CASE STUDY REPORT

18CSE386T PENETRATION TESTING AND VULNERABILITY ASSESSMENT

Submitted by

ISHAAN SINGH ARORA(RA2111030010136)

Under the Guidance of

Dr. Deepika D
Assistant Professor

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in Cyber Security

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2024

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

SCHOOL OF COMPUTING
College of Engineering and Technology
SRM Institute of Science and Technology

CASE STUDY ON “SECURITY ASSESSMENT FOR PUBLC DOMAIN”

EVEN Semester (2023-2024)

Course Code & Course Name: 18CSE386T – Penetration Testing and Vulnerability Assessment

Year & Semester : III/VI

Report Title : Security Assessment For Public Domain

Course Faculty **:** Dr. Deepika D

Student Name : (RA2111030010136) ISHAAN SINGH ARORA

Evaluation:

S.No	**Parameter**	**Marks**
1	Problem Investigation & Methodology Used	/5
2	Tool used for investigation	/5
3	Demo of investigation	/5
4	Uploaded in GitHub?	/5
5	Viva	/5
6	Report	/5
	Total	**/30**

Date :

Staff Name :

Signature :

CASE STUDY ON “SECURITY ASSESSMENT FOR PUBLC DOMAIN”

INTRODUCTION:

In the realm of security assessments conducted in the public domain, a diverse range of methodologies and tools are employed to evaluate the robustness of systems, networks, and infrastructures against potential threats and vulnerabilities.

These assessments are crucial for identifying weaknesses, enhancing resilience, and safeguarding critical assets from malicious actors.

SCOPE:

Comprehensive Evaluation: Security assessments in the public domain involve comprehensive evaluations of systems, networks, and infrastructures to identify vulnerabilities and assess their susceptibility to various threats.

Diverse Methodologies: Security assessments utilize a variety of methodologies such as vulnerability scanning, penetration testing, code reviews, and configuration audits to uncover vulnerabilities and assess the overall security posture.

Regulatory Compliance: Public domain security assessments often include assessments against regulatory frameworks and industry standards to ensure compliance with legal and regulatory requirements, such as GDPR, HIPAA, PCI DSS, etc.

Risk Mitigation Strategies: Security assessments not only identify vulnerabilities but also provide recommendations and strategies for mitigating risks and enhancing security controls to protect against potential threats.

OBJECTIVE:

1. Data Collection: In security assessments within the public domain, the primary objective is to gather comprehensive information about the target systems, networks, and infrastructure. This involves conducting thorough reconnaissance activities to identify potential vulnerabilities, configuration weaknesses, and areas of concern.

2. Identification of Weaknesses: Security assessments aim to identify and exploit weaknesses within the target environment, including but not limited to software vulnerabilities, misconfigurations, inadequate access controls, and lack of security protocols.

3. Evaluation of Defenses: Assessments seek to evaluate the effectiveness of existing security defenses, including firewalls, intrusion detection/prevention systems, antivirus solutions, and encryption mechanisms, to determine their ability to withstand various attack scenarios.

4. Adherence to Best Practices: Security assessments assess the extent to which the target organization adheres to industry best practices and standards for information security, such as ISO 27001, NIST Cybersecurity Framework, and CIS Controls, in order to ensure a robust security posture.

5. Simulation of Attack Scenarios: Security assessments simulate real-world attack scenarios to test the resilience of the target environment against different threat actors and attack vectors. This includes conducting penetration testing, social engineering simulations, and red team exercises to identify vulnerabilities and weaknesses that may be exploited by adversaries.

6. Continuous Improvement: Security assessments provide recommendations and remediation strategies to address identified vulnerabilities and enhance the overall security posture. Additionally, they emphasize the importance of ongoing monitoring, testing, and improvement to adapt to emerging threats and evolving security challenges.

TOOL USED AND APPLICATION TO PERFORM:

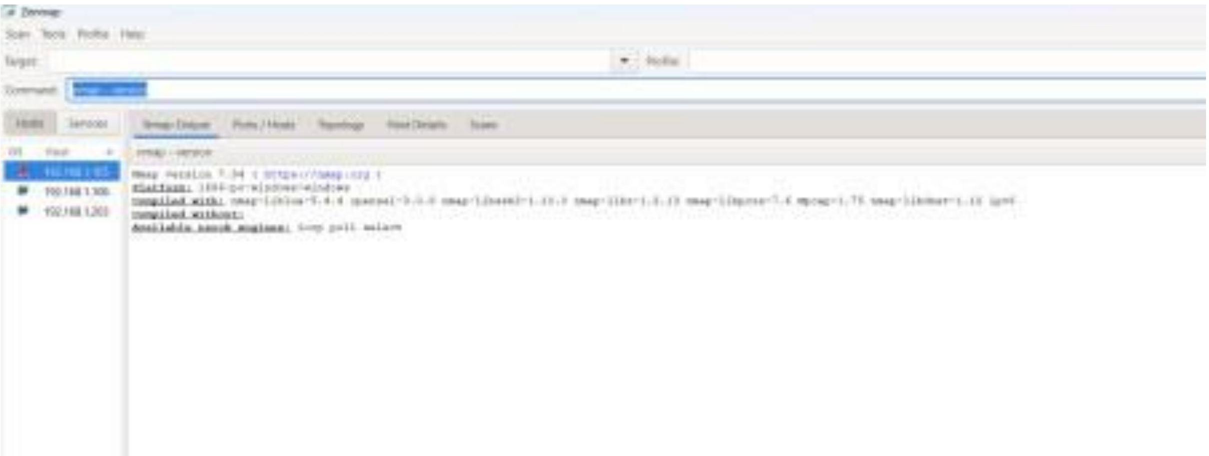
Nmap (Network Mapper)

Nmap (Network Mapper) is a powerful open-source tool used for network discovery and security auditing. It allows users to discover hosts and services on a computer network, thus creating a map of the network's topology.

TOP NMAPS COMMANDS:-

1. *TO FIND THE VERSION OF NMAP:*

   Syntax: nmap –version

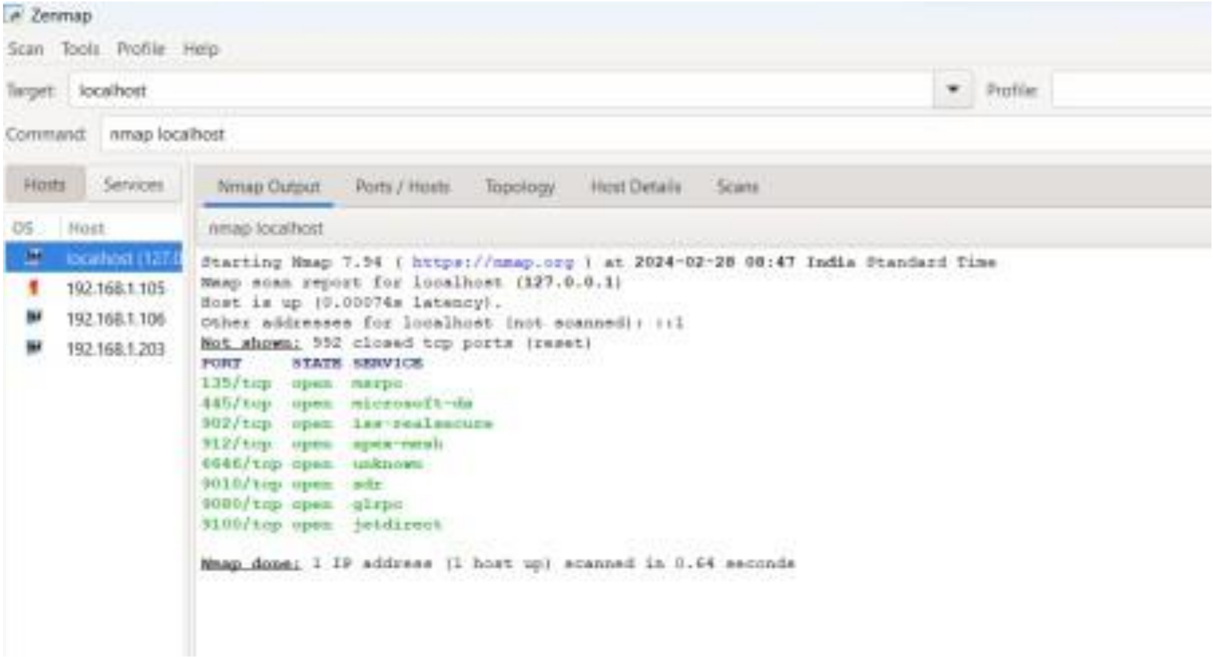


Command: nmap --version

```
nmap --version
Nmap version 7.94 ( https://nmap.org )
Platform: i686-pc-windows-windows
Compiled with: nmap-liblua-5.4.4 openssl-3.0.8 nmap-libssh2-1.11.0 nmap-libz-1.2.13 nmap-libpcre-7.6 npcap-1.78 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: iocp poll select
```

