Title: Network Scanning using Nmap and Wireshark with NSE

SURESH MOUDH

LinkedIn: https://www.linkedin.com/in/suresh-moudh/

Supervisor: DIMPLE CHAUHAN

Warning

This project is just about Learning

Agenda

- Introduction about Project
- Introduction about Tools:
  - Introduction about Wireshark
  - Introduction about Nmap and NSE
- Project requirements and Testing website/network
- Perform Wireshark IDS/Firewall evade techniques
- Perform NSE scanning for testing network/website
- Conclusion
- Reference
- Thank You

Introduction

Understanding network behavior and vulnerabilities is essential in the quickly changing field of cybersecurity to guarantee the stability and security of digital environments. For network managers, cybersecurity experts, and IT supporters, network scanning and analysis tools like Nmap and Wireshark are essential.

The goal of this project is to use Nmap, a powerful network scanning tool, to find open ports, active devices, and possible network vulnerabilities. In parallel, network traffic is captured, inspected, and analyzed in real-time using Wireshark, an advanced packet analysis tool. When combined, these resources offer a thorough understanding of how network communications and security systems operate.

This project is to improve knowledge of network reconnaissance, traffic analysis, and anomaly detection by investigating the features, approaches, and real-world uses of Nmap and Wireshark. It provides a practical method for strengthening cybersecurity defenses and developing expertise in network diagnostics.

Introduction about Tools

1.Introduction about Nmap

- Network Mapper, sometimes known as Nmap, is a powerful and popular open-source program for network exploration and security analysis. It was developed by Gordon Lyon and has grown to be an essential tool for network managers and cybersecurity experts. By using Nmap, users can find vulnerabilities that an attacker could exploit, discover connected devices, scan networks, and find open ports. Proactive security measures require Nmap since it offers useful information about operating systems, running services, and network topology by sending custom packets and examining the responses.

- The program is very flexible, providing a range of scanning methods that may be customized for particular use cases, including OS detection, UDP scans, and TCP SYN scans. The Nmap Scripting Engine (NSE), which makes automation and sophisticated vulnerability assessments possible, extensively expands its functionality. Nmap has become vital for analyzing, controlling, and defending recent network settings because of its capacity to efficiently supervise networks of various sizes.

2. Introduction to Wireshark

- Wireshark is a well-known open-source program used in cybersecurity to analyze network traffic and identify potential threats. It allows specialists to record and evaluate data packets in real time, providing a thorough perspective of network activities. With support for decoding hundreds of protocols, Wireshark delivers extensive insights into communication processes, making it a crucial tool for discovering vulnerabilities and analyzing security occurrences.

- Wireshark is an essential tool in the field of cybersecurity for monitoring illegal access, analyzing malware activity, and identifying security breaches. Its sophisticated filtering capabilities help users focus on particular traffic patterns, like suspect IP addresses or unusual protocol usage, which are essential for spotting and resolving security threats. Wireshark is also incredibly useful for forensic investigations, intrusion detection, and strengthening network defenses because to its capacity to reconstruct data streams and examine packet-level details.

Project requirements and Testing website/network

Tools:
1. Nmap
2. Wireshark
3. A testing website(ctf365.com)
4. Windows System
5. Kali software

Test website- Select a website such as the Damn Vulnerable Web Application such as ctf365.com that is intended for security testing.

Virtual machine- Setup a Kali Linux and Windows machine using a Virtualbox/VM.

Windows Server ip - 192.168.23.32(Target Machine)

Kali Linux - 10.12.10.2 (Test Machine)

Test Website - ctf365.com, pentesting.com

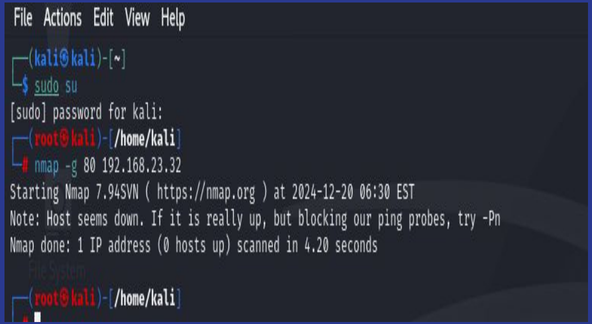
Perform Wireshark IDS/Firewall evade techniques

Using tools like Wireshark to avoid detection by a firewall or Intrusion Detection System (IDS) is a delicate area of cybersecurity. Although understanding these ideas is helpful in identifying system vulnerabilities, testing and applying these strategies should always adhere to legal and ethical requirements.

