMAC is a Windows MAC Address Modifying Utility (MAC Address spoofing) that allows users to change MAC address for almost any Network Interface Cards (NIC) on the Windows systems, regardless of whether the manufacturers allow this option or not.

Following are some of the features of SMAC tool:

- MAC Address Lookup
- Automatically activate new MAC address right after changing it (you have the option to manually activate)
- Show the manufacturer of the MAC Address
- Show All or "Only Active" network adapters
- Randomly Generate any new MAC Address or based on a selected manufacturer

- Pre-load MAC addresses list and choose the new MAC address from the list (Professional Edition)
- "IPConfig" button—click to show network adapter (IPConfig) info to confirm changes

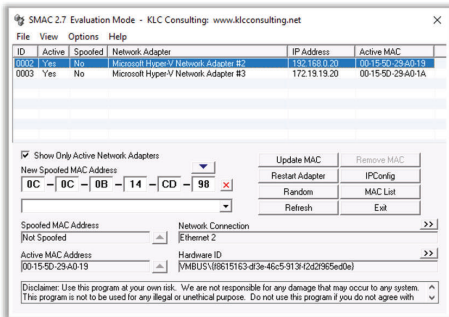


Figure 7.46: Screenshot of SMAC tool

■ Find MAC Address

Source: <https://lizardsystems.com>

This tool finds the MAC address of a local or remote computer on the network. You can select the target and method to find the MAC address of a local or remote computer on the network.

Following are the key features of Find the MAC Address tool:

- Getting the MAC address of the local computer
- Getting the MAC address of one remote computer either by its name or by IP address
- Getting the MAC addresses of all computers on the network
- Getting the MAC addresses of all computers within the specified range of IP addresses
- Getting the MAC addresses of all computers from the specified computer list
- Five methods for finding MAC addresses (ARP, NetBIOS, NetAPI, WMI, SNMP)
- Detecting the manufacturer of the network card by its MAC address
- Detecting the names of network cards (only if WMI or SNMP is used)

- Searching the scan results
- Saving the scan results to and loading them from a file in the XML format
- Exporting the network scan results to text files or a web document

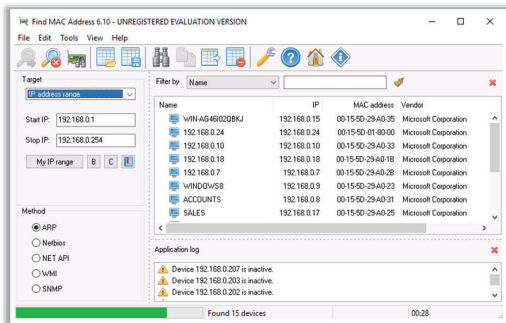


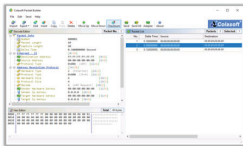
Figure 7.47: Screenshot of Find MAC Address tool

Test the IDS by IP Spoofing



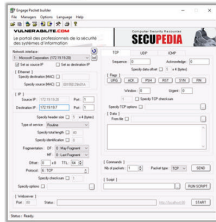
- Spooft the **IP address** to flood the IDS
- Check for the responses received by the **target system**

Colasoft Packet Builder



Source: <https://www.colasoft.com>

Engage Packet Builder



Source: <http://www.engagesecurity.com>

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

Test the IDS by IP Spoofing

IP spoofing in a network means generating the IP packets with a fake source IP address for performing various attacks. You can flood the IDS with IP address spoofed packets and analyze the responses received by the target system.

Colasoft Packet Builder

Source: <https://www.colasoft.com>

Colasoft Packet Builder enables creating custom network packets. You can use Colasoft Packet Builder to generate IP packets and perform flood attack on the IDS.

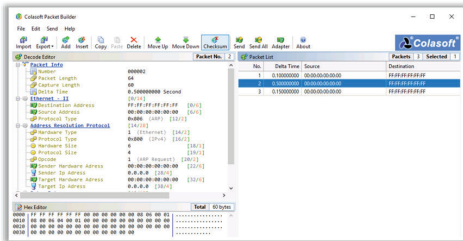


Figure 7.48: Screenshot of Colasoft Packet Builder tool

Engage Packet Builder

Source: <http://www.engagesecurity.com>

Engage Packet Builder is a scriptable packet builder for Windows platform. It is completely based on a modified libnet library. It is also used for Firewall and IDS testing. Flood attacks can also be performed with this tool. This tool is also used for spoofing the IP address and performing flood attacks on the target device.

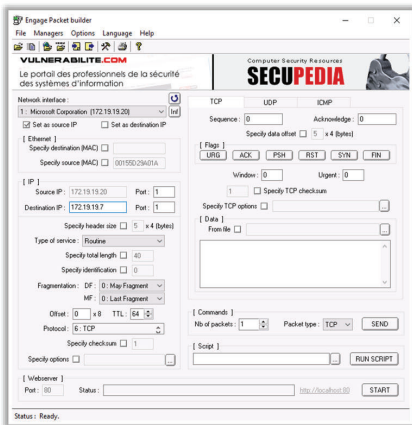


Figure 7.49: Screenshot of Engage Packet Builder tool

Test the IDS by Sending SYN Floods



1

Many TCP implementations are vulnerable to a **resource-exhaustion attack** known as SYN flooding, in which excessive requests are made to create sessions, causing memory utilization to occur

2

If these SYN packets are spoofed from addresses that do not exist, **no response packet containing SYN/ACK** will be received, and the pending connection queue will expand

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

Test the IDS by Sending SYN Floods

Many TCP implementations are vulnerable to a resource-exhaustion attack known as SYN flooding, in which excessive requests are made to create sessions, thus causing excessive memory utilization to occur. In this type of attacks, numerous SYN packets are sent to an open TCP port. The victim host responds with a SYN/ACK packet, creates an entry in its pending-connection queue, and waits to complete the TCP three-way handshake. When the target host waits for the response, it takes a finite amount of memory. Hence, if the target host gets numerous SYN packets and fails to complete the three-way handshake, it can lead to an increase in the memory and make it unable to process the legitimate connection requests.

Spoof the IP address and send the SYN packets to the IDS to test the IDS. If these SYN packets are spoofed from addresses that do not exist, no response packet containing SYN/ACK will be received, and the pending connection queue will expand.

Test the IDS by Editing and Replaying Captured Network Traffic





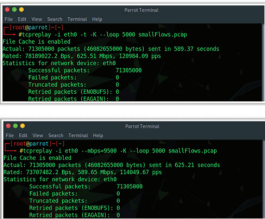
Capture the traffic running on a target computer network by using **packet sniffing tools**, such as Wireshark



Use **Tcpreplay** tool for editing and replaying captured network traffic



Replay the traffic back onto the network



Source: <http://tcpreplay.apnic.cc.com>

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

Test the IDS by Editing and Replaying Captured Network Traffic

Capture the traffic running on a target computer network by using packet-sniffing tools such as Wireshark. Use tools such as Tcpreplay for editing and replaying captured network traffic, and attempt to Replay the traffic back onto the network.

■ Wireshark

Source: <https://www.wireshark.org>

Wireshark is a packet-sniffing tool with live capture and offline analysis feature. It runs on multiple platforms such as Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others. Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility. You can use this tool to capture the network traffic where IDS is installed.

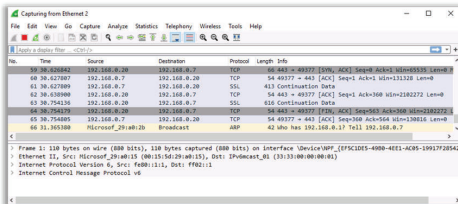
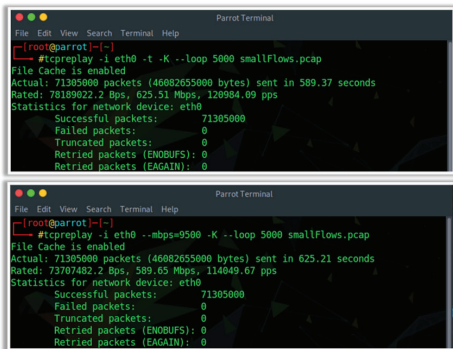


Figure 7.50: Screenshot of Wireshark tool

- **Tcpreplay**

Source: <http://tcpreplay.appneta.com>

Tcpreplay tool is used to replay & edit pcap files at arbitrary speeds onto the network. It is an open-source utility used for editing and replaying previously captured network traffic. Firstly, it was designed to replay malicious traffic patterns to IDS/IPS. The latest version 4.0.0 supports switches, IP flow/NetFlow appliances, and routers. It supports both single and dual NIC modes for testing both sniffing and in-line devices.



```
Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot[~]
#tcpreplay -i eth0 -t -K --loop 5000 smallFlows.pcap
File Cache is enabled
Actual: 71305000 packets (46082655000 bytes) sent in 589.37 seconds
Rated: 78189022.2 Bps, 625.51 Mbps, 120984.09 pps
Statistics for network device: eth0
    Successful packets: 71305000
    Failed packets: 0
    Truncated packets: 0
    Retried packets (ENOBUFS): 0
    Retried packets (EAGAIN): 0

Parrot Terminal
File Edit View Search Terminal Help
[~]root@parrot[~]
#tcpreplay -i eth0 --mbps=9500 -K --loop 5000 smallFlows.pcap
File Cache is enabled
Actual: 71305000 packets (46082655000 bytes) sent in 625.21 seconds
Rated: 73707482.2 Bps, 589.65 Mbps, 114049.67 pps
Statistics for network device: eth0
    Successful packets: 71305000
    Failed packets: 0
    Truncated packets: 0
    Retried packets (ENOBUFS): 0
    Retried packets (EAGAIN): 0
```

Figure 7.51: Screenshots of Tcpreplay tool

Test the IDS for a Denial-of-Service (DoS) Attack

Many IDSes today employ **central logging servers** that are used exclusively to store IDS alert logs

Perform a Denial-of-Service (DoS) attack on the central log server to slow down or crash the **central logging system**

Search for the central log server's **IP address**

Use DoS Attack tools such as HOIC, DDOSIM, DoS HTTP, Tor's Hammer, etc. to perform the DoS attack

HOIC

Source: <https://sourceforge.net>

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

Test the IDS for a Denial-of-Service (DoS) Attack

Many IDSes today employ central logging servers that are used exclusively to store IDS alert logs. Hence, you can search for central log server's IP address and perform a DoS attack to prevent legitimate users in accessing central logging servers, and this may lead to slow down or crash the central logging system. You can use DoS attack tools such as HOIC, DDOSIM, DoS HTTP, Tor's Hammer, and so on to perform the DoS attack on the target IDS.

- **HOIC**

Source: <https://sourceforge.net>

The HOIC is a popular DDoS attack tool that is free to download and available for Windows, Mac, and Linux platforms. The HOIC is an upgrade to an older program, the Low Orbit Ion Cannon (LOIC), documented in IntelliShield Alert 22057, which is an open-source tool written in the C# programming language. You can use this tool to perform DoS attack on the target IDS and analyze the response.

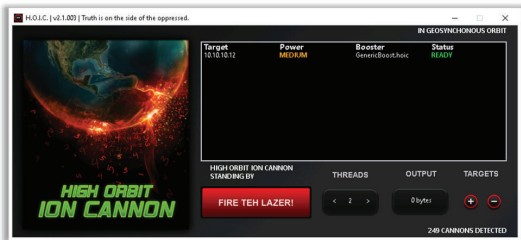


Figure 7.52: Screenshot of HOIC DoS attack tool

Try to Bypass IDS using Anonymous Website Surfing Sites and a Proxy Server



- Search over the Internet for anonymous website surfing sites which provide options to **encrypt the URLs** of the websites
- These websites will **hide the actual IP address** and will show another IP address, which could prevent the website from being blocked by the IDS, thus allowing access to the target system
- Search the Internet for a proxy server, configure it in your system, and try to bypass the IDS

Anonymous Web-Surfing Sites

1	http://anonymouse.org	6	http://proxify.com
2	http://www.anonymizer.com	7	http://www.spysurfing.com
3	http://www.webproxyserver.net	8	http://zendproxy.com
4	http://kproxy.com	9	http://anype.com

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

Try to Bypass IDS using Anonymous Website Surfing Sites and a Proxy Server

Search over the Internet for anonymous website surfing sites which provide options to encrypt the URLs of the websites. Anonymous web-surfing sites and proxies help you to browse the internet anonymously without reflecting your IP address and unblock the blocked websites (i.e., evade IDS restrictions). There are number of anonymous web-surfing sites available online, some of which provide options to encrypt website URLs. You can search the Internet for a proxy server, configure it in your system, and try to bypass the IDS.