2. *TO SCAN THE IP ADDRESS OR HOST NAME:*

Syntax: nmap localhost or nmap <target>



Target: localhost

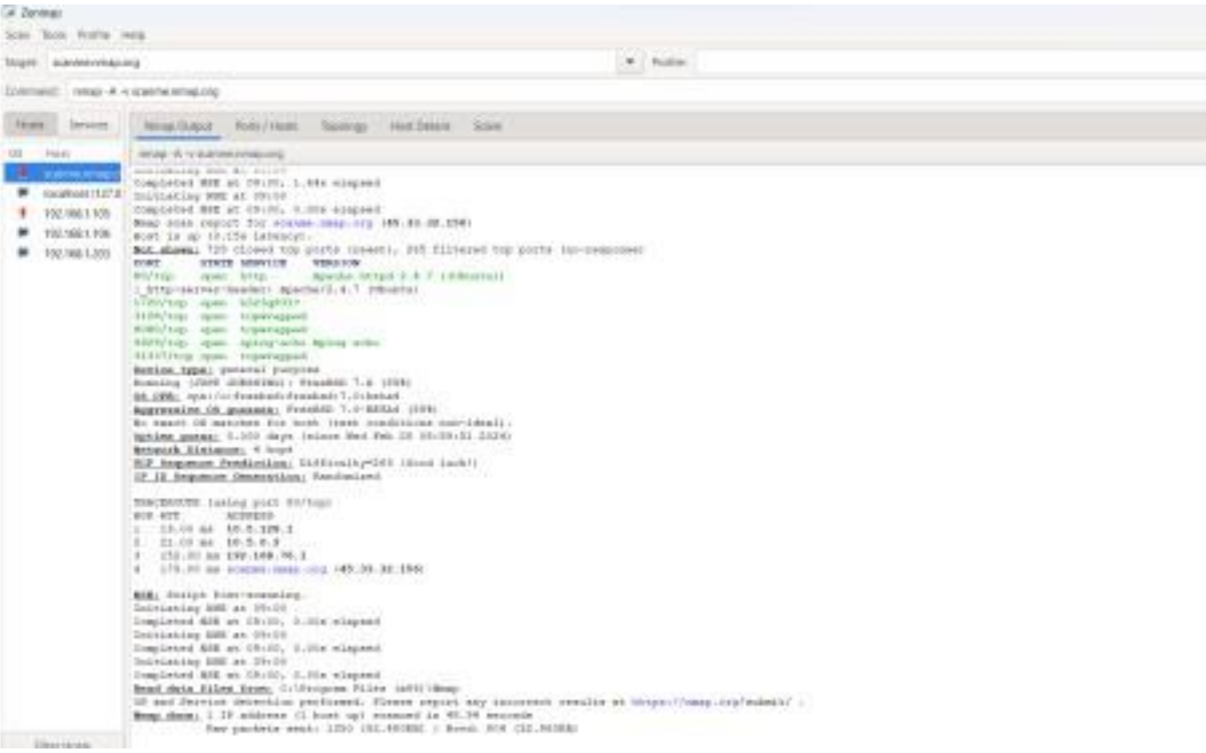
Command: nmap localhost

```
nmap localhost
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 08:47 India Standard Time
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00074s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 992 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
6646/tcp open  unknown
9010/tcp open  sdr
9080/tcp open  glrpc
9100/tcp open  jetdirect

Nmap done: 1 IP address (1 host up) scanned in 0.64 seconds
```

3. Information out of the system: syntax:

nmap -v -A scanme.nmap.org



Command: nmap -A -v scanme.nmap.org

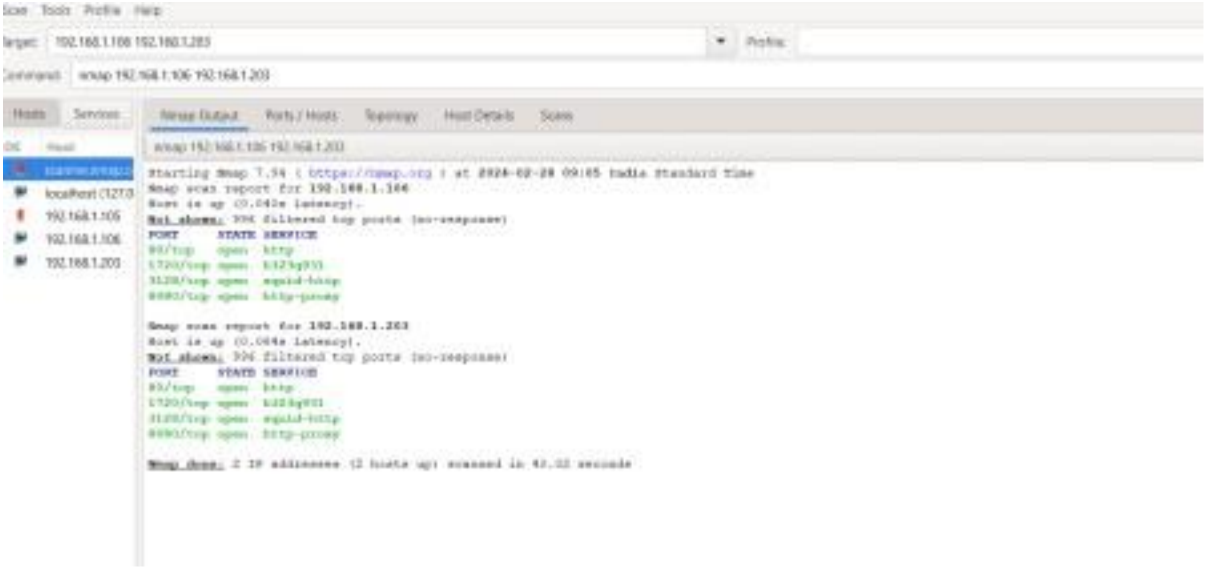
```
nmap -A -v scanme.nmap.org
Initiating NSE at 09:07
Completed NSE at 09:08, 1.44s elapsed
Initiating NSE at 09:08
Completed NSE at 09:08, 0.00s elapsed
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.15s latency).
Not shown: 720 closed tcp ports (reset), 275 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
1720/tcp  open  h323q931?
3128/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Device type: general purpose
Running (JUST GUESSING): FreeBSD 7.X (88%)
OS CPE: cpe:/o:freebsd:freebsd:7.0:beta4
Aggressive OS guesses: FreeBSD 7.0-BETA4 (88%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.000 days (since Wed Feb 28 09:08:51 2024)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Randomized

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   15.00 ms  10.5.128.1
2   21.00 ms  10.5.0.3
3   152.00 ms 192.168.96.1
4   179.00 ms scanme.nmap.org (45.33.32.156)

NSE: Script Post-scanning.
Initiating NSE at 09:08
Completed NSE at 09:08, 0.00s elapsed
Initiating NSE at 09:08
Completed NSE at 09:08, 0.00s elapsed
Initiating NSE at 09:08
Completed NSE at 09:08, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.39 seconds
           Raw packets sent: 1250 (56.480KB) | Rcvd: 808 (32.960KB)
```

4. *SCAN MULTIPLE IP ADDRESS OR SUBNET (IPV4):*

Syntax: nmap <target1> <target2> …..



Command: nmap 192.168.1.106 192.168.1.203

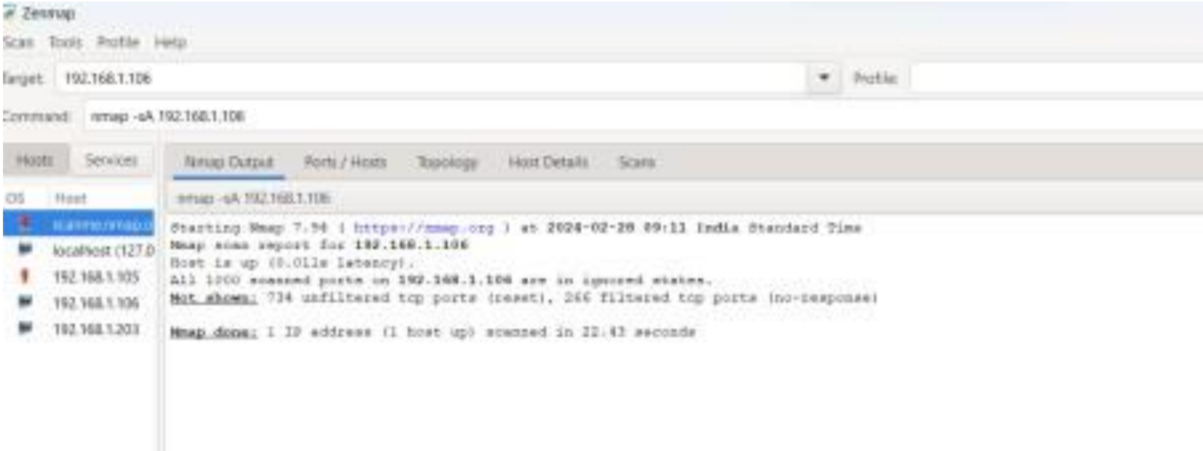
```
nmap 192.168.1.106 192.168.1.203
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:05 India Standard Time
Nmap scan report for 192.168.1.106
Host is up (0.042s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931
3128/tcp  open  squid-http
8080/tcp  open  http-proxy

Nmap scan report for 192.168.1.203
Host is up (0.064s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931
3128/tcp  open  squid-http
8080/tcp  open  http-proxy

Nmap done: 2 IP addresses (2 hosts up) scanned in 42.22 seconds
```