1. Source Port Manipulation :

Source port manipulation is a method of changing the source port number in packet headers to get around firewalls, intrusion detection systems, and other network monit
ring tools. Rules that use particular port configurations to filter traffic can be circumvented using this method. This strategy, its ramifications, and countermeasures are broken down below.

Nmap uses -g or –source-port options to perform source port manipulation.S
Syntax: nmap -g target ip

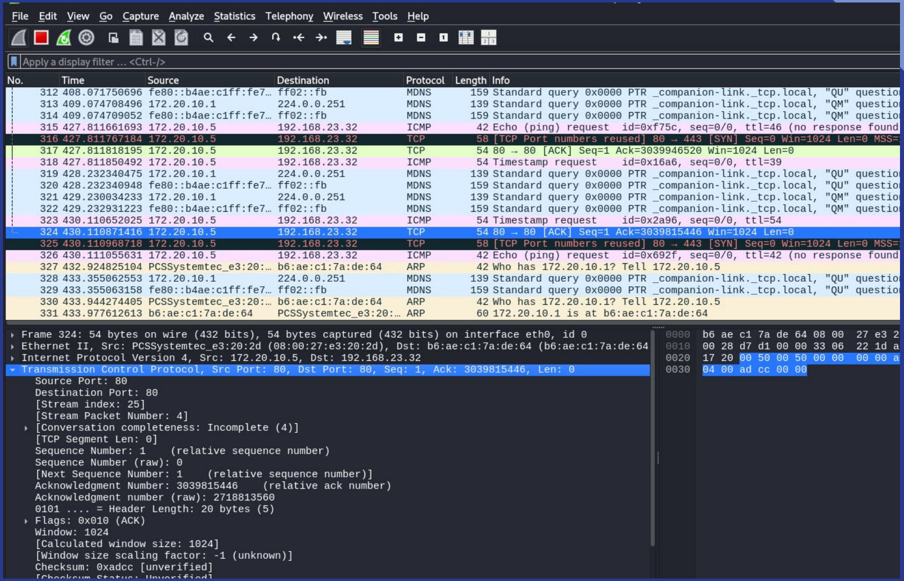


```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# nmap -g 80 192.168.23.32
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 06:30 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 4.20 seconds

┌──(root㉿kali)-[/home/kali]
└─#
```

Fig: A

In Kali linux terminal, after the execution of the command for source port manipulation, the result in Fig A, the command send the packets from port 80 of the ip address.

Simultaneously the result of wireshark in Fig B represents the details of the ip address packets with the detail of the source port 80 details info of port, port segment, sequence number, Acknowledgement, flags, window size is 1024.