Here is a list of proxy servers that can help you to access blocked websites:

- Anonymouse**
Source: <http://anonymouse.org>
- Anonymizer**
Source: <https://www.anonymizer.com>
- Webproxyserver**
Source: <http://www.webproxyserver.net>
- Proxify**
Source: <https://proxify.com>
- Spysurfing**
Source: <http://www.spysurfing.com>
- Zendproxy**
Source: <https://zendproxy.com>

- **kproxy**

Source: <https://kproxy.com>

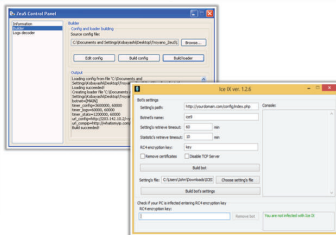
In order to bypass an IDS using Proxy Server, you should follow the following steps:

- Find an appropriate proxy server.
- On the Tools menu of any Internet browser, go to LAN of Network Connections tab, and then click LAN/Network Settings.
- Under Proxy server settings, select the use a proxy server for LAN.
- In the Address box, type the IP address of the proxy server.
- In the Port box, type the port number that is used by the proxy server for client connections (by default, 8080).
- Click to select the bypass proxy server for local addresses; check the box if you do not want the proxy-server computer to be used when connected to a computer on the local network.
- Click OK to close the LAN Settings dialog box.
- Click OK again to close the Internet Options dialog box.

Try to Bypass the IDS using Botnet



- Bots are software applications that **run automated tasks over the Internet** and perform simple repetitive tasks, such as web spidering and search engine indexing
- A botnet is a huge network of the compromised systems and can be used to **launch DOS attacks**
- Search the Internet for a **botnet builder tool** or botnet kit, such as Zeus (ZBot) or Ice IX
- **Configure** and build a botnet, and try to bypass the IDS



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

Try to Bypass the IDS using Botnet

Bots are software applications that run automated tasks over the internet and perform simple repetitive tasks, such as web spidering and search engine indexing. A botnet is a huge network of the compromised systems that perform various illegal activities by communicating through command and control. Botnets can be used to launch DOS attacks.

Generally, IDS have two different types of approaches for Botnet detection:

- Signature-based
- Anomaly-based

Signature-based: In this technique, IDS uses signatures of the botnets and attempts to match with the standard botnet signatures for detection.

Anomaly-based: In this technique, IDS considers several different network traffic anomalies to detect the presence of the bots in the network. These network anomalies can include the following:

- High network latency
- High traffic volume
- Traffic on unusual ports
- Unusual system behavior

Search the internet for a botnet builder tool or botnet kit, such as Zeus (ZBot) or Ice IX. Configure and build a botnet, and try to bypass the IDS.

▪ Zeus (ZBot)

Zeus, also known as Zbot, is a banking Trojan that specifically attempts to steal confidential information such as system information, online credentials, and banking details, and so on. Zeus is spread mainly through drive-by downloads and phishing schemes. Detecting Zeus malware is difficult due to its stealth techniques. This malware creates a network of infected machines and all the machines are connected to a C&C Server. The owners of the infected machines are not aware of the critical situation, since the Trojan runs silently in the background of the infected computer. You can create Zeus bot using Zeus Botnet builder tool and attempt to bypass the IDS.

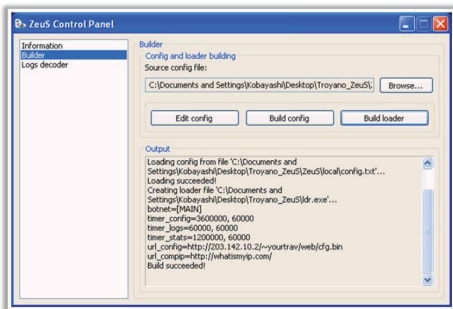


Figure 7.53: Screenshot of Zeus botnet builder tool

▪ ICE IX

Ice IX is a software used to create botnet to steal personal information of a user from a remote system, windows. ICE IX communicates using the HTTP protocol. It is used for various purposes such as manipulating the backing operations on the infected machines, grabbing the data from the target system, and clearing the storage, and so on to attempt to bypass IDS using ICE IX bot.

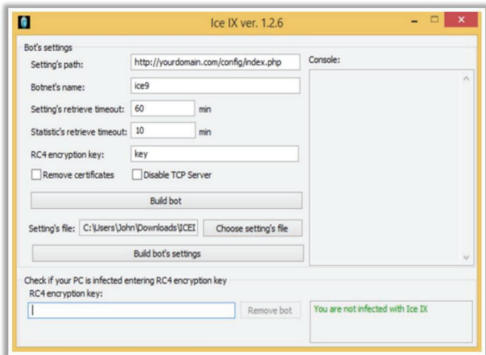


Figure 7.54: Screenshot of ICE IX botnet builder tool

Test the IDS by Sending Inconsistent Packets



In the IP header, the maximum packet length is **65,635 bytes**; Internet Header Length (IHL) is a **4-bit field**, and the header contains a **16-bit (total length) field**

The TCP header has an **Offset field** that specifies the length of header and data, whereas the **UDP header** has a UDP Length field that determines the total size of the UDP packet

Use packet **crafting tools** to send specially crafted **TCP/IP** or **UDP/IP** packets with different TCP/UDP and IP header sizes to the IDS

Packets with **inconsistent** information may bypass the sanity check at the **IDS**

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

Test the IDS by Sending Inconsistent Packets

IP Header used in the internet communication is called an Internet Header Length (IHL) that consists of 4-bit fields. An IP packet consists of the IP header that can be distinguishable using these 4-bit fields from that of the actual data.

In the IP header, maximum packet length is 65,635 bytes and IHL a 4-bit field, and header contains 16-bit total length field. The TCP header has an offset field that specifies the length of header and data, whereas the UDP header has a UDP length field that determines the total size of the UDP packet. You can use packet-crafting tools to send specially crafted TCP/IP or UDP/IP packets with different TCP/UDP and IP header sizes to the IDS. Packets with inconsistent information may bypass the sanity check at the IDS.

Test the IDS for IP Packet Fragmentation



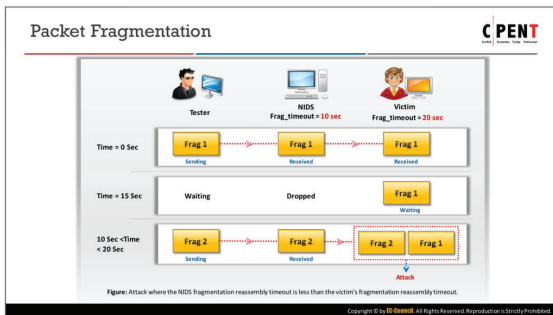
- 1 IP packets must follow standard Maximum Transmission Unit (MTU) size while traveling across the network
- 2 If the packet size exceeds, it splits into **multiple fragments** (called fragmentation) and then is reassembled later
- 3 Send malicious packets to the IDS at **regular interval of time** (greater than the IDS fragmentation reassembly timeout) to attack the target system
- 4 See the **example** in the next slide

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

Test the IDS for IP Packet Fragmentation

IP packets must follow standard maximum transmission unit (MTU) size while traveling across the network. If the packet size exceeds, it splits into multiple fragments (called fragmentation) which are reassembled later.

The IP header contains a fragment ID, fragment offset, fragment length, and fragments flags besides the original data. In a network, the flow of packets is irregular, so the systems need to keep or store the fragments around, wait for future fragments, and then reassemble them in order. You can take advantage of this fragmenting and reassembling process, by sending malicious packets to the IDS at regular intervals of time (greater than the IDS fragmentation reassembly timeout) to attack the target system. This process can help you exploiting the target system without being detected by IDS.



Packet Fragmentation

Example: Attacking where the NIDS fragmentation reassembly timeout is less than the victim's fragmentation reassembly timeout

If, for example, the fragment-reassembly timeout is 10 seconds at the IDS and 20 seconds at the target system, an attacker can send the second fragment after 15 seconds of sending the first fragment. In this scenario, the IDS will drop the fragment upon receiving the second fragment after its reassembly timeout, but the target host will reassemble the fragments. Attacker continues sending fragments with intervals of 15 seconds until the attack payload reassembles at the target system. Thus, the victim will reassemble the fragments and receive the attack code, whereas the IDS will not make any noise or generate alerts as the IDS drops the fragments of packets.

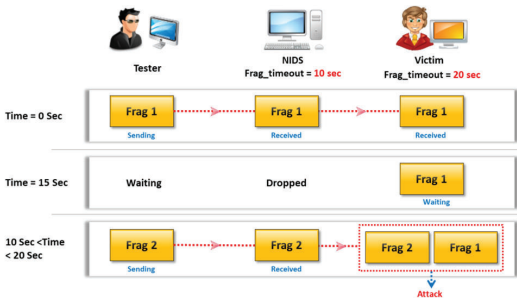


Figure 7.55: Schematic diagram displaying fragmentation attack

Test the IDS for Polymorphic Shellcode



- 1 Most IDSs contain **signatures** for commonly used strings within shellcode
- 2 This is easily bypassed by using **encoded shellcode** containing a stub that decodes the shellcode that follows
- 3 Polymorphic shellcode is a shellcode that contains **hidden malicious code** in an encrypted form
- 4 Use **existing buffer-overflow exploit** to carry polymorphic shellcode, and set the "return" memory address on the overflowed stack to the entrance point of the decryption code
- 5 Check whether the **IDS** identifies the polymorphic shellcode

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

Test the IDS for Polymorphic Shellcode

A signature-based network intrusion detection system (NIDS) identifies an attack by matching attack signatures with incoming and outgoing data packets. Most IDSs contain signatures for commonly used strings with shellcode that can be easily bypassed by using encoded shellcodes containing a stub that decodes the shellcode that follows.

With polymorphic shellcodes, attackers hide their shellcode (malicious code) by encrypting it with an unknown encryption algorithm and including the decryption code as part of the attack packet. To carry out polymorphic shellcode attacks, they use an existing buffer-overflow exploit and set the "return" memory address on the overflowed stack to the entrance point of the decryption code. This makes it difficult for the IDS to identify it as shellcode. Therefore, when attackers modify/transform their attacks in this way, the NIDS cannot identify them. This method also hides the commonly used shellcode strings, thus making the shellcode signatures unusable.

Try to Evade the IDS by Obfuscating or Encoding the Attack Payload



- Obfuscation is the process of creating **obfuscated code** that is difficult for the IDS to understand

- Evade the IDS by obfuscating or **encoding the attack payload** in a way that the target system understands but the IDS does not

- Encode the **attack patterns** in unicode to bypass IDS filters, but be understood by an IIS web server

- Try to manipulate the **path referenced** in the signature to fool the IDS

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

Try to Evade the IDS by Obfuscating or Encoding the Attack Payload

Obfuscation is the process of creating obfuscated code that is difficult for the IDS to understand. Obfuscation makes code harder to understand or read, generally for privacy or security purposes. You can use a tool called as an obfuscator that converts a straightforward program into one that works the same way, but is much harder to understand. Evade the IDS by obfuscating or encoding the attack payload in a way that the target system understands but the IDS does not. Try to manipulate the path referenced in the signature to fool the IDS. Encode the attack patterns in Unicode to bypass IDS filters, but be understood by an IIS web server. You can perform obfuscated attacks on encrypted protocols such as HTTPS.

Check the IDS for False-Positive Generation



I

Within the IDS, check the packets that are **generating the alerts**, and whether it has activated a large number of false reports

II

Examine the large amount of alert data that is generated and **logged** by the IDS

III

To verify the log **data**, it is very difficult to differentiate between **false positives** and **legitimate** attacks

IV

With knowledge of the specific IDS, a tester can generate **false positives**

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

Check the IDS for False-Positive Generation

False positive generation is similar to DoS attack, however, in false positive generation huge amount of alert data is generated. Any person who is an expert at writing specific IDS signatures can generate false positives that can trigger IDS alerts. Just by verifying logs, it is very difficult to differentiate between the false positives and legitimate attacks. As a penetration tester, you have to check the packets that are generating the alerts, and whether it has activated a large number of false reports. Also, examine the large amount of alert data that is generated and logged by the IDS.