5. *FIND OUT IF A NETWORK/HOST IS PROTECTED THROUGH FIREWALL*

Syntax. nmap -sA <target>



Target: 192.168.1.106

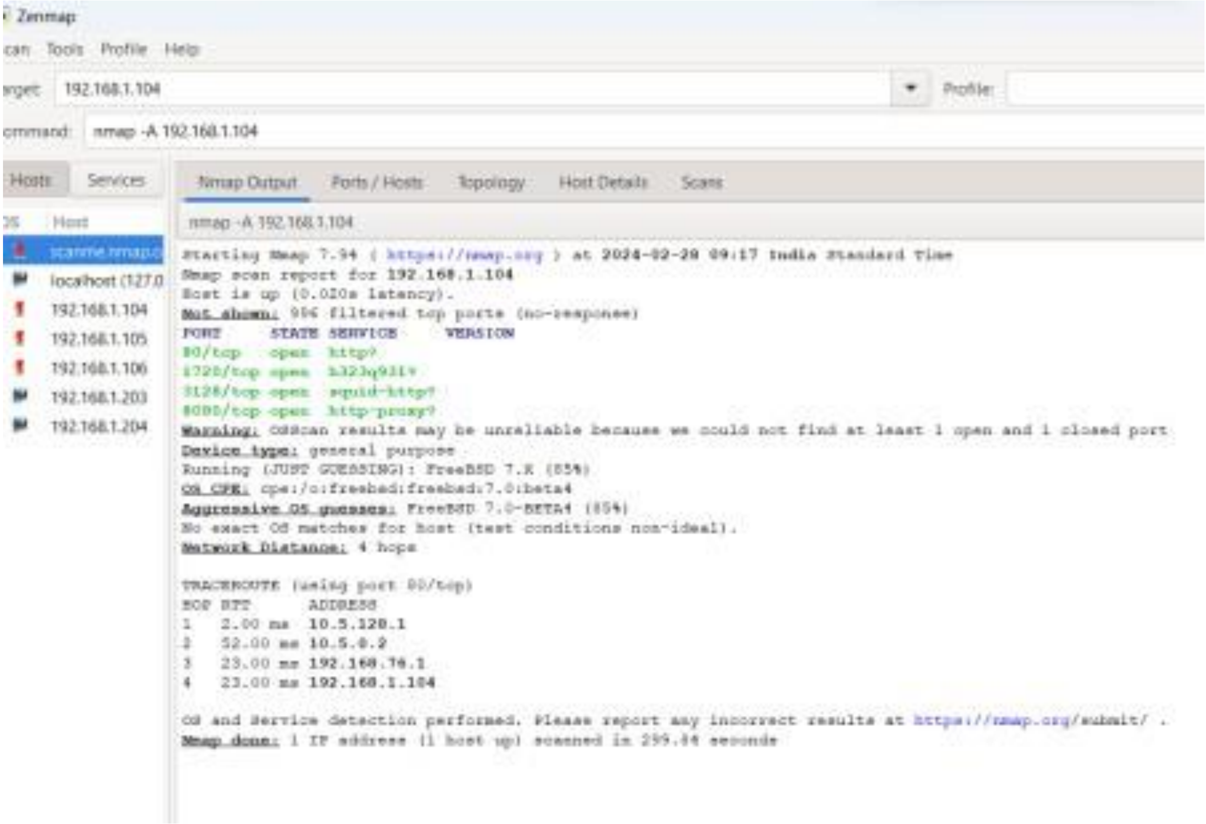
Command: nmap -sA 192.168.1.106

```
nmap -sA 192.168.1.106
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:11 India Standard Time
Nmap scan report for 192.168.1.106
Host is up (0.011s latency).
All 1000 scanned ports on 192.168.1.106 are in ignored states.
Not shown: 734 unfiltered tcp ports (reset), 266 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 22.43 seconds
```

6. *TURN ON OS AND VERSION DETECTION SCANNING SCRIPT (IPV4):*

Syntax: nmap -A <target>



rget: 192.168.1.104

ommand: nmap -A 192.168.1.104

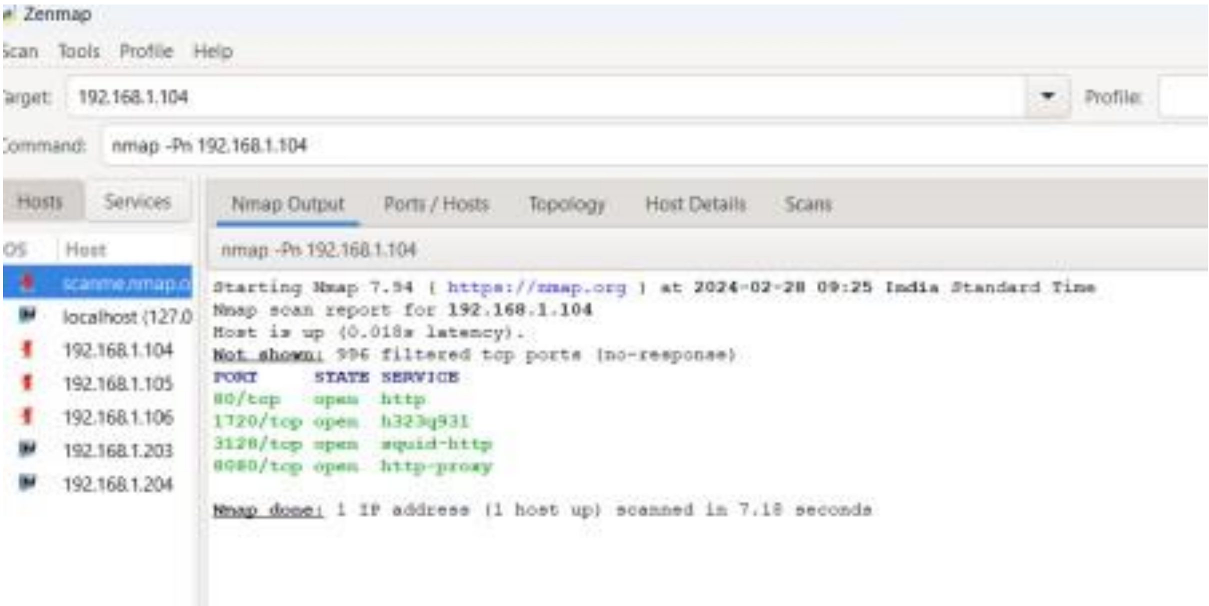
```
nmap -A 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:17 India Standard Time
Nmap scan report for 192.168.1.104
Host is up (0.020s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE     VERSION
80/tcp    open  http?
1720/tcp open  h323q931?
3128/tcp open  squid-http?
8080/tcp open  http-proxy?
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): FreeBSD 7.X (85%)
OS CPE: cpe:/o:freebsd:freebsd:7.0:beta4
Aggressive OS guesses: FreeBSD 7.0-BETA4 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   2.00 ms  10.5.120.1
2   52.00 ms 10.5.0.2
3   23.00 ms 192.168.76.1
4   23.00 ms 192.168.1.104

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 299.84 seconds
```

7. *SCAN A HOST WHEN PROTECTED BY FIREWALL;*

Syntax: nmap -PN <target>



Target: 192.168.1.104

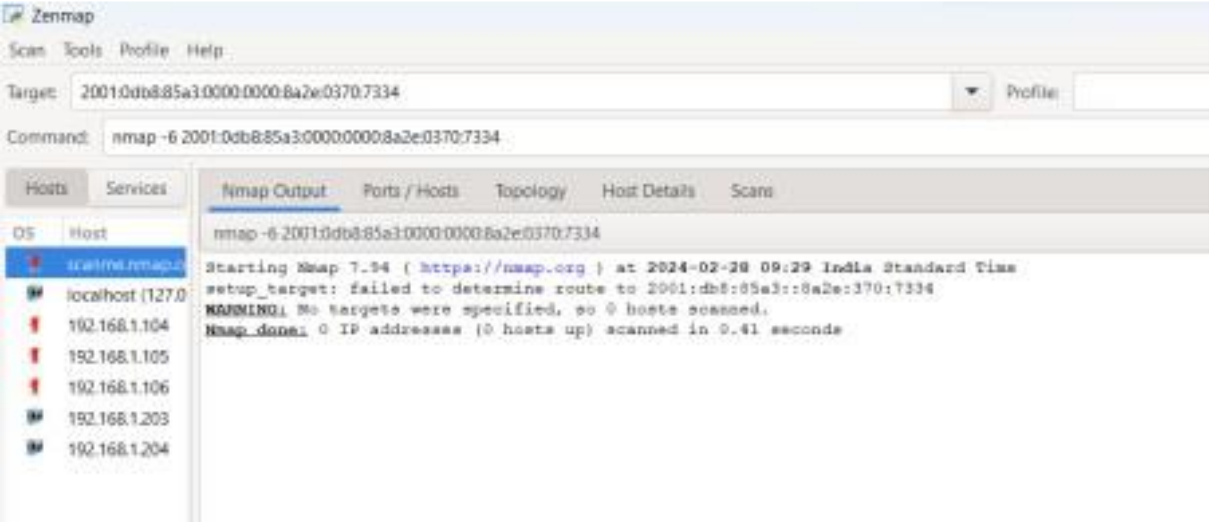
Command: nmap -Pn 192.168.1.104

```
nmap -Pn 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:25 India Standard Time
Nmap scan report for 192.168.1.104
Host is up (0.018s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
1720/tcp open  h323q931
3128/tcp open  squid-http
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 7.18 seconds
```