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

```
No.   Time          Source                 Destination            Protocol Length Info
  312 408.071750696 fe80::b4ae:c1ff:fe7…   ff02::fb               MDNS      159 Standard query 0x0000 PTR _companion-link._tcp.local, "QU" questio
  313 409.074708496 172.20.10.1            224.0.0.251            MDNS      139 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" questio
  314 409.074709052 fe80::b4ae:c1ff:fe7…   ff02::fb               MDNS      159 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" questio
  315 427.811661693 172.20.10.5            192.168.23.32          ICMP       42 Echo (ping) request  id=0xf75c, seq=0/0, ttl=46 (no response found
  316 427.811767184 172.20.10.5            192.168.23.32          TCP        58 [TCP Port numbers reused] 80 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=
  317 427.811818195 172.20.10.5            192.168.23.32          TCP        54 80 → 80 [ACK] Seq=1 Ack=3039946520 Win=1024 Len=0
  318 427.811850492 172.20.10.5            192.168.23.32          ICMP       54 Timestamp request    id=0x16a6, seq=0/0, ttl=39
  319 428.232340475 172.20.10.1            224.0.0.251            MDNS      139 Standard query 0x0000 PTR _companion-link._tcp.local, "QU" questio
  320 428.232340948 fe80::b4ae:c1ff:fe7…   ff02::fb               MDNS      159 Standard query 0x0000 PTR _companion-link._tcp.local, "QU" questio
  321 429.230034233 172.20.10.1            224.0.0.251            MDNS      139 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" questio
  322 429.232931223 fe80::b4ae:c1ff:fe7…   ff02::fb               MDNS      159 Standard query 0x0000 PTR _companion-link._tcp.local, "QM" questio
  323 430.110652025 172.20.10.5            192.168.23.32          ICMP       54 Timestamp request    id=0x2a96, seq=0/0, ttl=54
  324 430.110871416 172.20.10.5            192.168.23.32          TCP        54 80 → 80 [ACK] Seq=1 Ack=3039815446 Win=1024 Len=0
  325 430.110968718 172.20.10.5            192.168.23.32          TCP        58 [TCP Port numbers reused] 80 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=
  326 430.111055631 172.20.10.5            192.168.23.32          ICMP       42 Echo (ping) request  id=0x692f, seq=0/0, ttl=42 (no response found
  327 432.924825104 PCSSystemtec_e3:20:…   b6:ae:c1:7a:de:64      ARP        42 Who has 172.20.10.1? Tell 172.20.10.5
  328 433.355062553 172.20.10.1            224.0.0.251            MDNS      139 Standard query 0x0000 PTR _companion-link._tcp.local, "QU" questio
  329 433.355063158 fe80::b4ae:c1ff:fe7…   ff02::fb               MDNS      159 Standard query 0x0000 PTR _companion-link._tcp.local, "QU" questio
  330 433.944274405 PCSSystemtec_e3:20:…   b6:ae:c1:7a:de:64      ARP        42 Who has 172.20.10.1? Tell 172.20.10.5
  331 433.977612613 b6:ae:c1:7a:de:64      PCSSystemtec_e3:20:…   ARP        60 172.20.10.1 is at b6:ae:c1:7a:de:64
```

```
› Frame 324: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
› Ethernet II, Src: PCSSystemtec_e3:20:2d (08:00:27:e3:20:2d), Dst: b6:ae:c1:7a:de:64 (b6:ae:c1:7a:de:64
› Internet Protocol Version 4, Src: 172.20.10.5, Dst: 192.168.23.32
⌄ Transmission Control Protocol, Src Port: 80, Dst Port: 80, Seq: 1, Ack: 3039815446, Len: 0
    Source Port: 80
    Destination Port: 80
    [Stream index: 25]
    [Stream Packet Number: 4]
  › [Conversation completeness: Incomplete (4)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 0
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 3039815446    (relative ack number)
    Acknowledgment number (raw): 2718813560
    0101 .... = Header Length: 20 bytes (5)
  › Flags: 0x010 (ACK)
    Window: 1024
    [Calculated window size: 1024]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xadcc [unverified]
    [Checksum Status: Unverified]
```

```
0000  b6 ae c1 7a de 64 08 00  27 e3 2
0010  00 28 d7 d1 00 00 33 06  22 1d a
0020  17 20 00 50 00 50 00 00  00 00 a
0030  04 00 ad cc 00 00
```

Fig: B

2.Packet Fragmentation:

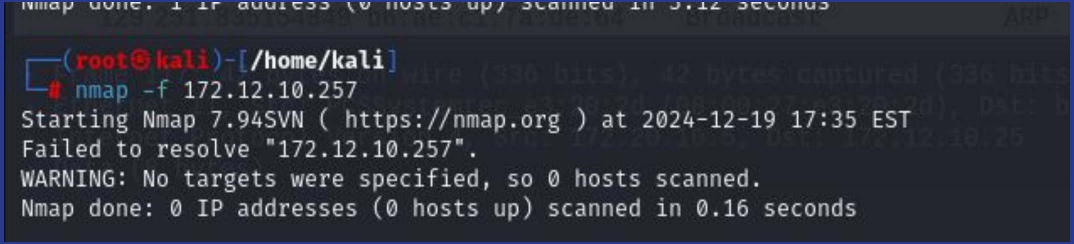
In network scanning, packet fragmentation is a technique that divides packets into smaller pieces prior to transmission. In order to get beyond firewalls, intrusion detection systems (IDS), and other security measures that examine traffic, this technique is frequently used.

There are several uses for packet fragmentation when utilizing programs like Nmap and Wireshark:

- Avoid detection by security systems
- Test firewall rules
- Evade MTU restrictions

The TCP header is separated into various packets so packet filters are not able to detect what packets are intended to do.