Test the IDS for TTL Evasion



In the TTL evasion technique, an IDS **rejects the packets** that an end system accepts

The tester tries to send request packets, which are **mistakenly rejected** by the IDS to remove parts of the stream from the IDS's view

A **malicious host** uses a combination of TTL to fool the IDS and retransmits the fragments to the target host

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

Test the IDS for TTL Evasion

TTL-based attacks are the type of attacks for which you must have the prior knowledge about the topology of the network that the victim has, and also you must know the distance to the end host and whether the IDS is placed in front of the end host. This information can be obtained by using different tools. In the TTL evasion technique, an IDS rejects the packets that an end system accepts. Try to send request packets, which are mistakenly rejected by the IDS, to remove parts of the stream from the IDS's view. A malicious host uses a combination of TTL to fool the IDS, and retransmits the fragments to the target host.

Test the IDS by Sending a Packet to Port 0



For both TCP and UDP, port 0 traffic is considered unusual because it is officially a **reserved port** and should not be used for **network communications**

Port 0 traffic is probably **not legitimate** because the packets are probably generated synthetically

When using port 0, it is recommended that one should **block the traffic** at the firewall

The **Gobbler tool** can be used to test port 0 fingerprinting

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

Test the IDS by Sending a Packet to Port 0

For both TCP and UDP, port 0 traffic is considered unusual, because it is officially a reserved port and should not be used for network communications. The traffic from this port is considered to be unusual and probably not legitimate, because the packets are probably generated synthetically. IDS/IPS identifies such unusual traffic and blocks them using signatures. The network administrator must be aware that if any communication coming from TCP/IP and UDP packets with a source port 0 could specify some ongoing attacks, such as identifying the target's host operating system or spoofing. When using port 0, it is recommended that one should block the traffic at the firewall. Send a packet to port 0 to test the IDS whether it will detect such unusual traffic or not. You can use tools such as Gobbler to test port 0 fingerprinting.

Test the IDS for UDP Checksum



The UDP checksum is only optionally computed if this **16-bit field** is exactly 0; it signifies that the UDP checksum was not computed on **transmission** and should not be checked upon **reception**



Any packets that have the **UDP checksum** turned off are questionable and may be **subtle evasion** attempts



Send UDP packets with a **wrong checksum** to the IDS, and see the IDS response and how it reacts to this packet

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

Test the IDS for UDP Checksum

UDP offers only a nominal transport service, which is a non-guaranteed datagram delivery. It gives direct access applications to the datagram of the IP layer.

Basically, checksums are used to check the data integrity in most of the network protocols, because during the data transmissions the data might altered, missed, or duplicated. Hence, the checksums are calculated at the transmitter and sent along with the data.

The UDP checksum is only optionally computed if this 16-bit field is exactly 0; it signifies that the UDP checksum was not computed on transmission and should not be checked upon reception. Any packets that have the UDP checksum turned off are questionable and may be subtle evasion attempts. Send UDP packets with a wrong checksum to the IDS, and see the IDS response and how it reacts to this packet.

Test the IDS for TCP Retransmissions



- TCP retransmits packets to introduce a level of **reliability** to the unreliable **IP transport mechanism**

- If an IDS sees a **retransmitted** packet (with correct checksums) that has different contents than the **original packet**, it can assume either a buggy **TCP/IP** implementation or a malicious attack

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

Test the IDS for TCP Retransmissions

TCP (Transmission Control Protocol) retransmits packets to introduce a level of reliability to the unreliable IP transport mechanism. Immediately after sending a packet, a host requires an acknowledgment from the destination host whenever it is received. If the sender does not receive any sort of acknowledgment within a certain period of time, then it assumes that the packet did not reach its destination and retransmits the packet. Whenever TCP retransmits a packet, the retransmitted packet must be exactly same as the original packet. Test the IDS for TCP retransmissions. If an IDS sees a retransmitted packet (with correct checksums) that has different contents than the original packet, it can assume either a buggy TCP/IP implementation or a malicious attack.

Test the IDS using Covert Channels



A covert channel can be defined as a **hidden communication** mechanism

When a system has been **compromised** by other means, some hackers will use these covert channels in an attempt to hide their activities

Detect the covert channel's deliberate attempt to make information leaks possible, when bypassing the **security policy** or causing a compromised system to obey an external system

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

Test the IDS using Covert Channels

Covert channel can be defined as a hidden communication mechanism through the established security infrastructure. Covert channels are a means used by attackers to conceal their activities. When a system has been compromised by other means, some hackers will use these covert channels in an attempt to hide their activities. As a pen tester, you need to detect the covert channel's deliberate attempt to make information leaks possible when bypassing the security policy or causing a compromised system to obey an external system.

Test the IDS for Reverse Traversal



- Break apart a **signature**, such as:

`"/cgi-bin/some.cgi"`
by using reverse traversal directory tricks:
`GET /cgi-bin/blahblah/../../some.cgi HTTP/1.0`

- Equates to `"/cgi-bin/some.cgi"` once the directory traversal has been accounted for

- Most IDSs can **detect** this technique

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

Test the IDS for Reverse Traversal

The ideal way for breaking any signature is using reverse traversal directory. This method differs from the URL encoding technique. The raw IDS gives the information that the request contains `"/../../"`. You can break apart a signature, such as `"/cgi-bin/some.cgi"` by using reverse traversal directory tricks such as `"GET /cgi-bin/blahblah/../../some.cgi HTTP/1.0."` This equates to `"/cgi-bin/some.cgi"` once the directory traversal has been accounted for. Most IDSs can detect this technique.

Test for Self-Referencing Directories



- The "." means the parent directory
- "." means the current directory
- So "c:\temp\.\.\.\.\\" is equivalent to:
- "c:\temp\" ("/tmp/././././" being "/tmp/" for you Unix folk)
- This technique will prevent raw IDS systems from matching signatures like "/cgi-bin/phf"

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

Test for Self-Referencing Directories

The new technique that is involved in the directory group is termed as self-referencing directory and "." refers to parent directory, whereas "." refers to the current directory. It can be said that "c:\temp\.\.\.\.\\" is same as "c:\temp\" whereas in Unix "/tmp/././././" is equivalent to "/tmp/." This technique can prevent raw IDS systems from matching signatures like "/cgi-bin/phf"

Test for Premature Request Ending



- Some IDS systems can **stop looking** after the "HTTP/1.0\r\n"

```
GET %20HTTP/1.0%0d%0aHeader:%20../../../../  
cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

- This translates to:

```
GET / HTTP/1.0\r\nHeader: ../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

Or:

```
GET / HTTP/1.0\r\nHeader: ../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

- The IDS will **decode the encoding first** and **stop scanning** at the fake "premature" ending, rather than the real one

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

Test for Premature Request Ending

Many IDS stop scanning a request after HTTP/1.0\r\n in order to save time and processing power. The method of premature request ending is mainly aimed at smart IDS. In order to save time and processing power, a smart IDS could choose a reliable way to detect a scan, check any request, and dump the extra data submitted by the client. Following is an example of a typical request:

```
GET /some.file HTTP/1.0\r\nHeader:                                blah \r\nHeader:                                blah \r\nHeader:                                blah \r\nHeader:                                blah \r\n\r\n
```

Considering the examples of any of the following submissions, there is no value if the headers are scanned by the smart IDS.

```
GET /%20HTTP/1.0%0d%0aHeader: %20../../../../cgi-bin/some.cgi  
HTTP/1.0\r\n\r\n
```

This transforms to:

```
GET / HTTP/1.0\r\nHeader: ../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

Or:

```
GET / HTTP/1.0\r\nHeader: ../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
```

The IDS will decode the encoding first and stop scanning at the fake “premature” ending, rather than the real one. This kind of request or submission is always acceptable. The scanning is halted if the IDS initially decodes the encoded characters. The actual method would be to logically extract the URL that is based on the parsing rules, convert this particular portion, and then search the entire portion for the string that matches.

Test for the IDS Parameter Hiding



- The IDS sometimes does not **scan** the **data** in the **parameters**

- Some IDSs may **stop processing** once the "?" is reached, which indicates the **rest of the data is parameters**:

```
GET/index.htm%3fparam=../cgi-bin/some.cgi HTTP/1.0
```

- This **translates** to:

```
GET/index.htm?param=../cgi-bin/some.cgi HTTP/1.0
```

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

Test for the IDS Parameter Hiding

Designing smart IDS involves submission of parameters with dynamic content. An ID sometimes does not scan the data in the parameters. The smart IDS stops the processing once the "?" is encountered. This not only saves the processing power but also time. This gives an indication that the remaining data are parameters. Consider the following example request:

```
GET /index.htm%3fparam=../cgi-bin/some.cgi HTTP/1.0
```

which transforms to:

```
GET /index.htm?param=../cgi-bin/some.cgi HTTP/1.0
```

In the preceding example, the request was transformed in to a valid format. The appropriate method for parsing is to extract the required part that is the set of encoded characters to be examined before decoding. This is implemented in Whisker v1.3 as -I 5.

Test the IDS for HTTP Misformatting



Some **IDS systems** that implement **minimal signatures** depend on the **trailing space** for matching

For example, matching **"/phf"** could lead to many false positives, but **"/phf "** (notice the trailing space) helps to **ensure** that the final requested page is closer to the actual **"phf"** and not just starting with the letters **"phf"**

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

Test the IDS for HTTP Misformatting

The URL of any request can be extracted by scanning the string without the parameters. This is done by the smart IDS. The v1.0 request of HTTP RFC is:

Method <space> URL <space> HTTP/ Version CRLF

The HTTP protocol segregates each component from another by using the spaces and thereby extracts the required part from the request. Most of the IDSs do not scan the entire request or they implement minimal signatures and depend on trailing space for matching the strings.

For example, matching **"/phf"** could lead to many false positives, but **"/phf "** (notice the trailing space) helps ensure that the final requested page is closer to the actual **"phf,"** and not just starting with the letters **"phf."**

The user has to consider the following HTTP v0.9 syntax:

GET <space> URI CRLF

The v0.9 provides the GET method, which does not return headers. This makes the automatic scanning difficult by CGI scanners.

Test the IDS for Long URLs



- 1 Some IDSs only look within the first **xx bytes** of the request. Generally, this works well since the first line of the **request** needs to contain the **URL**
- 2 However, we can **exploit** this by **submitting** a request along the lines of:
`GET /rfprfp<lots of characters>rfprfp/./cgi-bin/some.cgi HTTP/1.0`
- 3 The key is to include **enough** characters to move the rest of the submitted request, outside the scope of the **ID systems'** scan limit



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

Test the IDS for Long URLs

Some IDSs only look within the first **xx** bytes of the request. Generally, this works well, since the first line of the request needs to contain the URL. However, we can exploit this by submitting a request along the lines of:

`GET /rfprfp<lots of characters>rfprfp/./cgi-bin/some.cgi HTTP/1.0`

The key is to include enough characters to move the rest of the submitted request, outside the scope of the ID systems' scan limit.



Figure 7.56: Screenshot displaying long URL

Test for Null Method Processing



Many **C** string libraries use the **NULL** character to denote the end of the **string**



Some **IDSs** still use these libraries, so the occurrence of using **NULLs** to denote the end of strings is still quite common



We can use this to our **advantage** with the following type of request:
`GET%00 /cgi-bin/some.cgi HTTP/1.0`

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

Test for Null Method Processing

String conclusion is represented with a NULL character in most of the C string libraries. It is the basic character for denoting the end of the string. Some IDSs still use these libraries, so the occurrence of using NULLs to denote the end of strings is still quite common. We can use this to our advantage with the following type of request:

```
GET%00 /cgi-bin/some.cgi HTTP/1.0
```

Try to Bypass the IDS using Compressed Media Files



Applications accessing media files (audio, video, and images) can be **compressed** to **smaller size** for maximizing data transfer rate

IDS cannot **verify** the **signature** of compressed file formats

Embed the attack **code** within the compressed media files and transfer it in smaller parts to the **target system**

Check whether the IDS signatures **identify** the **attack code** within the compressed data, or it allows the packet

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

Try to Bypass the IDS using Compressed Media Files

Media files such as images, audio, and videos can be compressed so that they rapidly transfer as smaller chunks. These applications accessing media files (audio, video, and images) can be compressed to smaller size for maximizing data transfer rate. Since the IDS cannot verify the signature of compressed file formats, you can take advantage of this and attempt to bypass IDS by embedding the attack code within the compressed media files, and transfer it in smaller parts to the target system. Check whether the IDS signatures identify the attack code within the compressed data or it allows the packet.