8. *SCAN AN IPV6 HOST/ADDRESS:*

Syntax: nmap -6 <ipv6 address>



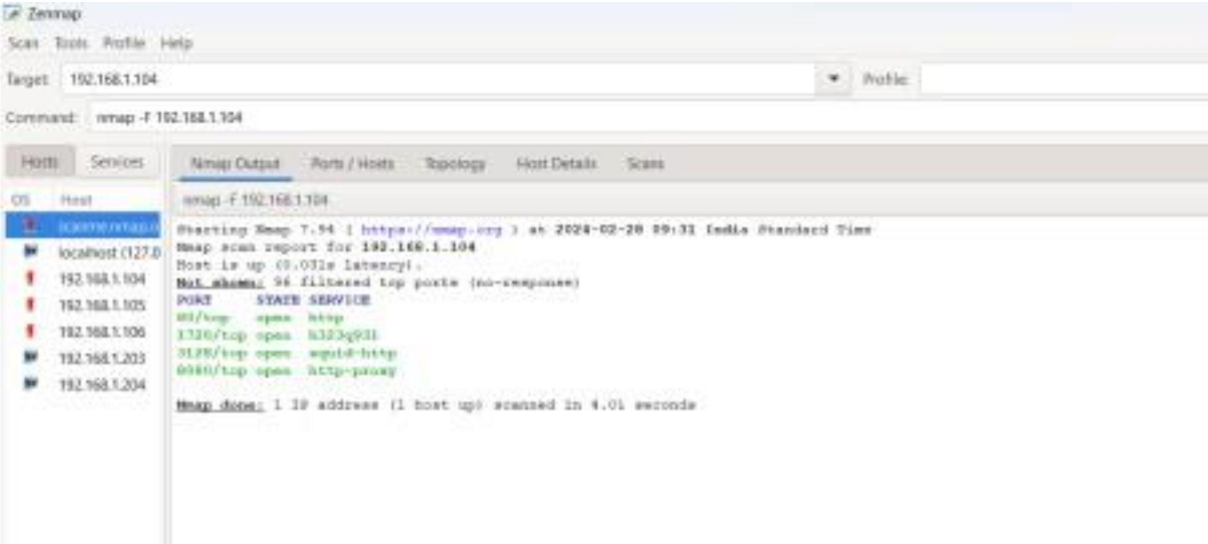
Target: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Command: nmap -6 2001:0db8:85a3:0000:0000:8a2e:0370:7334

```
nmap -6 2001:0db8:85a3:0000:0000:8a2e:0370:7334
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:29 India Standard Time
setup_target: failed to determine route to 2001:db8:85a3::8a2e:370:7334
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.41 seconds
```

9:*HOW TO PERFORM FAST SCAN :*

Syntax: nmap -F <target>



Target: 192.168.1.104

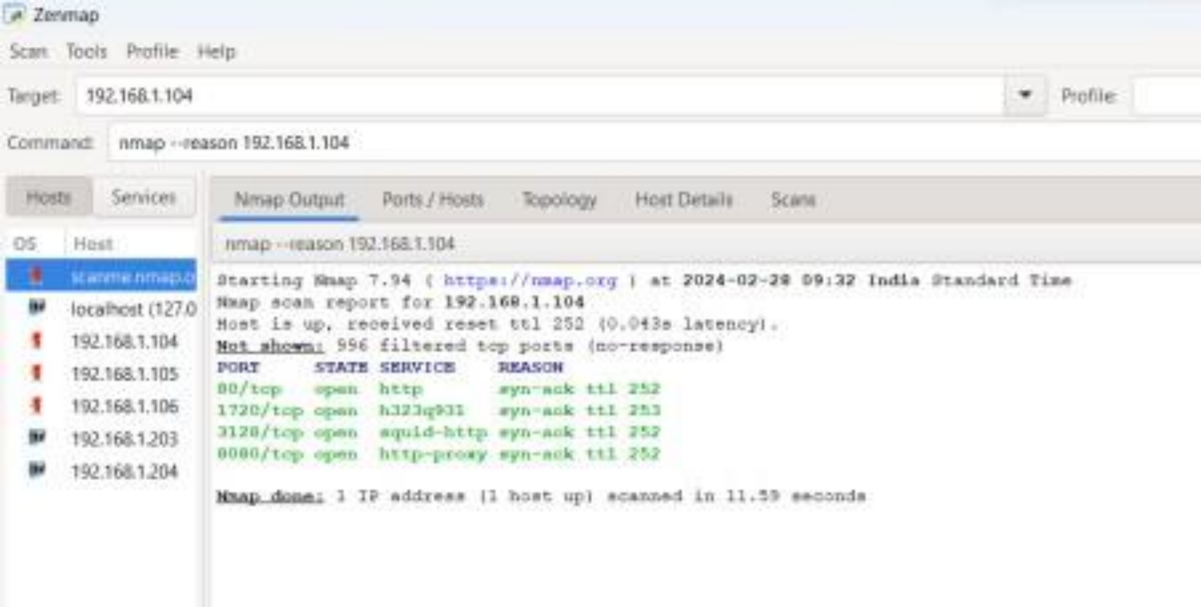
Command: nmap -F 192.168.1.104

```
nmap -F 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:31 India Standard Time
Nmap scan report for 192.168.1.104
Host is up (0.031s latency).
Not shown: 96 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
1720/tcp open  h323q931
3128/tcp open  squid-http
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 4.01 seconds
```

10:DISPLAY THE REASON THE PORT IS IN PARTICULAR STATE:

syntax: nmap --reason <target>



Target: 192.168.1.104

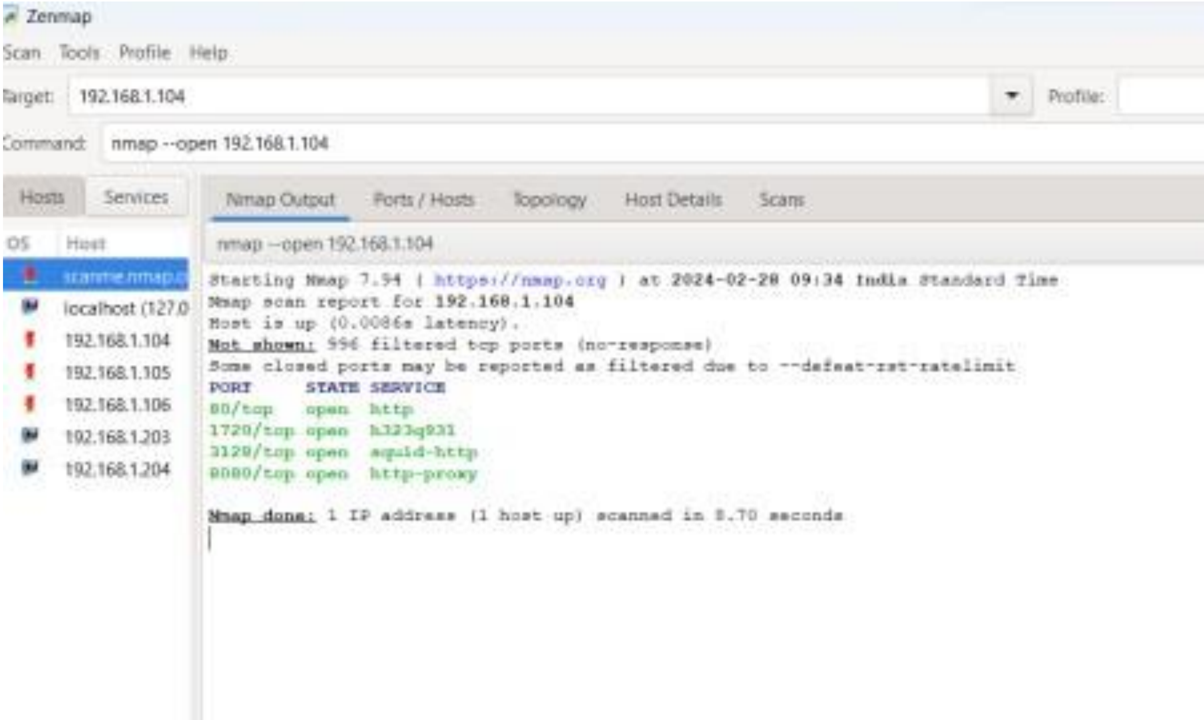
Command: nmap --reason 192.168.1.104

```
nmap --reason 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:32 India Standard Time
Nmap scan report for 192.168.1.104
Host is up, received reset ttl 252 (0.043s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE    REASON
80/tcp   open  http       syn-ack ttl 252
1720/tcp open  h323q931   syn-ack ttl 253
3128/tcp open  squid-http syn-ack ttl 252
8080/tcp open  http-proxy syn-ack ttl 252

Nmap done: 1 IP address (1 host up) scanned in 11.59 seconds
```

11.ONLY SHOW OPEN PORTS;