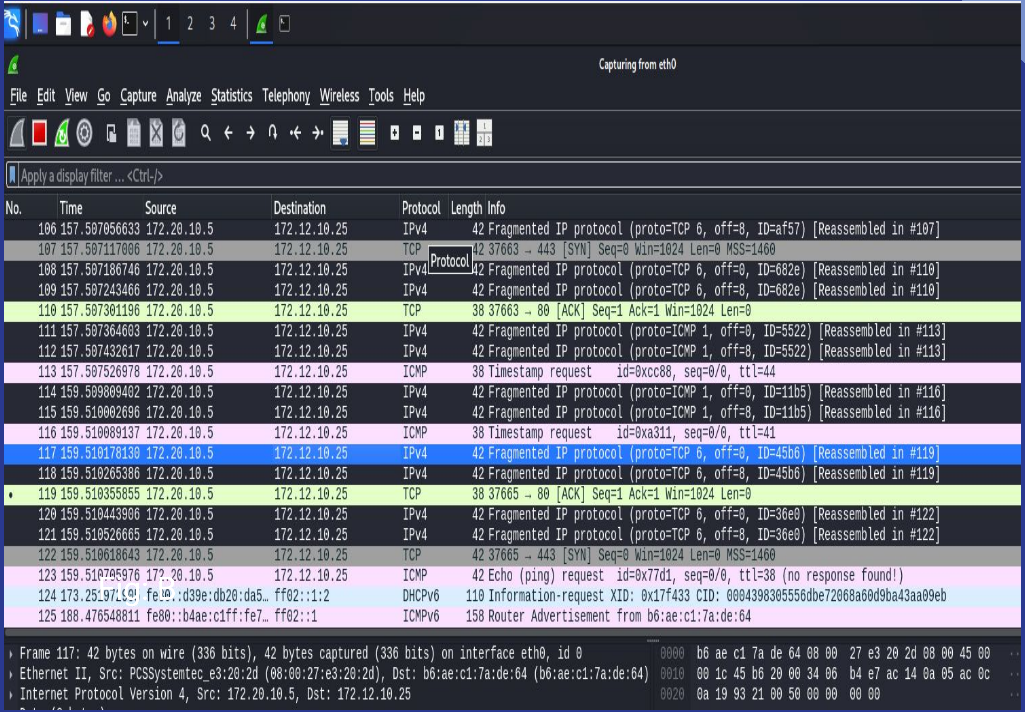
Syntax: nmap -f targeted windows ip



```
Nmap done: 1 IP address (0 hosts up) scanned in 3.12 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -f 172.12.10.257
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 17:35 EST
Failed to resolve "172.12.10.257".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.16 seconds
```

Fig: A

The result of the command shows the list of packets with fragmentation in the output of wireshark in Fig: B



```
No.   Time           Source              Destination    Protocol Length Info
  106 157.507056633  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=8, ID=af57) [Reassembled in #107]
  107 157.507117006  172.20.10.5         172.12.10.25   TCP      42 37663 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  108 157.507186746  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=0, ID=682e) [Reassembled in #110]
  109 157.507243466  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=8, ID=682e) [Reassembled in #110]
  110 157.507301196  172.20.10.5         172.12.10.25   TCP      38 37663 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
  111 157.507364603  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=ICMP 1, off=0, ID=5522) [Reassembled in #113]
  112 157.507432617  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=ICMP 1, off=8, ID=5522) [Reassembled in #113]
  113 157.507526978  172.20.10.5         172.12.10.25   ICMP     38 Timestamp request    id=0xcc88, seq=0/0, ttl=44
  114 159.509809402  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=ICMP 1, off=0, ID=11b5) [Reassembled in #116]
  115 159.510002696  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=ICMP 1, off=8, ID=11b5) [Reassembled in #116]
  116 159.510089137  172.20.10.5         172.12.10.25   ICMP     38 Timestamp request    id=0xa311, seq=0/0, ttl=41
  117 159.510178130  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=0, ID=45b6) [Reassembled in #119]
  118 159.510265386  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=8, ID=45b6) [Reassembled in #119]
• 119 159.510355855  172.20.10.5         172.12.10.25   TCP      38 37665 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
  120 159.510443906  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=0, ID=36e0) [Reassembled in #122]
  121 159.510526665  172.20.10.5         172.12.10.25   IPv4     42 Fragmented IP protocol (proto=TCP 6, off=8, ID=36e0) [Reassembled in #122]
  122 159.510618643  172.20.10.5         172.12.10.25   TCP      42 37665 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  123 159.510705976  172.20.10.5         172.12.10.25   ICMP     42 Echo (ping) request  id=0x77d1, seq=0/0, ttl=38 (no response found!)
  124 173.25197[illegible] fe80::d39e:db20:da5… ff02::1:2 DHCPv6  110 Information-request XID: 0x17f433 CID: 0004398305556dbe72068a60d9ba43aa09eb
  125 188.476548811  fe80::b4ae:c1ff:fe7… ff02::1       ICMPv6  158 Router Advertisement from b6:ae:c1:7a:de:64

› Frame 117: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
› Ethernet II, Src: PCSSystemtec_e3:20:2d (08:00:27:e3:20:2d), Dst: b6:ae:c1:7a:de:64 (b6:ae:c1:7a:de:64)
› Internet Protocol Version 4, Src: 172.20.10.5, Dst: 172.12.10.25

0000  b6 ae c1 7a de 64 08 00  27 e3 20 2d 08 00 45 00
0010  00 1c 45 b6 20 00 34 06  b4 e7 ac 14 0a 05 ac 0c
0020  0a 19 93 21 00 50 00 00  00 00
```