Test Session Splicing



- Session splicing is an **IDS evasion** technique that exploits how some **IDSs** do not reconstruct **sessions** before performing pattern matching on the data
- Test the IDS systems in use; they could **add delays** between packets to bypass reassembly checking
- Check how the IDSs reassemble communication streams; if a packet is not received in a **reasonable amount of time**, many IDSs stop reassembling and handling that stream
- If the application under attack keeps a **session active** longer than an **IDS** will spend on reassembling it, the IDS will stop
- Check the result: any session after the **IDS stops reassembling** the sessions will be susceptible to malicious data

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

Test Session Splicing

Session splicing is an IDS evasion technique that exploits how some IDSs do not reconstruct sessions before performing pattern matching on the data. Test the IDS systems in use; they could add delays between packets to bypass reassembly checking.

For example, consider the request "GET / HTTP/1.0." This may split across multiple packets as "GE", "T", "/", " ", "H", "T", "TP", "/", "1", " ", "0."

Check how the IDSs reassemble communication streams, if a packet is not received in a reasonable amount of time, many IDSs stop reassembling and handling that stream. If the application under attack keeps a session active for longer than an IDS will spend on reassembling it, the IDS will stop. Check the result: any session after the IDS stops reassembling the sessions will be susceptible to malicious data.

Try to Bypass Invalid RST Packets through the IDS



- The **TCP protocol** adds a checksum to every transmitted segment that is **checked** at the **receiving end**
- When a **checksum differs** from the checksum expected by the receiving host, the TCP protocol **drops the packet** at the receiver's end
- The TCP protocol also **uses an RST packet** to end two-way communications

Process

- Send the RST packet to the IDS with an invalid checksum
- The target system will receive the packet, check the RST packet's checksum, and drop it
- The IDS stops processing the packet thinking that the communication session has ended
- This allows you to communicate with the target system

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

Try to Bypass Invalid RST Packets through the IDS

TCP is one of the core protocols of the Internet Protocol Suite. TCP/IP provides a broad range of communication protocols for various applications on the network. The set of protocols that are included as a part of TCP/IP is called the TCP/IP protocol stack or TCP/IP protocol suite. RST is one of the control bits whose field size is 6 bits, and starts from right to left. The TCP protocol adds a checksum to every transmitted segment that is checked at the receiving end. When a checksum differs from the checksum expected by the receiving host, the TCP protocol drops the packet at the receiver's end. The TCP protocol also uses an RST packet to end a two-way communication.

Bypass invalid RST packets through IDS as follows:

- Send the RST packet to the IDS with an invalid checksum.
- The target system will receive the packet. Check the RST packet's checksum and drop it.
- The IDS stops processing the packet thinking that the communication session has ended.
- This allows you to communicate with the target system.



Assessing Security of Routers

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

Assessing Security of Routers

A router is a device that forwards data packets by determining the destination network point on the internet. These devices are used to direct the network traffic across the network. As a network penetration tester, it is essential for you to assess the security of the installed routers in the organization.

Need for Router Testing





Router testing is needed to provide a single point of reference for **router security** assessment and countermeasures for **identified weaknesses**





You will need to assess **end-to-end router security** with target knowledge and/or without target knowledge



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

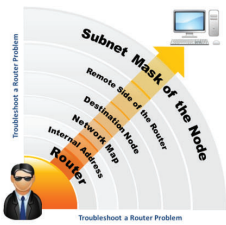
Need for Router Testing

Router testing is needed to provide a single point of reference for router security assessment and countermeasures for identified weaknesses. You will need to assess end-to-end router security with target knowledge and/or without target knowledge.

Router testing is necessary for the following reasons:

- It helps to assess end-to-end router security with or without target knowledge.
- Isolated testing of router is important to validate functionality.
- Router testing helps to analyze, measure, and record the bandwidth and speed of internet connection.
- It can be used for the manipulation, analysis, and recording the bandwidth of internet connection and the speed of the data transfer.
- It determines the reliability and performance of the router. A router's performance decides how efficiently it forwards the packets.
- Testing of router helps to guarantee whether the router is able to handle various network environments.
- It provides a single-point reference for router security assessment and a countermeasure for the identified flaws.

Router Testing Issues



- Test for **misconfigurations** of routers
- Test for **router product-specific** vulnerabilities (example: IOS vulnerabilities in Cisco routers)
- A compromised routing device compromises all **network traffic**
- Without directly compromising a **routing device**, it can be used to compromise the **entire network**
- Routing devices are used to **direct network traffic**, and any one router can be used to **manipulate** network traffic

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

Router Testing Issues

While performing the penetration test on a router for security, you should consider the following issues:

- Test for misconfigurations of routers. A misconfigured router presents itself as the best route to the network, resulting in disorder. The routers that use Border Gateway Protocol (BGP) completely trust the other neighboring networks on the internet without requesting for any digital identification. You can take advantage of this situation to redirect the traffic or wire trap data.
- Test for router product-specific vulnerabilities. Consider the example of the IOS vulnerabilities and HTTP configuration administrative access vulnerability in Cisco routers. These vulnerabilities can be found using vulnerability scanners. The vulnerabilities thus known allow successful attacks through remote access.
- A compromised routing device compromises all network traffic.
- Without directly compromising a routing device, it can be used to compromise the entire network.
- Routing devices are used to direct network traffic, and any one router can be used to manipulate network traffic.

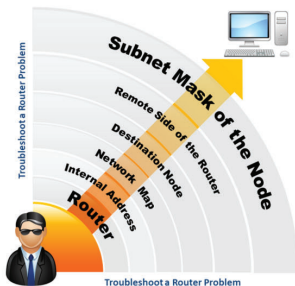


Figure 7.57: Illustration of troubleshooting a router

Identify the Router Hostname



- Identifying router hostnames lets you know which **router you are working on**
- If the router is registered with DNS, a reverse query on the **router's IP address** will give the DNS name of the router
- This DNS name might be the same as the hostname
- Tools:**
 - DIG
 - Nslookup

[illegible]

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

Identify the Router Hostname

Identifying router hostnames lets you know the router on which you are working. If the router is registered with DNS, a reverse query on the router's IP address will give the DNS name of the router. This DNS name might be the same as the hostname.

- **Nslookup**

Source: <https://docs.microsoft.com>

Nslookup is a tool for querying DNS information for host name resolution. It is bundled with both UNIX and Windows operating systems, and can be accessed at the command prompt. When Nslookup is executed, it shows the host name and IP address of the DNS server that is configured for the local system, and then displays a command prompt for further queries. This is the interactive mode. Interactive mode allows the user to query name servers for information about various hosts and domains, or to print a list of hosts in a domain.

```
C:\WINDOWS\system32\cmd.exe - nslookup

C:\Users\test>nslookup
Default Server:  google-public-dns-a.google.com
Address:  8.8.8.8

> ?
Commands:  (identifiers are shown in uppercase, [] means optional)
NAME       - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help on ?  - print info on common commands
set OPTION - set an option
all        - print options, current server and host
[no]debug  - print debugging information
[no]d      - print exhaustive debugging information
[no]defname - append domain name to each query
[no]recurse - ask for recursive answer to query
[no]search - use domain search list
[no]vc     - always use a virtual circuit
domain=NAME - set default domain name to NAME
srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
root=NAME  - set root server to NAME
retry=X    - set number of retries to X
timeout=X  - set initial time-out interval to X seconds
type=X     - set query type (ex. A,AAAA,A+AAAA,ANY,CHNAME,FX,NS,PTR,SOA,SRV)
querytype=X - same as type
class=X    - set query class (ex. IN (Internet), ANY)
[no]mxfr   - use MX fast zone transfer
ixfr=NAME-X - current version to use in IXFR transfer request
lserver=NAME - set default server to NAME, using initial server
root       - set current default server to the root
ls [opt] DOMAIN [-o FILE] - list addresses in DOMAIN (optional: output to FILE)
-a        - list canonical names and aliases
-d        - list all records
-t TYPE   - list records of the given RFC record type (ex. A,CHNAME,FX,NS,PTR etc.)
view FILE - sort an 'ls' output file and view it with pg
exit      - exit the program

>
```

Figure 7.58: Screenshot of Nslookup tool

Port Scan the Router



Port scanning the router pings computers, scans for listening TCP ports, and displays default services and resources that are shared on the network

Scan for the **router's default services**

Port	Service	Protocol
23	Telnet	TCP
80	HTTP	TCP
161	SNMP	UDP

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

Port Scan the Router

Port scanning determines the ports that are open. It also scans for the services that are active and also listens to TCP ports, and displays default services and resources that are shared on the network. Each port is enabled with a unique service. Some of the common ports and services used are Telnet with port number 23, FTP with 21, and so on. Since port scanning identifies all open ports and services that are listening for the connection, as a pen tester, you can use tools such as Nmap to perform port scan on the installed routers to identify any such open ports and services that can pose risk to the organization. On identification of such unused open ports on installed router, you can recommend the organization to disable the services and ports that are not in use.

Scan for the router's default services such as:

Port	Service	Protocol
23	Telnet	TCP
80	HTTP	TCP
161	SNMP	TCP

Table 7.1: Common Ports and Services

Identify the Router Operating System and its Version

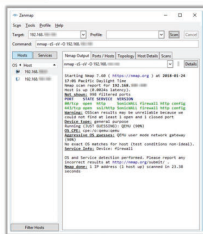


- If you know the router's **operating system** and its **version**, identify the vulnerabilities in the device:

- **Example:** Cisco router model 2500 and IOS version 11.2

- **Tool:** Nmap

- **# nmap -sS -sV -O <router ip address>**



Source: <https://nmap.org>

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

Identify the Router Operating System and its Version

Knowledge of the operating system of the network devices such as routers, firewalls, and so on enables you to identify the common vulnerabilities of that particular operating system and exploit the network device. You can use tools such as Nmap to gather information related to operating system and version of target router.

■ Nmap

Source: <https://nmap.org>

Nmap (network mapper) is a tool to search the network to find state of ports, services running on those ports, the operating systems, firewalls and filters, and so on. You can use the following nmap syntax to perform the OS scan on the target network device.

Syntax: **nmap -sS -sV -O <router ip address>**

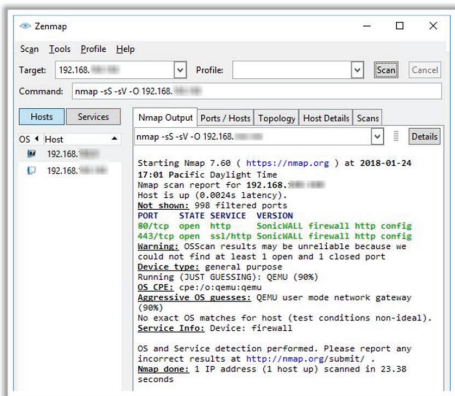


Figure 7.59: Screenshot displaying Nmap (Zenmap) scanning tool

Identify Protocols Running



- Identify the **router protocols** running on the router

Example:

- CDP (Cisco Discovery Protocol)
- RIP (Routing Information Protocol)
- RIPv2 (Routing Information Protocol Version 2)
- IGMP (Internet Group Management Protocol)
- OSPF (Open Shortest Path First)

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

Identify Protocols Running

As a penetration tester, while performing penetration testing on the routers of an organization, it is essential for you to identify the router protocols that are running on the router. These protocols provide crucial information about the routers security infrastructure.

Following are some of the router protocols:

- **CDP (Cisco Discovery Protocol)**

The Cisco Discovery Protocol (CDP) is a self-governing protocol supported by every Cisco networking device. It facilitates you in obtaining information about the adjacent routers directly linked to a router. These details include the uniqueness of the device, hardware platform, the IP address of the adjacent router, and the hold-time for the router. It is activated by default on all routers, regardless of the IOS version used on them. Ensure that it must be deactivated on all routers inside a network.

- **RIP (Routing Information Protocol)**

The Routing Information Protocol (RIP) is a distance vector routing protocol. To choose the route, the RIP uses hop count, which is a router metric. In a RIP network, the maximum hop count is 15. The sixteenth hop is infinity.