Syntax: nmap –open <target>



Target: 192.168.1.104

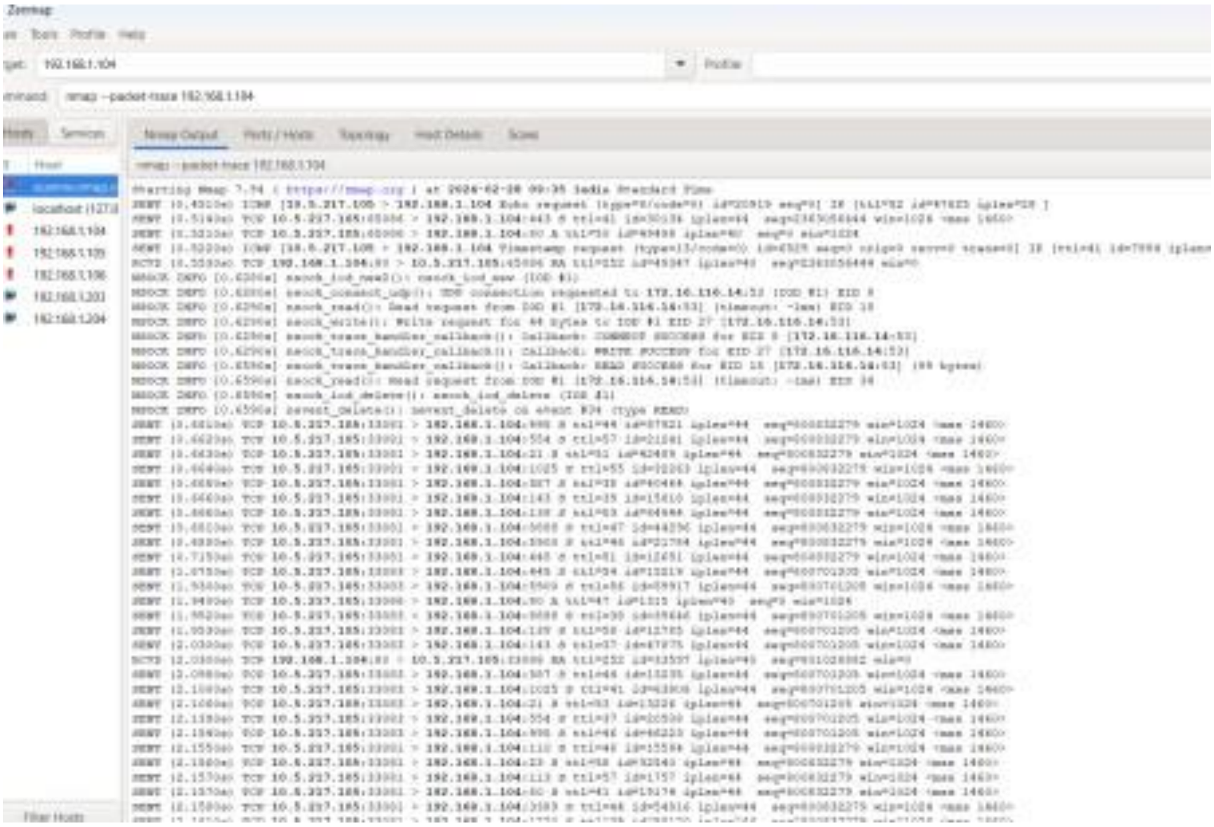
Command: nmap --open 192.168.1.104

```
nmap --open 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:34 India Standard Time
Nmap scan report for 192.168.1.104
Host is up (0.0086s latency).
Not shown: 996 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE
80/tcp   open  http
1720/tcp open  h323q931
3128/tcp open  squid-http
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 8.70 seconds
```

12: SHOW ALL PACKETS SEND AND RECEIVED:

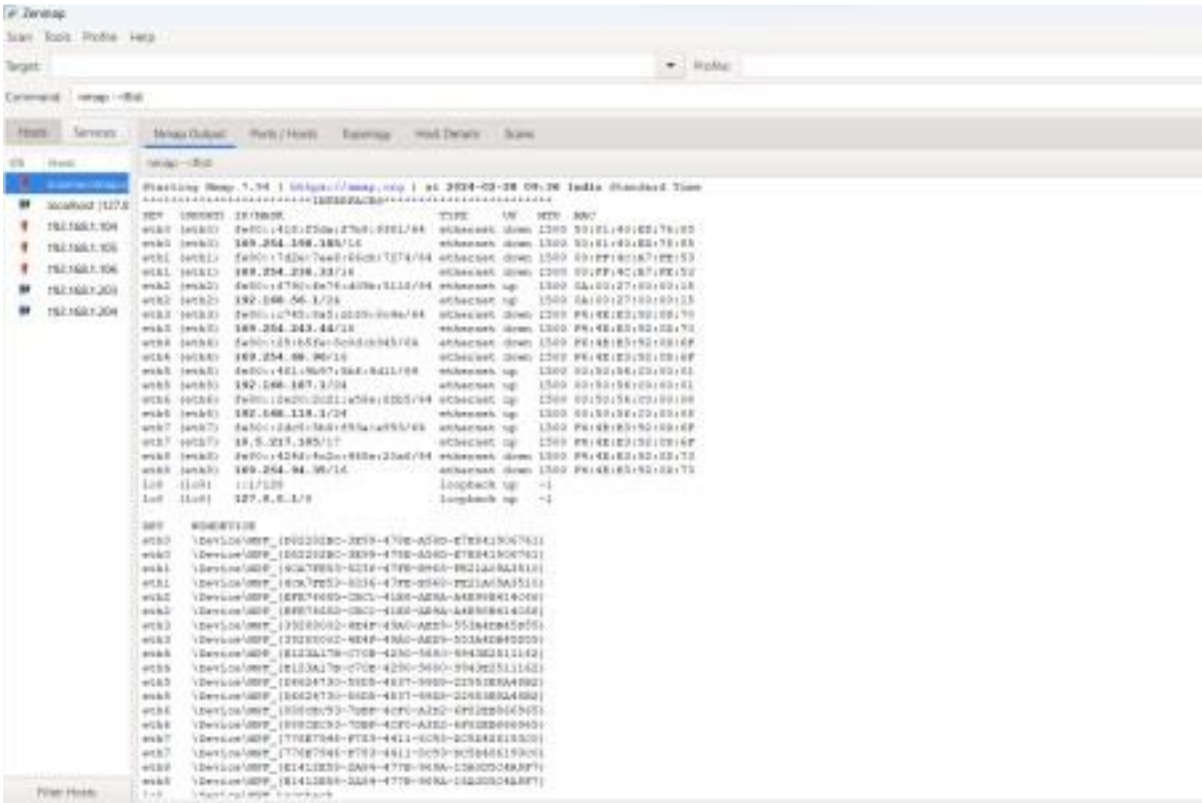
Syntax: nmap –packet-trace <target>



```
nmap --packet-trace 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:35 India Standard Time
SENT (0.4310s) ICMP [10.5.217.185 > 192.168.1.104 Echo request (type=8/code=0) id=[illegible] seq=0] IP [ttl=52 id=[illegible] iplen=28 ]
SENT (0.5140s) TCP 10.5.217.185:45006 > 192.168.1.104:443 S ttl=41 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.5210s) TCP 10.5.217.185:45006 > 192.168.1.104:80 A ttl=50 id=[illegible] iplen=40 seq=0 win=1024
SENT (0.5220s) ICMP [10.5.217.185 > 192.168.1.104 Timestamp request (type=13/code=0) id=[illegible] seq=0 orig=0 recv=0 trans=0] IP [ttl=41 id=[illegible] iplen=[illegible]
RCVD (0.5530s) TCP 192.168.1.104:80 > 10.5.217.185:45006 RA ttl=252 id=[illegible] iplen=40 seq=[illegible] win=0
NSOCK INFO [0.6200s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.6200s] nsock_connect_udp(): UDP connection requested to 172.16.116.14:53 (IOD #1) EID 8
NSOCK INFO [0.6290s] nsock_read(): Read request from IOD #1 [172.16.116.14:53] (timeout: -1ms) EID 18
NSOCK INFO [0.6290s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [172.16.116.14:53]
NSOCK INFO [0.6290s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.16.116.14:53]
NSOCK INFO [0.6290s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.16.116.14:53]
NSOCK INFO [0.6590s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.16.116.14:53] (44 bytes)
NSOCK INFO [0.6590s] nsock_read(): Read request from IOD #1 [172.16.116.14:53] (timeout: -1ms) EID 34
NSOCK INFO [0.6590s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.6590s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.6610s) TCP 10.5.217.185:33001 > 192.168.1.104:995 S ttl=44 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6620s) TCP 10.5.217.185:33001 > 192.168.1.104:554 S ttl=57 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6630s) TCP 10.5.217.185:33001 > 192.168.1.104:21 S ttl=51 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6640s) TCP 10.5.217.185:33001 > 192.168.1.104:1025 S ttl=55 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6650s) TCP 10.5.217.185:33001 > 192.168.1.104:587 S ttl=38 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6660s) TCP 10.5.217.185:33001 > 192.168.1.104:143 S ttl=35 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6660s) TCP 10.5.217.185:33001 > 192.168.1.104:139 S ttl=53 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6810s) TCP 10.5.217.185:33001 > 192.168.1.104:5668 S ttl=47 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.6830s) TCP 10.5.217.185:33001 > 192.168.1.104:3968 S ttl=46 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (0.7150s) TCP 10.5.217.185:33001 > 192.168.1.104:445 S ttl=51 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (1.0750s) TCP 10.5.217.185:33003 > 192.168.1.104:445 S ttl=34 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (1.5340s) TCP 10.5.217.185:33003 > 192.168.1.104:5509 S ttl=56 id=[illegible] iplen=44 seq=[illegible] win=1024 <mss 1460>
SENT (1.9490s) TCP 10.5.217.185:33006 > 192.168.1.104:80 A ttl=47 id=[illegible] iplen=40 seq=0 win=1024
[illegible]
```

13.SHOW HOST INTERFACE AND ROUTES:

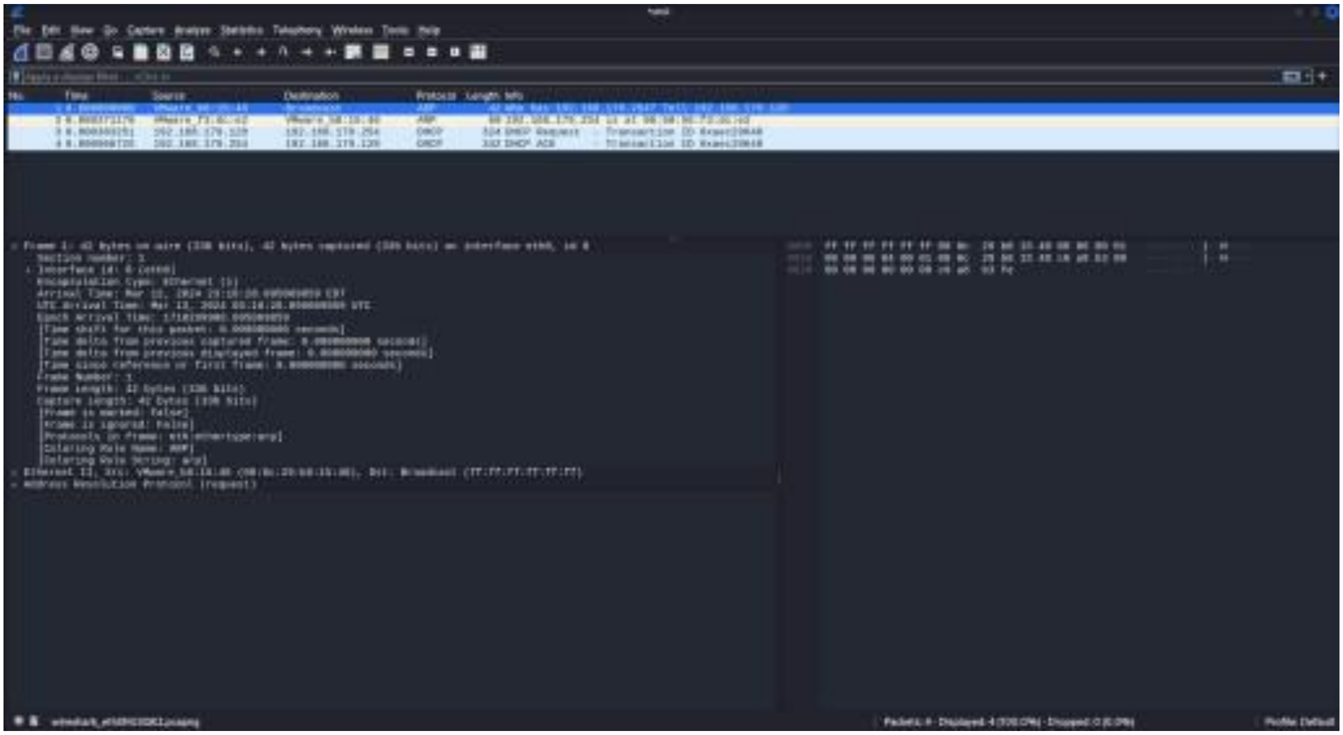
Syntax: nmap –iflist



```
nmap --iflist
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-28 09:36 India Standard Time
************************INTERFACES************************
DEV  (SHORT) IP/MASK                            TYPE     UP   MTU  MAC
eth0 (eth0)  fe80::410:f5de:2768:8881/64         ethernet down 1500 50:81:40:E2:76:85
eth0 (eth0)  169.254.108.185/16                 ethernet down 1500 50:81:40:E2:76:85
eth1 (eth1)  fe80::7d2e:7ee8:66cb:7274/64        ethernet down 1500 00:FF:4C:87:FE:53
eth1 (eth1)  169.254.238.33/16                  ethernet down 1500 00:FF:4C:87:FE:53
eth2 (eth2)  fe80::4790:8e76:d08e:5118/64        ethernet up   1500 0A:00:27:00:00:18
eth2 (eth2)  192.168.56.1/24                    ethernet up   1500 0A:00:27:00:00:18
eth3 (eth3)  fe80::c745:3a5:d220:0c0e/64         ethernet down 1500 F6:4E:E3:92:08:70
eth3 (eth3)  169.254.243.44/16                  ethernet down 1500 F6:4E:E3:92:08:70
eth4 (eth4)  fe80::25:b5fa:5cbd:cb45/64          ethernet down 1500 F6:4E:E3:92:08:6F
eth4 (eth4)  169.254.88.96/16                   ethernet down 1500 F6:4E:E3:92:08:6F
eth5 (eth5)  fe80::481:9b97:5bd:8d11/64          ethernet up   1500 00:50:56:C0:00:01
eth5 (eth5)  192.168.107.1/24                   ethernet up   1500 00:50:56:C0:00:01
eth6 (eth6)  fe80::2e20:2021:a58e:6305/64        ethernet up   1500 00:50:56:C0:00:08
eth6 (eth6)  192.168.118.1/24                   ethernet up   1500 00:50:56:C0:00:08
eth7 (eth7)  fe80::2de5:5b8:f99a:a993/64         ethernet up   1500 F6:4E:E3:92:08:6F
eth7 (eth7)  10.5.217.185/17                    ethernet up   1500 F6:4E:E3:92:08:6F
eth8 (eth8)  fe80::424d:4e2c:988e:23ad/64        ethernet down 1500 F6:4E:E3:92:08:72
eth8 (eth8)  169.254.94.38/16                   ethernet down 1500 F6:4E:E3:92:08:72
lo0  (lo0)   ::1/128                            loopback up   -1
lo0  (lo0)   127.0.0.1/8                        loopback up   -1

DEV    WINDEVICE
eth0   \Device\NPF_{D6220[illegible]}
eth0   \Device\NPF_{D6220[illegible]}
eth1   \Device\NPF_{6CA7FE53-[illegible]}
eth1   \Device\NPF_{6CA7FE53-[illegible]}
eth2   \Device\NPF_{EFB7[illegible]-CBC1-41[illegible]}
eth2   \Device\NPF_{EFB7[illegible]-CBC1-41[illegible]}
eth3   \Device\NPF_{[illegible]}
eth3   \Device\NPF_{[illegible]}
eth4   \Device\NPF_{[illegible]}
eth4   \Device\NPF_{[illegible]}
eth5   \Device\NPF_{[illegible]}
eth5   \Device\NPF_{[illegible]}
eth6   \Device\NPF_{[illegible]}
eth6   \Device\NPF_{[illegible]}
eth7   \Device\NPF_{[illegible]}
eth7   \Device\NPF_{[illegible]}
eth8   \Device\NPF_{[illegible]}
eth8   \Device\NPF_{[illegible]}
lo0    \Device\NPF_Loopback
```

Frames



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

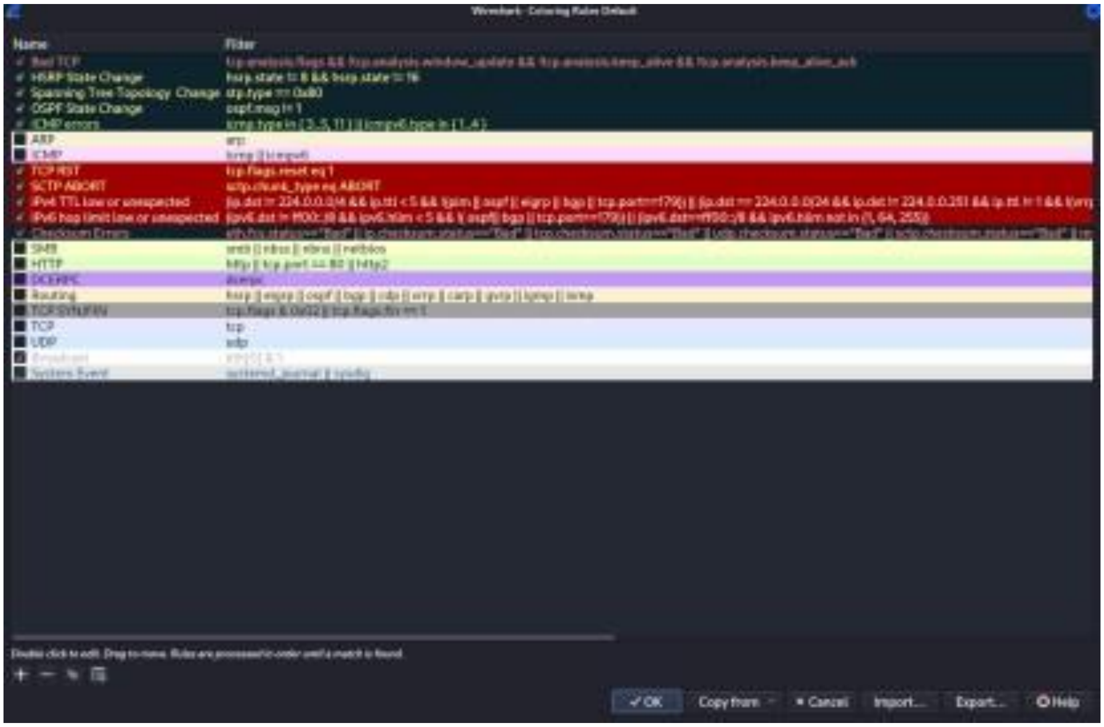
```
No.  Time          Source             Destination        Protocol Length Info
   1 0.000000000   VMware_[illegible] Broadcast          ARP      42 Who has 192.168.179.254? Tell 192.168.179.128
   2 0.000371179   VMware_f3:[illegible] VMware_58:18:8d ARP      60 192.168.179.254 is at 00:50:56:f3:[illegible]
   3 0.000[illegible] 192.168.179.128 192.168.179.254    DHCP     324 DHCP Request  - Transaction ID 0x[illegible]
   4 0.000[illegible] 192.168.179.254 192.168.179.128    DHCP     332 DHCP ACK      - Transaction ID 0x[illegible]
```

```
› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
    Section number: 1
  › Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 11, 2024 [illegible] CDT
    UTC Arrival Time: Mar 12, 2024 [illegible] UTC
    Epoch Arrival Time: 1710[illegible]
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
› Ethernet II, Src: VMware_58:18:8d (00:0c:29:58:18:8d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff [illegible]
0010  [illegible]
0020  [illegible]
```

wireshark_eth0[illegible].pcapng
Packets: 4 · Displayed: 4 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