Fig: B

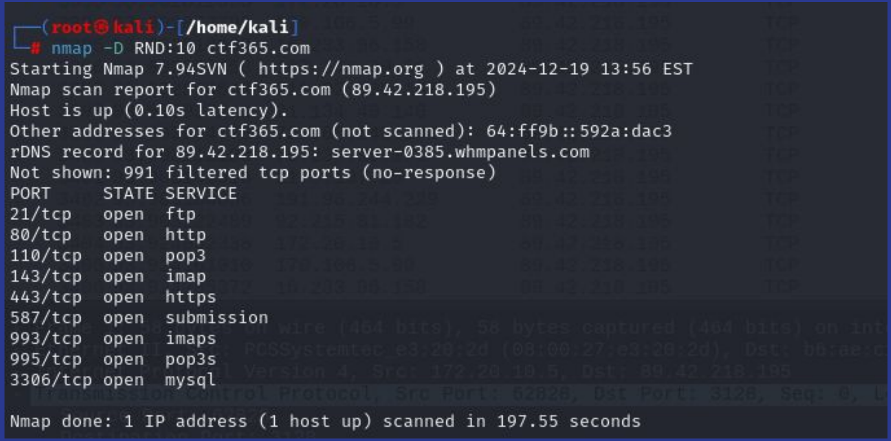
3. IP Address decoy:

Decoying IP addresses is a network spying method that is mostly used to hide the true source of a scan. This technique involves inserting several fictitious IP addresses (decoys) in the packet headers together with the attacker's actual IP address. Tools such as Nmap facilitate this technique.

The decoy can be performed in two ways.

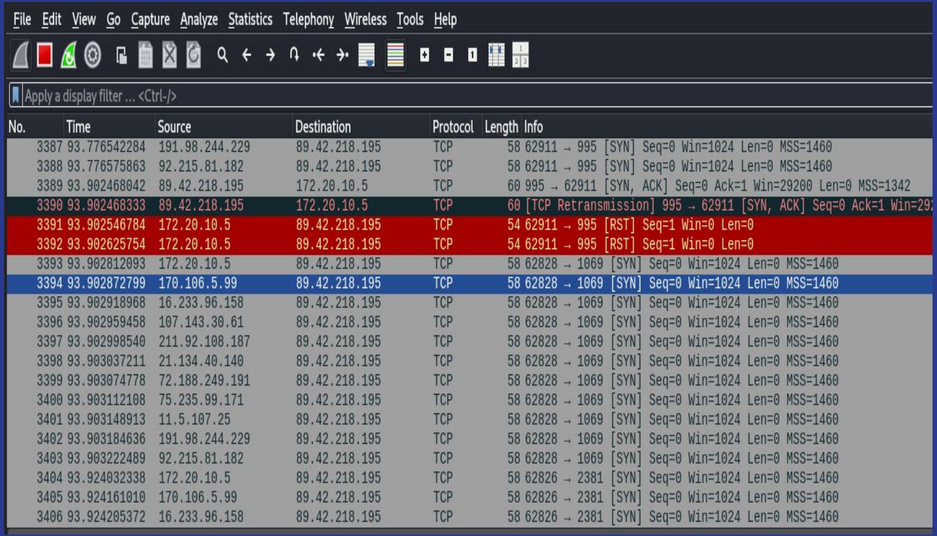
Static Decoy:
Specify a fixed list of decoy IP addresses manually.
Syntax 1: nmap -D decoy 1, decoy 2, decoy 3, etc

Random Decoy:
Use a specified number of randomly generated decoy IPs.
Syntax 2: nmap _D RND: number target-ip/ target- website