You can view the RIP information by using the following commands:

- Syntax: **show ip route**

This command displays and checks IP routes in the routing table of a router. It displays all IP routes that are connected statically or directly to the local router.

- Syntax: **show ip protocol**

This command displays and checks information, such as the protocol used, routing timers, and other information linked with the local router.

- **RIPv2 (Routing Information Protocol Version 2)**

RIPv2 is the version 2 of the distance vector protocol. The protocol includes the ability to carry the subnet information that supports in Classless Inter-Domain Routing (CIDR). RIPv2 supports not only CIDR but also Variable Length Subnet Mask (VLSM). The default version of RIP is version 1. The command version 2 is used to modify RIPv1 to RIPv2.

You can view the RIPv2 information by using the following command:

- Syntax: **Show ip protocols**

This command helps you to view RIP sending and receiving version 2 updates and whether or not automatic summarization is in effect.

- **IGMP (Internet Group Management Protocol)**

The Internet Group Management Protocol (IGMP) allows the dynamic participation of the internet hosts using multitasking. A host group is a group of hosts that uses a specific multicast address. The protocol permits the router to decide the host groups existing on the network.

- **OSPF (Open Shortest Path First)**

The Open Shortest Path First (OSPF) protocol is developed for huge networks with no hop-count constraint. It provides scalability by allowing division of the domain for ease of management. The protocol allows subnetting together with VLSM and non-contiguous subnets.

The OSPF checks the link operation by sending "hello" packets that do not consume excess bandwidth. Depending on the Type of Service (TOS) field, the protocol routes the packets. The protocol has the option to tag the routes through which the packets traverse.

Test for TFTP Connections





Trivial File Transport Protocol (TFTP) uses **UDP for data transfer**, and it is a connectionless protocol, which doesn't support authentication

TFTP is a limited FTP service with **no authentication**





It is commonly used by routers, switches, and other devices to connect to a **TFTP server** during firmware upgrade

In a lot of routers, **TFTP** is used to fetch and push configuration files to these routers



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

Test for TFTP Connections

Trivial File Transport Protocol (TFTP) uses UDP for data transfer and it is a connectionless protocol that doesn't support authentication. TFTP is a limited FTP service with no authentication. It is commonly used by routers, switches, and other devices to connect to a TFTP server during firmware upgrade. In a lot of routers, TFTP is used to fetch and push configuration files to these routers.

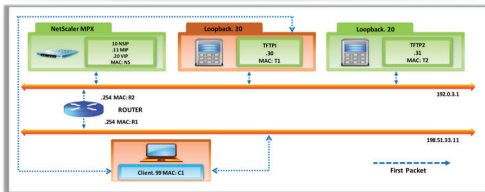
Following are some of the key features of TFTP service:

- It employs UDP (port 69) as a form of transport protocol.
- It cannot enumerate directory contents.
- It has no verification or encoding mechanisms.
- It is employed for reading files from, or writing files to, a remote server.

Try to Retrieve the Router Configuration File



- Try to **sniff** TFTP traffic from the wire
- Try to retrieve the router configuration file using **tftp commands**:
`C:\tftp <tftp server> get <devicename>.cfg`



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

Try to Retrieve the Router Configuration File

Configuration files of a router contain the commands and instructions that drive the functionality of a router. These configuration files can be downloaded or viewed using various protocols such as TFTP, FTP, or RCP. These configuration files provide crucial information about the functioning of the router. You can capture or retrieve these configuration files by sniffing the traffic from the network.

You can use the following TFTP commands to retrieve the router configuration file.

Syntax: `C:\tftp <tftp server> get <devicename>.cfg`

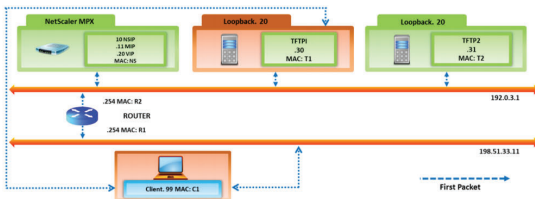


Figure 7.60: Schematic diagram displaying retrieving cfg file through TFTP

Test for Router Misconfigurations



- Router config files give **useful information** to the penetration tester, such as router configuration settings, interfaces, the Access Control Lists (ACLs), SNMP community strings, and router passwords
- Analyze the Router Configuration
- One can **identify** new targets, sensitive systems, or networks by analyzing the ACLs
- Check for **router misconfigurations**
- An attacker can easily gain access to the system if the **router is misconfigured**

[illegible]

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

Test for Router Misconfigurations

Router config files give useful information such as router configuration settings, interfaces, the Access Control Lists (ACLs), SNMP community strings, and router passwords. Analyze the Router Configuration and you can identify new targets, sensitive systems, or networks by analyzing the ACLs. Check for router misconfigurations and if you find any misconfigurations in the router, you can easily gain access to the system. You can use tools such as Rough Auditing Tool for Security (RATS) to check the misconfigurations in the CISCO routers.

```
! version 12.0 service timestamps debug uptime service timestamp log
uptime service password-encryption ! hostname router2 ! logging
buffered errors enable secret 5 $1$sz)o$FYajL33gyTuHm9a8/UfmC1 !
username XYZadmin password 7 05331F35754843001754 ip subnet-zero no ip
routing ! ! interface Ethernet0 description Internal Corporate
Link ip address 10.0.1.199 255.255.255.0 no ip directed-broadcast no
ip route-cache no ip moute-cache ! interface Ethernet1 description
Link to DMZ ip address 170.16.1.1 255.255.255.0 no ip directed-
broadcast no ip route-cache no ip mroute-cache ! interface serial10
description Link from PSInet bandwidth 1536 no ip address no ip
directed-broadcast no fair-queue ! interface serial11 no ip address
no ip directed-broadcast no ip route-cache no ip mroute-cache
shutdown ! ip default-gateway 10.0.1.1 ip http server ip classless !
logging histroy critical logging trap warnings logging 10.0.1.103
access-list 100 permit tcp host 192.168.2.99 host 10.0.1.199 eq finger
access-list 100 permit tcp ip 0.0.0.0 255.255.255.248 host 10.0.1.199
access-list 100 permit ip host 10.0.1.103 any access-list 100 deny ip
any snmp-server community public RO snmp-server community private RW
snmp-server location XYZ widget Inc. server Room (417) snmp-server
contact Network Admins snmp-server host 10.0.1.112 h3rn3c4 banner motd
^C THIS IS A PRIVATE COMPUTER SYSTEM. This computer system including all
related equipment, network devices (specifically including Internet
access), are provided only for authorized use. All compute systems may
be moniroted for all lawful purpose, including to ensure that their use
is authorized personal and their entities to test or verify the security
of the system. During monirotng, information may be examined, recorded,
copied and used for authorized purpose. All information including
personal information, place on or sent over this system may be monitored
Uses of this system, authorized or unauthorized, constitutes consent to
monitoring of this system. Unauthorized use may subject you to criminal
prosecution. Evidence of any such authorized use collected during
monitoring may be used for administrative, criminal or other adverse
aciton. Use of this system constutues consnet to monitoring for these
perposes. ^C ! line con 0 password 7 01030717481C091D25 transport
input none line aux 0 line vty 0 4 password 7 95C4FlA1218000F
login ! end
```

Figure 7.61: Screenshot displaying sample router configuration file

Try to Recover Router Passwords from Config File



01

Locate the **router password** from the config file; passwords could be plain-text or encrypted using Vigenere/MD5 algorithms

02

Use a **router password recovery tool**, such as Router Password Cracker and RouterPassView to decrypt the encrypted router password



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

Try to Recover Router Passwords from Config File

Try to locate the router password from the config file; passwords could be plain-text or encrypted using Vigenere/MD5 algorithms. You can use router-password-recovery tools such as Router Password Cracker and RouterPassView to decrypt the encrypted router password.

- **Router Password Cracker**

Source: <https://securityxplored.com>

Generally, routers or modems control their access by using HTTP BASIC or DIGEST authentication mechanism. In simple words, when you connect to your modem/router from the browser (typically <http://192.168.1.1>), you will be asked to enter username and password. Without this password you will not be able to access your router/modem configuration. Even some websites use this BASIC authentication to allow only certain users to access their site. You can use Router Password Cracker to quickly recover the password and access the configuration file.



Figure 7.62: Screenshot of Router Password Cracker tool



Figure 7.63: Screenshot of Router Password Recovery Report

Test for VTY/TTY Connections



- Try to connect to the router using the **console port**
- You should have **physical access** to the router to try this
- VTY/TTY** connections are used to attach a terminal directly into the router
- In the default configuration of a **router, no security** is applied to the console port



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

Test for VTY/TTY Connections

These VTY/TTY connections will enable you to connect the terminal network devices directly to the router using the console port but to try this, you should have physical access to the router. The console port is not set to the secured mode in the router's default configuration as the setup does not prompt you to do so. This means the VTY/TTY access is insecure. Asynchronous access to the router is possible, which should be checked. To enable connections simultaneously, the 5 ports that are available on the router can be used. To test the VTY/TTY connections, the following information is required:

- IP address of the router
- Telephone number using which the router and modem are connected
- Console access to the router
- Open port to access from the attack point

Try to Gain Access to the Router



- Try standard ports for **Telnet**, **SSH**, and **rlogin**
- Try the other ports found with the **port scan**
- If a modem is connected to the device:
 - Try dialing into the router
- If unsuccessful, try to bring up the terminal window (dial-up setting):
 - `telnet <Device IP address> <standard/high port>`
 - `ssh <Device IP address> <standard/high port>`
- The minimum expected result is a login prompt; if the router is not secured, terminal access will be possible

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

Try to Gain Access to the Router

Accessing a router may allow changes to the network's configuration and can help to secure the network better. To access any router, establish a connection with the router to the computer or network. After the connection is established, open any web browser and type 192.168.1.1, which is the router's IP address, and press Enter. Following are the steps that are required to get access to the router:

- Try standard ports for Telnet, SSH, rlogin commands.
- Search for other ports that may be identified using any port scanning tool.
- During the scan, if any modem is found connected to the device:
 - Try dialing into the router.
- If the attempts made to identify modems go unsuccessful, then try to bring up the terminal window (dial-up setting):
 - `telnet <Device IP address> <Standard/High Port>`
 - `ssh <Device IP address> <standard/high port>`
- The minimum expected result is a login prompt, if the router is not secured, terminal access will be possible.

Test for Router Running Modes





Routers are configured for many **different modes**

Common modes are **"user mode"** and **"privileged mode"**





In **user mode**, the router displays the hostname followed by '>'

Example of **user mode** access:

- TargetRouter >
- Collect the password hash and decrypt it; Cain & Abel can be used to decrypt it



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

Test for Router Running Modes

Routers are configured for many different modes. Common modes of router are "user mode" and "privileged mode."

▪ User mode

In user mode, the router displays the hostname followed by ">". Following is an Example of user mode access:

- **TargetRouter >**
- Collect the password hash and decrypt it; you can use tools such as Cain and Abel to decrypt it.

Privileged Mode Attacks



- 

Commands in user mode are limited. Enable mode is also known as **privileged mode**
- 

To access enable mode, type the following:
• **TargetRouter>enable**
- 

You have fully compromised the router if the password is not configured, and you get the following prompt:
• **TargetRouter#**
- 

If the router prompts you for the password, perform **brute-force password attacks**

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

Privileged Mode Attacks

In the user mode, the commands are limited. Enable mode is also termed as the privileged mode.

- To access the enable mode, type the following command:
 - **TargetRouter>enable**
- You have fully compromised the router, if the password is not configured and you get the following prompt:
 - **TargetRouter#**
- If the router prompts for the password, perform brute-force password attacks.

Test for SNMP Capabilities



1

SNMP is a protocol used to manage routers using management stations such as **HP OpenView** and **IBM Tivoli**

2

Use a port scanning tool such as **Nmap** to check if **UDP port 161** (SNMP) is open and accessible

3

Check for **SNMP** version installed on the machine:
Example: SNMP v1 is insecure and the password is sent as cleartext

4

You can run a tool like **snmpsniff** to extract the **password** from the network when someone connects to the device using **SNMP**

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

Test for SNMP Capabilities

Simple Network Management Protocol (SNMP) is a division of the internet protocol suite as defined by the Internet Engineering Task Force (IETF). The protocol can sustain examination of the network-connected devices for any situation that guarantees managerial attention. The SNMP protocol is a solution for all types of TCP/IP networks. It is useful for the transfer of data from the client location to the server location, where the data is stored in the form of logs for centralized analysis. SNMP is a protocol used to manage routers using management stations such as HP OpenView and IBM Tivoli. Use a port scanning tool such as Nmap to check if UDP port 161 (SNMP) is open and accessible. Also, check for SNMP version installed on the machine, as certain versions are vulnerable, such as the SNMP v1, and so on, which transfer passwords in clear text. You can use tools such as snmpsniff to extract the password from the network when someone connects to the device through SNMP.