Coloring Rules

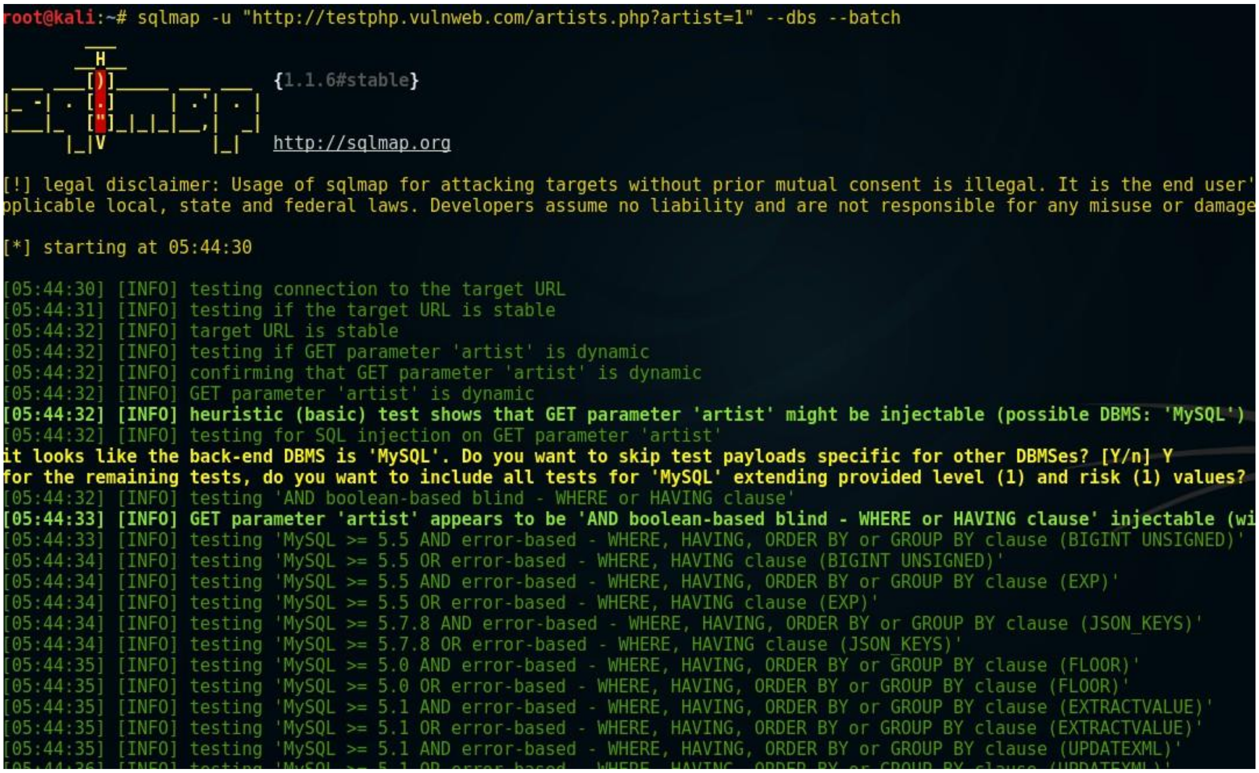


Name	Filter
✓ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update && !tcp.analysis.keep_alive && !tcp.analysis.keep_alive_ack
✓ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
✓ Spanning Tree Topology Change	stp.type == 0x80
✓ OSPF State Change	ospf.msg != 1
✓ ICMP errors	icmp.type in { 3..5, 11 } || icmpv6.type in { 1..4 }
■ ARP	arp
■ ICMP	icmp || icmpv6
✓ TCP RST	tcp.flags.reset eq 1
✓ SCTP ABORT	sctp.chunk_type eq ABORT
✓ IPv4 TTL low or unexpected	(ip.dst != 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf && !eigrp && !bgp && !tcp.port==179) || (ip.dst == 224.0.0.0/24 && ip.dst != 224.0.0.251 && ip.ttl != 1 && !(vrrp
✓ IPv6 hop limit low or unexpected	(ipv6.dst != ff00::/8 && ipv6.hlim < 5 && !(ospf|| bgp || tcp.port==179)) || (ipv6.dst==ff00::/8 && ipv6.hlim not in {1, 64, 255})
✓ Checksum Errors	eth.fcs.status=="Bad" || ip.checksum.status=="Bad" || tcp.checksum.status=="Bad" || udp.checksum.status=="Bad" || sctp.checksum.status=="Bad" || m
■ SMB	smb || nbss || nbns || netbios
■ HTTP	http || tcp.port == 80 || http2
■ DCERPC	dcerpc
■ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvrp || igmp || ismp
■ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
■ TCP	tcp
■ UDP	udp
■ Broadcast	eth[0] & 1
■ System Event	systemd_journal || syslog

Double click to edit. Drag to move. Rules are processed in order until a match is found.

SQL Map

SQLMap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications.



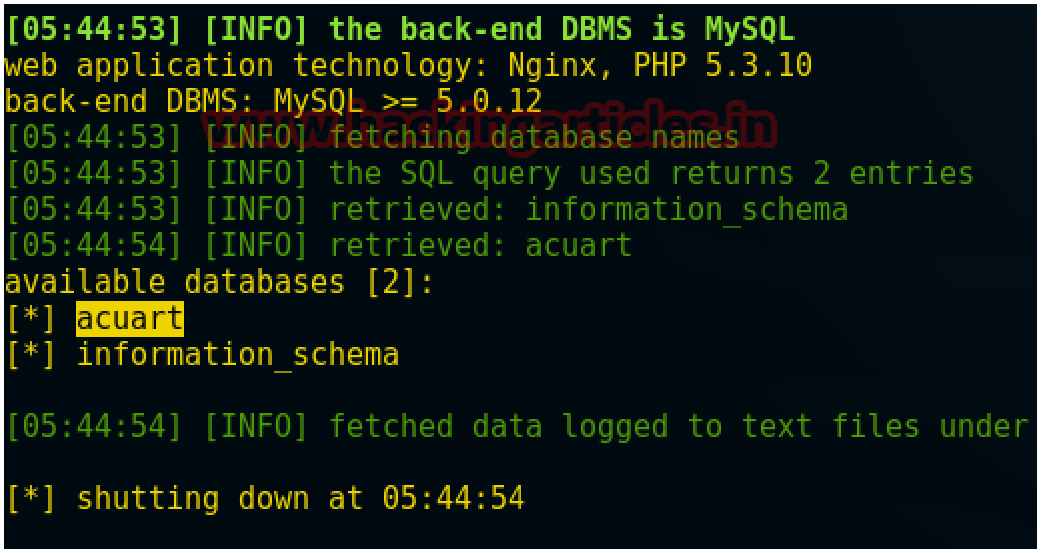
```
root@kali:~# sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs --batch
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.1.6#stable}
|_ -| . [.]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user'
pplicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage

[*] starting at 05:44:30

[05:44:30] [INFO] testing connection to the target URL
[05:44:31] [INFO] testing if the target URL is stable
[05:44:32] [INFO] target URL is stable
[05:44:32] [INFO] testing if GET parameter 'artist' is dynamic
[05:44:32] [INFO] confirming that GET parameter 'artist' is dynamic
[05:44:32] [INFO] GET parameter 'artist' is dynamic
[05:44:32] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[05:44:32] [INFO] testing for SQL injection on GET parameter 'artist'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values?
[05:44:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:44:33] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (wi
[05:44:33] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[05:44:34] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE, HAVING clause (BIGINT UNSIGNED)'
[05:44:34] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[05:44:34] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE, HAVING clause (EXP)'
[05:44:34] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[05:44:34] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE, HAVING clause (JSON_KEYS)'
[05:44:35] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:44:35] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:44:35] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[05:44:35] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[05:44:35] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[05:44:36] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
```

It is designed to help security professionals and ethical hackers identify and assess the security posture of web applications by probing for SQL injection flaws.