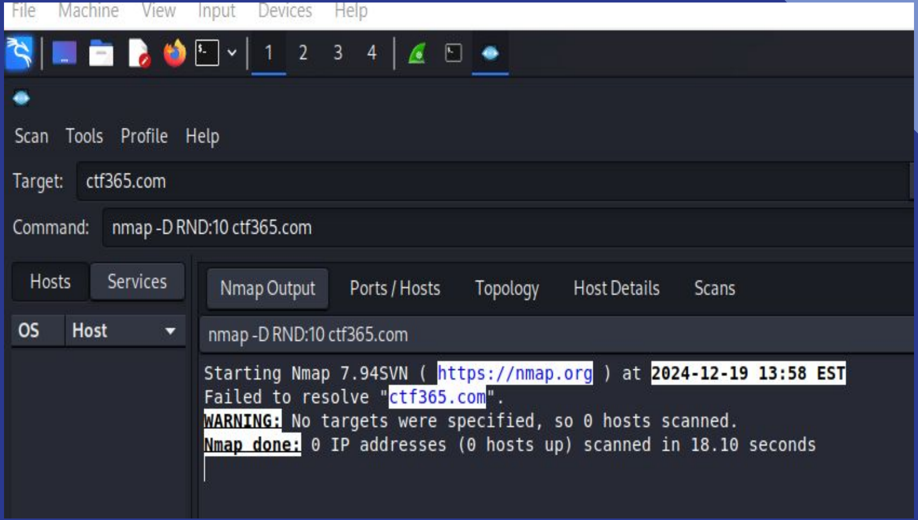
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -D RND:10 ctf365.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 13:56 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.10s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 197.55 seconds
```

In Fig: A, the execution of command, shows the RND as the random saple of 10 for the web address of ctf365.com, the out result in the port numbers, state and services.


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

No.	Time	Source	Destination	Protocol	Length	Info
3387	93.776542284	191.98.244.229	89.42.218.195	TCP	58	62911 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3388	93.776575863	92.215.81.182	89.42.218.195	TCP	58	62911 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3389	93.902468042	89.42.218.195	172.20.10.5	TCP	60	995 → 62911 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1342
3390	93.902468333	89.42.218.195	172.20.10.5	TCP	60	[TCP Retransmission] 995 → 62911 [SYN, ACK] Seq=0 Ack=1 Win=29
3391	93.902546784	172.20.10.5	89.42.218.195	TCP	54	62911 → 995 [RST] Seq=1 Win=0 Len=0
3392	93.902625754	172.20.10.5	89.42.218.195	TCP	54	62911 → 995 [RST] Seq=1 Win=0 Len=0
3393	93.902812093	172.20.10.5	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3394	93.902872799	170.106.5.99	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3395	93.902918968	16.233.96.158	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3396	93.902959458	107.143.30.61	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3397	93.902998540	211.92.108.187	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3398	93.903037211	21.134.40.140	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3399	93.903074778	72.188.249.191	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3400	93.903112108	75.235.99.171	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3401	93.903148913	11.5.107.25	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3402	93.903184636	191.98.244.229	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3403	93.903222489	92.215.81.182	89.42.218.195	TCP	58	62828 → 1069 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3404	93.924032338	172.20.10.5	89.42.218.195	TCP	58	62826 → 2381 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3405	93.924161010	170.106.5.99	89.42.218.195	TCP	58	62826 → 2381 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3406	93.924205372	16.233.96.158	89.42.218.195	TCP	58	62826 → 2381 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Fig: B


File Machine View Input Devices Help
1 2 3 4
Scan Tools Profile Help
Hosts Services
Nmap Output Ports / Hosts Topology Host Details Scans
OS Host

Target: ctf365.com

Command: nmap -D RND:10 ctf365.com