Perform SNMP Bruteforcing



🔍 Scan the router for SNMP and use an SNMP brute-force tool to decrypt **SNMP community** strings

Tools to Use

- Engineer's Toolset**
(Source: <https://www.solarwinds.com>)
- ADMSnmp**
(Source: <https://www.freshports.org>)

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

Perform SNMP Bruteforcing

Scan the router for SNMP and use an SNMP brute-force tool to decrypt SNMP community strings. You can use tools such as Engineer's Toolset and ADMSnmp to perform SNMP brute-forcing.

- **Engineer's Toolset**

Source: <https://www.solarwinds.com>

Engineer's Toolset is a collection of over 60 powerful tools that let you discover, configure, monitor, and troubleshoot your network.

Features:

- Automated network discovery
- Real-time monitoring and alerting
- Powerful diagnostic capabilities
- Enhanced network security
- Configuration and log management
- IP address and DHCP scope monitoring

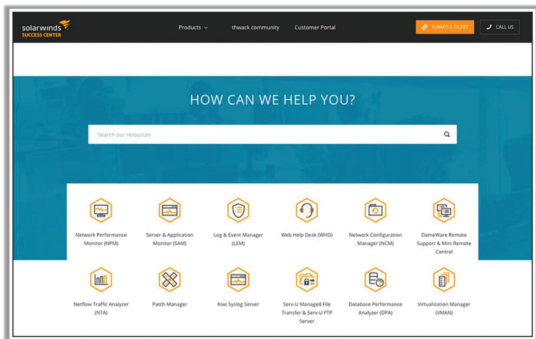


Figure 7.64: Screenshot of Engineer's Toolset

- **ADMSnmp**

Source: <https://www.freshports.org>

ADMSnmp is an snmpd audit scanner and can be used to brute-force the snmp community name (with a wordfile) or make a wordfile list derived the hostname. ADMSnmp can report to you all valid community names found and inform you if writable access to the MIB has been attained.

Try to Log in using default SNMP Community String

The infographic is titled 'Try to Log in using default SNMP Community String' and features the C|PENT logo. It contains three numbered steps: 01, 02, and 03. Step 01 states 'SNMP protocol runs on port 161'. Step 02 states 'Check whether the default SNMP community string "public" is changed at the router. Try to log in using the default community string as "public"'. Step 03 states 'If that does not work, then try brute-force by dictionary attacks'. The background is light blue with large numbers 01, 02, and 03 on the left and right sides respectively.

- 01 SNMP protocol runs on port 161
- 02 Check whether the default **SNMP** community string "public" is changed at the router. Try to log in using the default community string as "**public**"
- 03 If that does not work, then try brute-force by **dictionary attacks**

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

Try to Log in using default SNMP Community String

Simple Network Management Protocol is used to communicate the data among network organization stations and maintain devices. It is also known as SNMP protocol to differentiate it from SNMP framework. It consists of a number of protocol operations that describe the discrete message exchanges that take place between devices and a set of communication mappings that define the way messages are transferred across the network. The Internet Protocol is the ordinary transport mapping used for SNMP. SNMP protocol runs on port 161. Check whether the default SNMP community string "public" is changed at the router. Try to log in using the default community string as "public." A default community string by name "public" is used to login. If that does not work, then you can try brute-force by dictionary attacks.

Test if Finger is Running on the Router



- Finger services expose system user on **port 79 TCP/UDP** by default
- Verify if **finger service** is running on the router
- ```
#finger -l @router-ip-address
```

```
#finger -l Hroot@Hrouter-ip-address
```

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

### Test if Finger is Running on the Router

Finger daemon is a finger service that runs in the background and stores the logged in user information such as username, full name, and so on. Finger services expose system user on port 79 TCP/UDP by default. Firstly, you should verify if finger service is running on the router and then you can abuse the finger daemon functionality by running finger client along with the remote users IP address.

Use **finger -l @router-ip-address** command to enumerate the list of logged in users.

After acquiring the list of users, you can use **finger -l Hroot@Hrouter-ip-address** command to retrieve the logged in user details.

## Test for CDP Protocol Running on the Router



- Cisco Discovery Protocol (CDP) is a layer 2 protocol used by Cisco routers to **discover each other on the same link (segment)**
- The CDP protocol is used to manage **Cisco networks** across the organization
- Using CDP, Cisco routers send out the following messages:



- Cisco routers send these messages out every 30 seconds
- The CDP protocol sends this information to a special MAC address (01:00:0C:CC:CC:CC), and it is received from every Cisco router in the same segment
- CDP is **enabled by default** on Cisco routers

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

## Test for CDP Protocol Running on the Router (Cont'd)



- Use a **"cdp sniffer" command** to find information about the Cisco Discovery Protocol (CDP)

- Disable CDP** if it is not required

- To disable CDP globally, **run the following command:**
  - `#no cdp run`

- To disable CDP on an interface (interface command), **run the following command:**
  - `#no cdp enable`

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

## Test for CDP Protocol Running on the Router

Cisco Discovery Protocol (CDP) is a layer 2 protocol used by Cisco routers to discover each other on the same link (segment). The CDP protocol is used to manage Cisco networks across the organization. Information pertaining to the broadcasting Cisco router is present in the CDP messages. Using CDP, Cisco routers send out the messages given in a below figure:

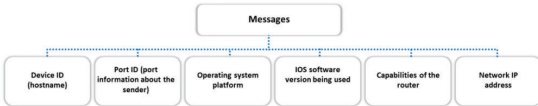


Figure 7.65: Schematic displaying CDP message

By default, CDP is enabled on Cisco routers. The message details that are sent include the MAC address that is visible to all the routers in the segment. The CDP protocol sends this information to a special MAC address (01:00:0C:CC:CC:CC), and it is received from every Cisco router in the same segment. However, the CDP protocol is not routed, and hence the tool is helpful in the local segment. If the default configuration options are not updated, the routers broadcast the messages once in every 30 seconds. Cisco IOS employs the device ID as a tool to identify whether or not the received message is an update and the adjacent routers are already identified.

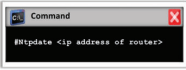
- You can use a “**cdp sniffer**” command to find information about the Cisco Discovery Protocol (CDP).
- Disable CDP if it is not required.
  - To disable CDP globally, run the following command:  
Syntax: **#no cdp run**
  - To disable CDP on an interface (interface command), run the following command:  
Syntax: **#no cdp enable**



## Test for NTP Protocol



- The **Network Time Protocol** (NTP) is often used on border routers, and it is enabled by default
- A lot of companies use the **border router** to synchronize internal servers
- A potential attacker can **corrupt** time if enabled
- Try to **synchronize** the router



```
C:\ Command
#Ntpdate <ip address of router>
```

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

### Test for NTP Protocol

The Network Time Protocol (NTP) is an internet protocol used for synchronizing time across the computer networks and is enabled by default. NTP is often used on border routers, because many companies use these border routers to synchronize their internal servers. NTP is one of the oldest internet protocols, which is designed to be fault tolerant.

NTP protocol is enabled on the router by default. NTP enables synchronization of time between the networks. It is a widely used internet protocol on the network borders and is shared by several company networks.

On the server system with the NTP software, the clock can be set. This time is passed to other systems connected to it. The time is communicated with the systems with time delays. Introducing radio clocks can rectify the communicated time delay. You can change the system clock and can corrupt the time by using following command:

Syntax: `Ntpdate <ip address of router>`



Figure 7.66: Screenshot displaying Ntpdate command

## Test for Loose and Strict Source Routing



📌 The path of the packet (outbound and return) is defined in the packet itself. It is of **two types**:

- Loose source routing
- Strict source routing

### Loose source routing

**Some hops** (routing device) in the path are defined, and the rest of the host as usual

### Strict source routing

**Every hop** (routing device) in the path is defined, from start to end

**Examples:**

Use the ping utility with the source routing options (on Windows):

```
C:>ping -j <hosts> (for loose)
C:>ping -k <hosts> (for strict source routing)
```

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

## Test for Loose and Strict Source Routing

Source routing is a method where the sender specifies a path that a packet should take in order to travel in a network. The path of the packet (outbound and return) is defined in the packet itself. It is used for many purposes, such as mapping the network, trouble shooting, performance, and hacking. Source routing is of two types:

- **Loose source routing**

In this type of routing, one or more hops that the packet must go through are defined by the sender. Some hops (routing device) in the path are defined, and the rest of the host as usual.

- **Strict source routing**

In this type of source routing, the exact path a packet must take is defined. Every hop (routing device) in the path is defined, from start to end.

You can use the ping utility with the source routing options (on Windows):

Syntax:

C:>ping -j <hosts> (for loose)

C:>ping -k <hosts> (for strict source routing)

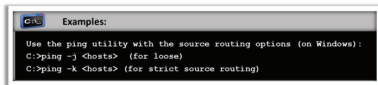


Figure 7.67: Screenshot displaying ping utility with source routing

## Test for IP Spoofing



- By using IP spoofing, an attacker can spoof by **assuming someone's identity**
- If a border router or gateway receives a packet with source as an external IP on its internal interface, or a packet with source as an internal IP on its external interface, the packet is most likely spoofed
- ACLs are used on the router; if **no access control lists** are used then it would be possible to perform IP spoofing

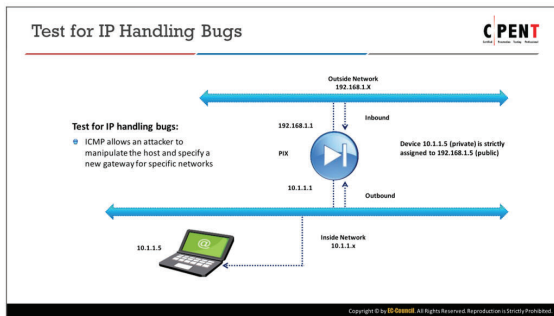
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

### Test for IP Spoofing

IP spoofing is a method where you can access the computer without proper authentication. You can spoof into systems by assuming someone's identity. You can do this by sending an IP address to a computer and making it appear that the message is coming from a trusted host. If a border router or gateway receives a packet with source as an external IP on its internal interface, or a packet with source as an internal IP on its external interface, the packet is most likely spoofed. Generally, ACLs are used on the router, and if no access control lists are used, then it would be possible for you to perform IP spoofing.

Different types of techniques are available for IP spoofing:

- Domain Name System (DNS)
- TCP sequence number prediction
- Packet forging using UDP
- Source routing



### Test for IP Handling Bugs

ICMP redirect helps in notifying and manipulating the hosts on the data link that a better route is available for a particular destination and can also specify new gateways for the specific networks.

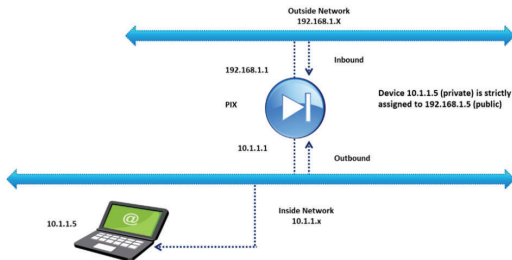


Figure 7.68: Schematic displaying the test for IP handling bugs

## Test ARP Attacks



In switched networks, packets are switched based on **MAC addresses**, and every host on a different network has a unique **48-bit** physical address



ARP requests are sent as **broadcast frames**



Test to determine if **ARP** spoofing is possible against this router

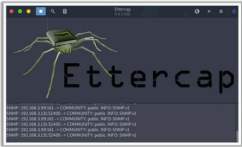


Attempt a **man-in-the-middle attack** against the router



ARP cache poisoning tools:

- Ettercap - <https://sourceforge.net>
- Ufasoft Sniff - <http://ufasoft.com>
- Arpspoof - <https://github.com>



Source: <https://sourceforge.net>

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

## Test ARP Attacks

The Address Resolution Protocol (ARP) maps the IP addresses to hardware addresses used by the data link layer. ARP is a TCP/IP protocol that operates as the interface between the OSI network and OSI link layer, and is located below the network layer. To test the possibility of ARP attacks, you have to attempt the following:

- In switched networks, packets are switched based on MAC addresses, and every host on a different network has a unique 48-bit physical address.
- ARP requests are sent as broadcast frames.
- Conduct a test to find out whether ARP spoofing is possible against this router. ARP spoofing is possible only on a local network.
- Attempt a man-in-the-middle attack against the router.