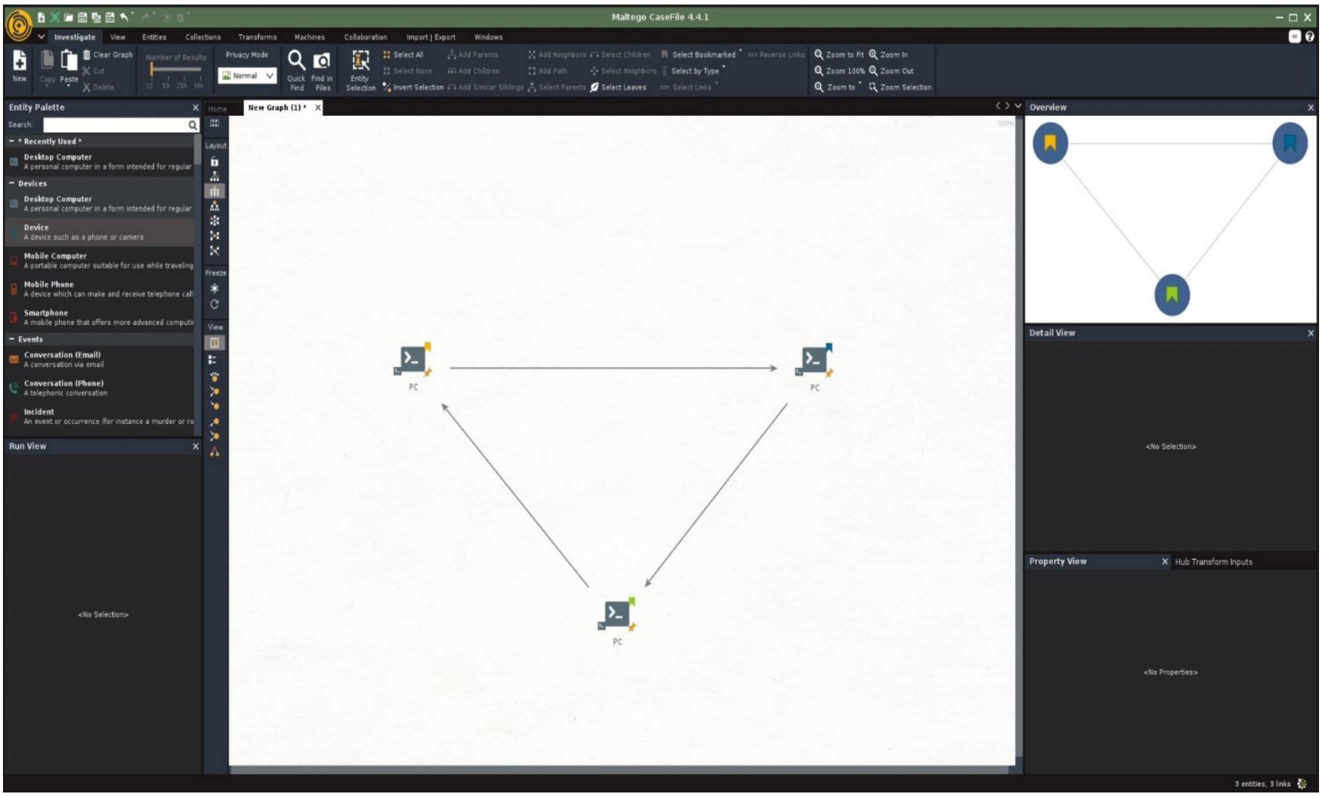
```
[05:44:53] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0.12
[05:44:53] [INFO] fetching database names
[05:44:53] [INFO] the SQL query used returns 2 entries
[05:44:53] [INFO] retrieved: information_schema
[05:44:54] [INFO] retrieved: acuart
available databases [2]:
[*] acuart
[*] information_schema

[05:44:54] [INFO] fetched data logged to text files under

[*] shutting down at 05:44:54
```

MALTEGO

Maltego is a data visualization and analysis tool used for link analysis, data mining, and information gathering. It allows users to gather and correlate information from various sources, such as public databases, social networks, websites, and other online platforms, to create a graphical representation of relationships and connections between entities. These entities could be people, organizations, websites, email addresses, IP addresses, and more.



1. **System Requirements:** Ensure that your system meets the minimum requirements for running Nmap. It is compatible with various operating systems including Linux, Windows, and macOS.
2. **Install Nmap:** If you're using a Debian-based Linux distribution like Ubuntu or Kali Linux, you can install Nmap from the official repositories using the package manager. Open a terminal window and run the following command:

```
sudo apt update
sudo apt install nmap
```

If you're using a different Linux distribution, you can typically find Nmap in the official repositories or compile it from source.

For Windows and macOS users, you can download the installer from the official Nmap website (https://nmap.org/download.html) and follow the installation instructions provided.

3. **Verify Installation:** After the installation is complete, you can verify that Nmap is installed correctly by running the following command in the terminal:

```
nmap --version
```

This command should display the installed version of Nmap along with other information.

4. **Usage:** Once Nmap is installed, you can start using it to perform network discovery, host scanning, port scanning, service enumeration, and various other network reconnaissance tasks. Refer to the Nmap documentation (https://nmap.org/book/man.html) or run man nmap in the terminal for detailed usage instructions and examples.
5. **Update Nmap (Optional):** It's a good practice to regularly update Nmap to ensure you have the latest features, bug fixes, and security patches. You can update Nmap using the package manager on Linux or by downloading the latest version from the official website.

By following these steps, you'll have Nmap installed on your system and ready to use for network scanning and reconnaissance purposes.

STEPS OF ETHICAL HACKING THAT COULD BE PERFORMED ON PUBLIC DOMAIN:

Ethical hacking, follows a structured approach to ensure responsible and effective testing while minimizing potential harm.

1. Information Gathering:

Gather information about the target organization, its employees, infrastructure, and security measures. Utilize open-source intelligence (OSINT) techniques to collect publicly available information from sources such as social media, company websites, online forums, and public records.

2. Reconnaissance:

Conduct reconnaissance to identify potential attack vectors and vulnerabilities within the target organization. Analyze the gathered information to determine the most effective tactics and targets.

3.Scanning and enumeration:

Create a plausible pretext or scenario to establish trust and credibility with the target individual or organization. Craft a convincing story or persona that aligns with the pretext, such as posing as an IT support technician, vendor representative, or trusted colleague.

4.Gaining access/Exploitation:

Gaining access or exploitation is approached with the utmost caution and adherence to legal and ethical boundaries. Identifying vulnerabilities , Exploitation frameworks , Payload Deployment are done in the step gaining access / exploitation . Exploit human trust and vulnerabilities to elicit the desired response from the target individuals, such as clicking on malicious links, downloading attachments, or disclosing sensitive information.

5.Maintaining Access and Persistence:

After successful exploitation, maintain persistence within the target environment to gather additional information, escalate privileges, or conduct further attacks. Document the results of the attack, including any compromised credentials, sensitive data obtained, and lessons learned for future engagements.

6.Clearing Tracks:

In ethical hacking, clearing tracks after conducting an attack is crucial to maintain confidentiality, integrity, and legal compliance. Logging and Documentation , Reversibility , Data Sanitization , Covering Tracks , Restoration of original state are some of the clearing tracks done in social engineering attack.

CONCLUSION:

In conclusion, the security assessment conducted within the public domain has unearthed critical vulnerabilities and highlighted significant implications for the organization's overall security posture. The successful identification and exploitation of weaknesses emphasize the pressing need for strengthened defenses and heightened awareness among stakeholders.

Insights gleaned from the assessment shed light on human behaviors and organizational susceptibilities, underlining the importance of addressing these aspects alongside technical vulnerabilities. Recommendations stemming from the assessment encompass a multifaceted approach, including:

Enhanced Security Awareness: Implementing comprehensive security awareness training programs to educate employees about social engineering tactics and the importance of vigilance in safeguarding sensitive information.

Policy Enforcement: Strengthening policy enforcement mechanisms to ensure adherence to security protocols, including strict authentication procedures, access control policies, and incident response protocols.

Technical Controls: Implementing robust technical controls such as intrusion detection/prevention systems, endpoint security solutions, and encryption mechanisms to fortify the organization's defense against social engineering attacks.

Layered Defenses: Adopting a layered defense strategy that combines technical controls, security awareness training, and proactive monitoring to create multiple barriers against potential threats.

Continuous Improvement: Emphasizing the importance of continuous improvement through regular security assessments, penetration testing, and incident response drills to adapt to evolving threats and vulnerabilities.

REFERENCES:

Official Documentation and Guides:

OWASP (Open Web Application Security Project): Provides comprehensive documentation, guides, and tools for web application security testing and assessment.

Website: https://owasp.org/

NIST (National Institute of Standards and Technology) Special Publications: Offers a wide range of cybersecurity publications, guidelines, and frameworks for security assessments and risk management.

Website: https://www.nist.gov/publications

Books:

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto: Offers in-depth insights into web application security testing methodologies and techniques.

"Network Security Assessment: Know Your Network" by Chris McNab: Provides a comprehensive guide to network security assessments, covering tools, methodologies, and best practices.

"Hacking: The Art of Exploitation" by Jon Erickson: Offers hands-on examples and exercises for understanding security vulnerabilities and conducting security assessments.

Online Resources:

SANS Institute: Offers a variety of cybersecurity training courses, webinars, and resources covering security assessment techniques and methodologies.

Website: https://www.sans.org/

SecurityFocus: Provides security advisories, vulnerability databases, and

discussion forums for security professionals involved in security assessments and penetration testing.

Website: https://www.securityfocus.com/

Government Publications and Guidelines:

US-CERT (United States Computer Emergency Readiness Team): Offers cybersecurity alerts, tips, and publications for enhancing cybersecurity posture and conducting security assessments.

Website: https://www.us-cert.gov/

CISA (Cybersecurity and Infrastructure Security Agency): Provides resources, tools, and guidelines for securing critical infrastructure and conducting security assessments.

Website: https://www.cisa.gov/