```
nmap -D RND:10 ctf365.com

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 13:58 EST
Failed to resolve "ctf365.com".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 18.10 seconds
```

Fig: C

The result of the wireshark and the Nmap are shown in the Fig: B and Fig: C

Perform NSE scanning for testing network/website

The **Nmap Scripting Engine (NSE)** is a powerful feature of Nmap that extends its capabilities by using scripts to perform a wide variety of tasks. These tasks range from gathering information about the target system to exploiting vulnerabilities.

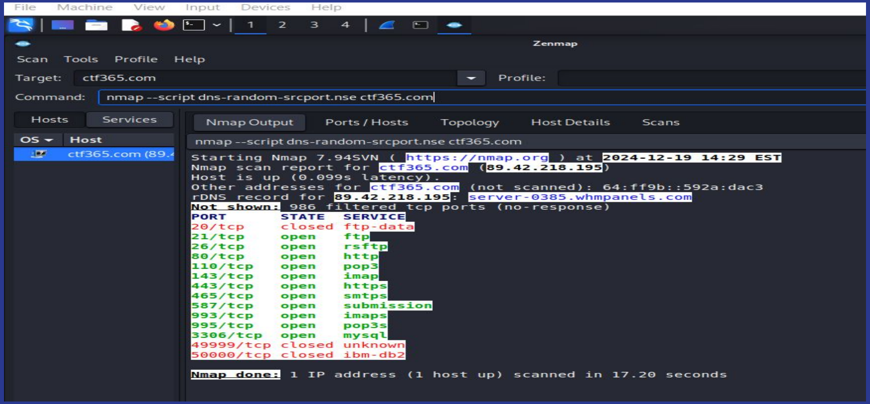
NSE scripts define a list of categories they belong to. Currently defined categories are auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version and vuln. Category names are not case sensitive.

Features of NSE Scanning

1. **Extensibility**: Custom scripts can be written in Lua, allowing users to tailor the scanning process.
2. **Automation**: Automates complex tasks like vulnerability assessment, brute-forcing, or service version detection.
3. **Wide Range of Scripts**: Nmap includes hundreds of pre-written scripts categorized by their function.
4. **Customizability**: Users can control which scripts to run and pass arguments to modify behavior.

1.dns-random-srcport.nse

An Nmap Scripting Engine (NSE) script called dns-random-srcport.nse checks DNS servers for vulnerability to DNS cache poisoning attacks. It accomplishes this by verifying that source port randomization, a crucial security feature to guard against spoof answers, is implemented correctly by the server.

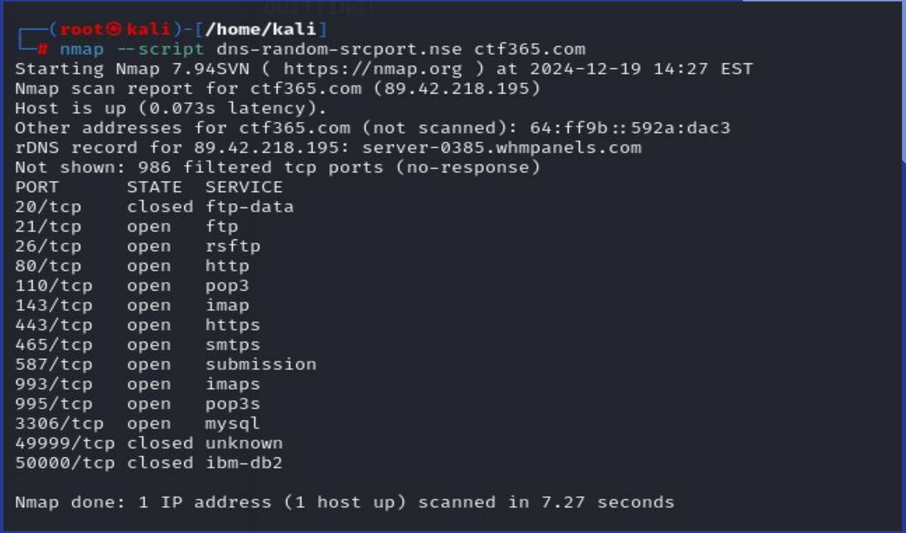


Target: ctf365.com

Profile:

Command: nmap --script dns-random-srcport.nse ctf365.com

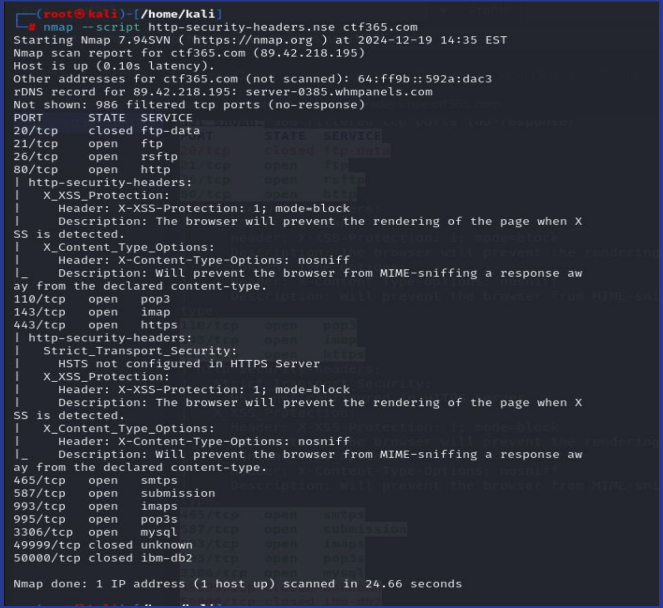
```
nmap --script dns-random-srcport.nse ctf365.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:29 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.099s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 986 filtered tcp ports (no-response)