These types of attacks involve you to gain communication access exchanged between two parties. With the aid of covert channels, you can spoof the identity and can access the information that is interchanged between the original host and the receiving host, and can alter the data without the knowledge of either party. This technique enables you to gain confidential information, thereby violating the robustness of data integrity. This technique can also be referred to as a man-in-the-middle attack.

- **Ettercap**

Source: <https://www.ettercap-project.org>

Ettercap is a comprehensive suite for man-in-the-middle attacks. It features sniffing of live connections, content filtering on the fly, and many other interesting tricks. It

supports active and passive dissection of many protocols and includes many features for network and host analysis.

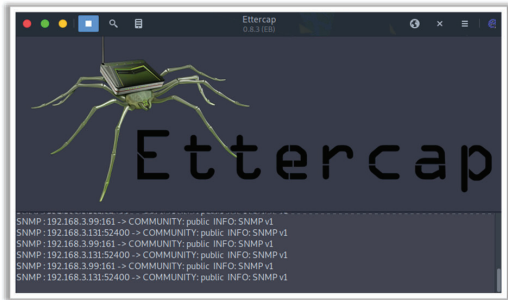


Figure 7.69: Screenshot displaying Ettercap tool

Some of the additional ARP cache poisoning tools include the following:

- Ufasoft Snif (<https://ufasoft.com>)
- Arpspoof (<https://sourceforge.net>)

## Test BGP Protocol



BGP is an external routing protocol which is used to communicate between different **autonomous systems**



A BGP session can be hijacked, and incorrect info about the **routing tables** could be injected with the hijacked session



Session hijacking is easy for someone who can predict the **TCP sequence number** for the TCP session the BGP protocol runs over



Try to hijack BGP sessions using tools, such as **T-sight**

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

### Test BGP Protocol

Border Gateway Protocol (BGP) is an external routing protocol. It enables communication between multiple transit autonomous systems. It is efficient and flexible, and requires less bandwidth. It provides greater administrative efficiency by making use of path attributes that include the AS (autonomous system) numbers along with the information of various routes. It supports internal sessions that run between the same autonomous system and external sessions that run between the routers in different autonomous systems.

BGP sessions can be hijacked, and incorrect information about the routing tables can be injected within the hijacked session. If TCP number sequence that uses BGP protocol is identified, session hijacking attack can be attempted. You can use tools, such as T-sight and Hunt to perform BGP session hijacking.

## Test for EIGRP Protocol

EIGRP is a proprietary routing protocol of **Cisco Systems**

EIGRP authentication works similar to **RIP v2**

EIGRP authentication supports only the **MD5 encryption**

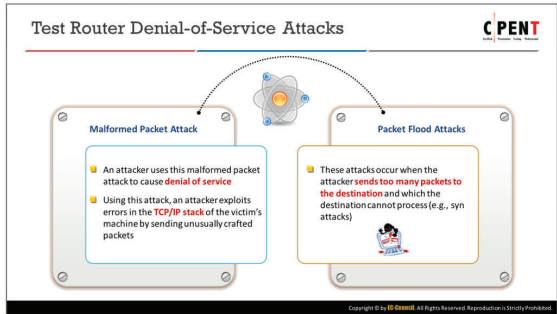
Try to brute-force EIGRP authentication by using **dictionary attacks**

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

### Test for EIGRP Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) is a proprietary routing protocol of Cisco systems. It is an enhanced version of the IGRP. This uses the distance vector method. The efficient and convergence behavior of the architecture of EIGRP has led to its enhancement. It is extended to be an independent network layered protocol. It has the capability to minimize the use of bandwidth and processing power along with the routing instability that occurs after changes in the topology. The maximum hop count of the packets routed by EIGRP is 224. Its authentication works similar to that of RIP v2 and supports MD5 encryption. You can try to brute-force EIGRP authentication by using dictionary attacks.





## Test Router Denial-of-Service Attacks

The denial-of-service attack is an attack on the network or the computer system that creates a disturbance in the services provided by the service provider. This interruption could be a jam in the network traffic by consuming most of the bandwidth or overloading the system's resources.

There are two ways of network denial-of-service attacks (DoS):

- **Malformed packet attack**

You can use this malformed packet attack to cause denial of service. Using this attack, you can exploit errors in the TCP/IP stack of the victim machine by sending unusually crafted packets.

- **Packet flood attacks**

These attacks occur when you send too many packets to the destination that the destination cannot process (e.g., syn attacks).

## Test Router's HTTP Capabilities



Possibly a web server might be running **on the router**  
(Not necessarily port 80 but some other port like 5644)



The new routers can be managed using a **web browser**



Check for the **presence of a web server** by connecting using a web browser

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

### Test Router's HTTP Capabilities

HTTP is used in remote management of router implementation. Possibly a web server might be running on the router (not necessarily port 80 but some other port such as 5644). The new routers can be managed using a web browser. Check for the presence of a web server by connecting to a web browser.

HTTP is presented in clear text and the passwords can be sniffed if an appropriate access-control technique is used. The prerequisite for this is a web-management port listener on the router. The process checks for the router being managed by the HTTP.

## Test for HTTP Configuration Vulnerabilities in Cisco Routers



- HTTP configuration arbitrary administrative access vulnerability enables an attacker to gain full remote administrative access on most of the Cisco devices using affected releases of IOS
- Use a URL of `http://router_address/level/$NUMBER/exec/...` where \$NUMBER is an integer between 16 and 99, and try to gain remote administrative access



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

## Test for HTTP Configuration Vulnerabilities in Cisco Routers

HTTP configuration arbitrary administrative access vulnerability enables an attacker to gain full remote administrative access on most of the Cisco devices using affected releases of IOS.

Use a URL of `http://router_address/level/$NUMBER/exec/...` where \$NUMBER is an integer between 16 and 99, and try to gain remote administrative access.



Figure 7.70: Screenshot displaying output of `http://router_address/level/$NUMBER/exec/...`

## Test through HSRP Attack



- Send packets with **high priority** so that the active router network slows down
- Forward all the incoming packets to the **correct destination**
- Test whether traffic sent via **HSRP group** is forwarded to your **IP address**
- A man-in-the-middle attack is established as all the traffic is forwarded to the **targeted IP**

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

### Test through HSRP Attack

Hot Standby Routing Protocol (HSRP) across router maintains consistency through the use of redundancy. Implementation of this attack requires two routers.

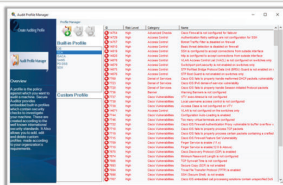
To test the HSRP attack, perform the following steps:

- **Step 1:** Send packets with high priority so that the active router network slows down.
- **Step 2:** Forward all the incoming packets to the correct destination.
- **Step 3:** Test whether the traffic sent via HSRP group is forwarded to your IP address.
- **Step 4:** Check whether the MITM attack is established as all traffic is forwarded to your IP address.

## Router Penetration Testing using Secure Cisco Auditor (SCA)



- The Secure Cisco Auditor is an advanced network **security** software that conducts security assessments and **security auditing** on Cisco **firewalls, routers, and switches**



Source: <http://www.secure-bytes.com>

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

## Router Penetration Testing using Secure Cisco Auditor (SCA)

Source: <http://www.secure-bytes.com>

The Secure Cisco Auditor is an advanced network security software that conducts security assessments and security auditing on Cisco firewalls, routers, and switches.

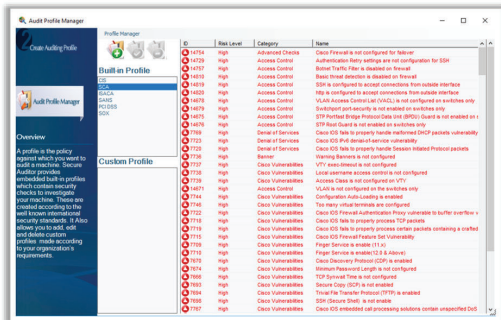


Figure 7.71: Screenshot of Secure Cisco Auditor (SCA) tool



## Assessing Security of Switches

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

### Assessing Security of Switches

The switches are the network communication devices that process the movement of traffic from one system to another in a network. The main purpose of the switches in a network is to provide connectivity, however, if these switches are not properly configured, this may compromise a network and ultimately lead to havoc. It is essential for a penetration tester to secure these switches along with other perimeter devices.

## Look for Security Misconfigurations in Cisco Switch Configuration



- Get the **switch configuration document** and **compare it** with standard security configuration baseline
- Some of the common switch security misconfiguration checks for **CISCO** and **other manufacturers**:
  - Default vulnerable configurations
  - Unused ports
  - DHCP snooping
  - Port security
  - Correct timestamp

```
S1# show port-security interface fastEthernet 0/18
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Last Source Address:Vlan : 0025.83e6.4b01:1
Security Violation Count : 0
```

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

### Look for Security Misconfigurations in Cisco Switch Configuration

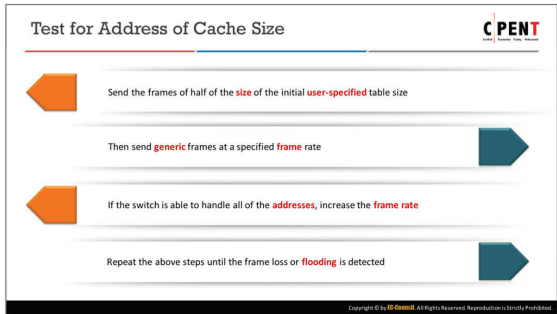
Cisco switches are effectively used in various organizations. Any misconfigurations while employing these switches can reveal crucial information. Get the switch configuration document and compare it with standard security configuration baseline to identify the presence of any misconfigurations.

```
S1# show port-security interface fastEthernet 0/18
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Last Source Address:Vlan : 0025.83e6.4b01:1
Security Violation Count : 0
```

Figure 7.72: Screenshot displaying sample switch configuration document

Following are some of the common switch security misconfiguration checks for CISCO and other manufacturers:

- Default vulnerable configurations
- Unused ports
- DHCP snooping
- Port security
- Correct timestamp



### Test for Address of Cache Size

Switch cache size is the size of the cache memory present in the switch. While performing penetration tests on a network switch, it is essential for you to know the size of the cache memory of a switch and its address table size. In order to test the address of cache size, send the frames of half of the size of the initial user-specified table size. Then send generic frames at a specified frame rate. If the switch is able to handle all of the addresses, increase the frame rate. Repeat the preceding steps until the frame loss or flooding is detected.



## Test for Data Integrity and Error Checking



Check the switch's ability to **forward frames** under certain traffic rates without corrupting the payload

Frames are transmitted with a **predefined data pattern**



Verify whether the switch forwards the frames properly

Calculate the number of **sequence errors** and the number of **data errors**



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

### Test for Data Integrity and Error Checking

To ensure the secure functioning of a switch you should ensure that the switch's data integrity and error checking capability are up to the mark. Check the switch's ability to forward frames under certain traffic rates without corrupting the payload. Check whether frames are transmitted with a predefined data pattern. Verify whether the switch forwards the frames properly. Calculate the number of sequence errors and the number of data errors. Ensure these errors are minimum for secure operations.

## Test for Back-to-Back Frame Capacity



- The back-to-back value is the number of frames in the **longest burst** that the switch will handle without the loss of any frames
- Send a burst of frames with **minimum inter-frame gaps** to the switch and count the number of frames forwarded by the switch
- If the count of **transmitted frames** is equal to the number of frames forwarded, the length of the burst is increased, and the test is **rerun**
- If the number of **forwarded frames** is less than the number transmitted, the length of the burst is reduced, and the test is rerun
- The trial length must be **2 seconds** and should be repeated 50 times with the average of the recorded values being reported

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

### Test for Back-to-Back Frame Capacity

The back-to-back value is the number of frames in the longest burst that the switch will handle without the loss of any frames. The main purpose of this test is to determine the maximum duration that the switch can receive and forward without any frame loss at a maximum traffic rate. Send a burst of frames with minimum inter-frame gaps to the switch and count the number of frames forwarded by the switch. If the count of transmitted frames is equal to the number of frames forwarded, the length of the burst is increased and the test is rerun. If the number of forwarded frames is less than the number transmitted, the length of the burst is reduced and the test is rerun. The trial length must be 2 seconds and should be repeated 50 times with the average of the recorded values being reported.

## Test for Frame Loss



Send a specific number of frames at a **specific rate** through the switch to be tested and count the frames that are transmitted by the switch



The frame loss rate at each point is calculated using the following equation

$$( ( \text{input\_count} - \text{output\_count} ) * 100 ) / \text{input\_count}$$

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

### Test for Frame Loss

Frame loss rate of a switch can be determined by sending a specific number of frames at a specific rate through the switch to be tested and count the number of frames that are transmitted by the switch. Ensure that the frame loss rate of a switch is minimum for a secure functioning of the switch.

The frame loss rate at each point is calculated using the following equation:

$$( ( \text{input\_count} - \text{output\_count} ) * 100 ) / \text{input\_count}$$

## Test for Latency



 Send a stream of frames through the **switch** at the determined **rate** to a specific destination for a duration of **120 seconds**

 Provide an identifying tag in one frame after **60 seconds**

 **Record** the time at which this frame is fully **transmitted** (timestamp A)

 Record the **time** at which the **tagged** frame was received by a **receiver** (timestamp B)

 The latency is **timestamp B** minus timestamp A

 Repeat the test **20 times** with the reported value being the average of the **recorded** values

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

### Test for Latency

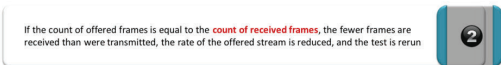
Latency is considered as the time delay that takes place before any operation is performed or transfer of data begins. This latency should always be minimum for fast and effective data transfer in switches. The purpose of this test is to determine the time required to forward a frame to every broadcast node by a switch. Latency may not be same for all switches because there is no specified way for broadcasting frames.

To test the latency of any switch, send a stream of frames through the switch at the determined rate to a specific destination for a duration of 120 seconds. Provide an identifying tag in one frame after 60 seconds. Then record the time at which this frame is fully transmitted (i.e., timestamp A). Also record the time at which the tagged frame was received by a receiver (i.e., timestamp B). Now, the latency can be calculated just by subtracting timestamp B minus timestamp A. Repeat the same test 20 times with the reported value being the average of the recorded values.

## Test for Throughput



Send a specific number of frames at a specific rate through the switch and then count the frames that are **transmitted by the switch**



If the count of offered frames is equal to the **count of received frames**, the fewer frames are received than were transmitted, the rate of the offered stream is reduced, and the test is rerun

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

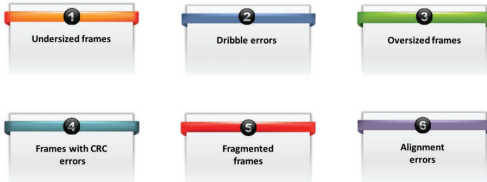
### Test for Throughput

Throughput is considered as the maximum rate at which the data is transferred in a channel. Throughput often affects the bandwidth of a network channel. The purpose of the test is to determine the maximum rate at which frames are transmitted to switch. To test the throughput of a switch over a network, send a specific number of frames at a specific rate through the switch and then count the frames that are transmitted by the switch. If the count of offered frames is equal to the count of received frames, the fewer frames are received than were transmitted, the rate of the offered stream is reduced and the test is rerun.

## Test for Frame Error Filtering



- Check if the switch correctly filters illegal frames, such as



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

### Test for Frame Error Filtering

A switch should filter out malicious and unauthorized frames to ensure safety. Check if the switch correctly filters illegal frames, such as:

- **Undersized frames**  
Frames that are less than 64 bytes and are being propagated must be filtered by switch. These frames should not be forwarded.
- **Dribble bit errors**  
Frames with dribbling bits must be corrected and forwarded by the switch. Frames that do not end in an octet boundary but have a valid check sequence must be accepted and corrected by the switch before forwarding.
- **Oversized frames**  
Switch must filter the frames that are oversized or larger than 1518 bytes.
- **Frames with CRC errors**  
Switch must filter the frames with error in check sequence validation. These frames should not be forwarded.
- **Alignment errors**  
Combination of both CRC error and dribble bit error is called alignment error. Frames with CRC errors and frames that do not end in an octet boundary are not to be transmitted by the switch.
- **Fragmented frames**

## Test for Fully Meshed Condition



Check the total number of **IP frames** that the **switch** can handle when it receives frames on all its **ports**

Each port in the **test** sends frames to all other **ports** in an evenly distributed, **round-robin** type fashion at a specific **user-defined** rate

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

### Test for Fully Meshed Condition

This test determines the total number of IP frames that a switch can handle when it receives traffic on all its ports. To test the switch for fully meshed conditions, check the total number of IP frames that the switch can handle when it receives frames on all its ports. Each port in the test sends frames to all other ports in an evenly distributed, round-robin type fashion at a specific user-defined rate.

## Functional Test for Stateless QoS



Measure the **baseline performance** of the switch:

- With QoS
- Without QoS

Measure and record:

- When QoS is **disabled** on the switch
- When QoS with **IP Precedence** classifying, and marking are enabled on the switch

Inject **stateless traffic** into the network

Check the **latency** and the **packet loss** on the egress traffic port

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

### Functional Test for Stateless QoS

This test is to determine the performance of the switch when stateless traffic is injected into the network with and without QoS. Stateless traffic is of type Layer 2–3 data and does not match true user application traffic. This test verifies that the latency and the packet loss on the egress traffic port degrades when QoS is enabled on the receiving switch.

Following are the steps to perform the functionality test for stateless QoS:

- Measure the baseline performance of the switch:
  - With QoS
  - Without QoS
- Measure and record:
  - When QoS is disabled on the switch
  - When QoS with IP Precedence classifying and marking are enabled on the switch
- Inject stateless traffic into the network
- Check the latency and the packet loss on the egress traffic port



## Performance Test for Spanning Tree Network Convergence

**CENT**  
Certified Penetration Testing Professional

Measure

- **Network** convergence based on the handling of **topology** change notifications
- Configuration **BDPUs**, as well as **traffic** switchover

**Performance**

- Check the **switch spanning tree** convergence performance

**Path Cost**

- Check for any **changes** in path cost to **root** **changes**

**Bridge**

- Check if the **bridge link** slows down

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

### Performance Test for Spanning Tree Network Convergence

Spanning Tree Protocol avoids loops in switched environments to prevent the network from becoming saturated and prevents the occurrence of broadcast storms. This can raise inconsistency in MAC table and lead to network crash.

As a penetration tester, you need to measure the network convergence based on the handling of topology change notifications. Also, measure the configuration of BDPUs as well as traffic switchover. To test the performance, check the switch spanning tree convergence performance. To test the path cost, check for any changes in path cost to root changes. To test the bridge, check if the bridge link slows down.

## Test for OSPF Performance



- Set the defined routes and a **topology**
- Test the **no-drop** throughput and latency
- Execute the test either with **OSPFv2** or **OSPFv3** protocols
- Measure the **OSPF** performance and scalability of a **switch**

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

### Test for OSPF Performance

Open Shortest Path First Protocol avoids excessive consumption of the bandwidth. To test the switch for this OSPF performance, set the defined routes and a topology. Test the no-drop throughput and latency of the switch. Execute the test either with OSPFv2 or OSPFv3 protocols. Then measure the OSPF performance and scalability of a switch.

## Test for VLAN Hopping



- 1 Spoof your computer to appear as another **switch**
- 2 Check whether the real switch turns on the **802.1Q** trunk
- 3 Check all traffic for all **VLANs** sent to your computer
- 4 Send a fake **DTP** negotiate message announcing that you would like to be a **trunk**

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

### Test for VLAN Hopping

VLAN hopping relies on Dynamic Trunking Protocol (DTP). DTP can be used to negotiate between two switches by IEEE 802.1Q trunking standard. In order to test the switch for VLAN hopping, spoof your computer to appear as another switch. Check whether the real switch turns on the 802.1Q trunk. Check all traffic for all VLANs sent to your computer. Then send a fake DTP negotiate message announcing that you would like to be a trunk.

## Test for MAC Table Flooding



Use the **macof tool** for flooding the content addressable memory (CAM) with random MAC addresses

Check whether **all ports are flooded**

Check whether you can **sniff in a switched environment**

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

### Test for MAC Table Flooding

To test the switch for MAC table flooding attack, you can flood the switch interface with large number of frames with various fake MAC address. You can use the macof tool for flooding the content addressable memory (CAM) with random MAC addresses. Check whether all ports are flooded. Check whether you can sniff in a switched environment.

## Testing for ARP Attack



- ➔ Check all the **frames** being sent to the **host address**
- ➔ Associate your **MAC address** with the host MAC address on the **MAC address table** of the **switch**
- ➔ Check the **MAC address** of another host
- ➔ Send a **spoofed Address Resolution Protocol** reply toward another host

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

### Testing for ARP Attack

ARP is a stateless protocol, designed to link logical IP addresses with MAC addresses. Check all the frames that are being sent to the host address. Associate your MAC address with the host MAC address on the MAC address table of the switch. Check the MAC address of another host. Send a spoofed Address Resolution Protocol reply toward another host.

## Check for VTP Attack



- Eliminate all the VLANs by using **VTP** (VLAN Trunking Protocol)
- Check whether you are on the same **VLAN** as every other **user**
- Change your **IP** to be on the same **network** on which the other users are present
- Check whether you can attack the **host**

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

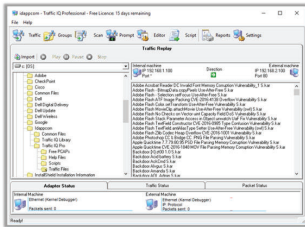
### Check for VTP Attack

VLAN Trunking Protocol is a protocol that manages and reduces the complexity of configuration in a switched environment. Eliminate all the VLANs by using VTP (VLAN Trunking Protocol). Check whether you are on the same VLAN as every other user. Change your IP to be on the same network on which the other users are present. Now, check whether you can attack the host.

## Router and Switch Security Auditing Tool: Traffic IQ Professional



- Traffic IQ Professional enables security professionals to **audit and validate the behavior of security devices** by generating **standard application traffic or attack traffic** between two virtual machines
- It can be used to **assess, audit**, and test the behavioral characteristics of any **non-proxy** packet-filtering device, including:
  - Application-layer firewalls
  - Intrusion detection systems
  - Intrusion prevention systems
  - Routers and switch



Source: <http://www.idogg.com>

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

## Router and Switch Security Auditing Tool: Traffic IQ Professional

Source: <https://www.idappcom.com>

Traffic IQ Professional enables security professionals to audit and validate the behavior of security devices by generating standard application traffic or attack traffic between two virtual machines. It can be used to assess, audit, and test the behavioral characteristics of any non-proxy packet-filtering device including:

- Application-layer firewalls
- Intrusion detection systems
- Intrusion prevention systems
- Routers and switches

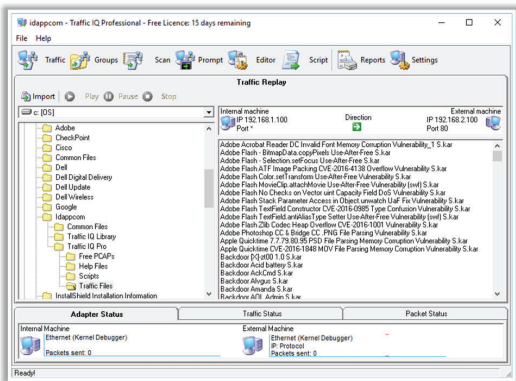


Figure 7.73: Screenshot of Traffic IQ Professional tool



## Document the Result



- Note down the flaws found in perimeter devices
- Firewall rules though which evasion has been possible
- IDS rules though which evasion has been possible
- Configuration mistakes found in router and switches

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

### Document the Result

- Note down the flaws found in perimeter devices
- Firewall rules though which evasion has been possible
- IDS rules though which evasion has been possible
- Configuration mistakes found in router and switches