PORT       STATE  SERVICE
20/tcp     closed ftp-data
21/tcp     open   ftp
26/tcp     open   rsftp
80/tcp     open   http
110/tcp    open   pop3
143/tcp    open   imap
443/tcp    open   https
465/tcp    open   smtps
587/tcp    open   submission
993/tcp    open   imaps
995/tcp    open   pop3s
3306/tcp   open   mysql
49999/tcp  closed unknown
50000/tcp  closed ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 17.20 seconds
```



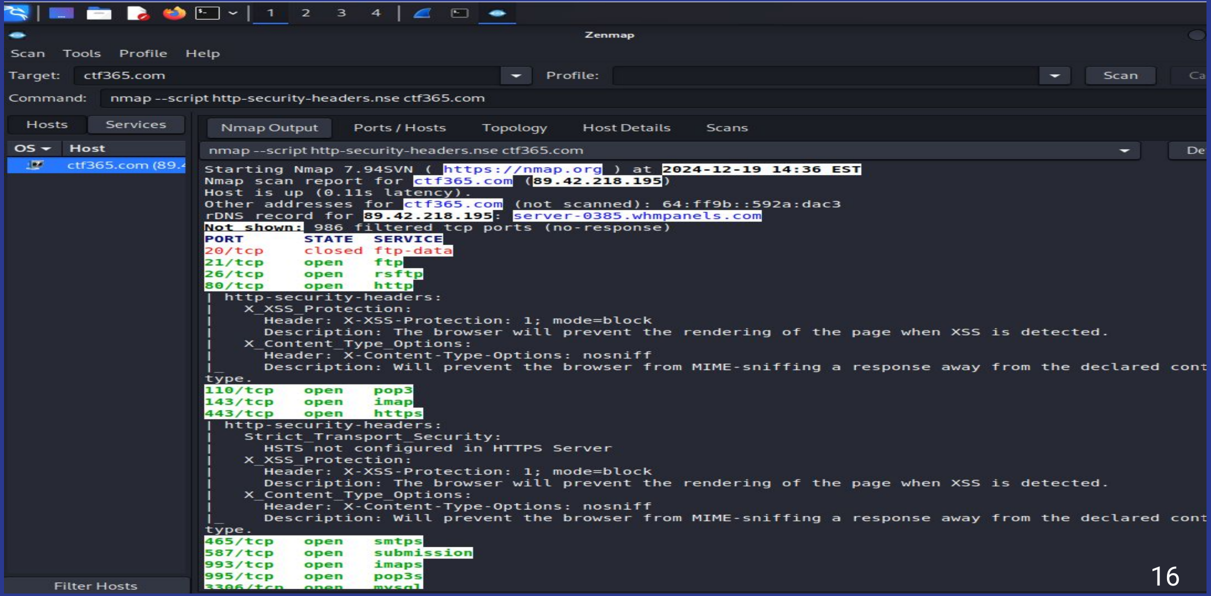
```
┌──(root㉿kali)-[/home/kali]
└─# nmap --script dns-random-srcport.nse ctf365.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:27 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.073s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 986 filtered tcp ports (no-response)
PORT       STATE  SERVICE
20/tcp     closed ftp-data
21/tcp     open   ftp
26/tcp     open   rsftp
80/tcp     open   http
110/tcp    open   pop3
143/tcp    open   imap
443/tcp    open   https
465/tcp    open   smtps
587/tcp    open   submission
993/tcp    open   imaps
995/tcp    open   pop3s
3306/tcp   open   mysql
49999/tcp  closed unknown
50000/tcp  closed ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 7.27 seconds
```

2. http-security-headers.nse script

This Script checks for the HTTP response headers related to security given in OWASP secure headers project and gives a brief description of the header and its configuration value.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap --script http-security-headers.nse ctf365.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:35 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.10s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 986 filtered tcp ports (no-response)
PORT      STATE  SERVICE
20/tcp    closed ftp-data
21/tcp    open   ftp
26/tcp    open   rsftp
80/tcp    open   http
| http-security-headers:
|   X_XSS_Protection:
|     Header: X-XSS-Protection: 1; mode=block
|     Description: The browser will prevent the rendering of the page when X
SS is detected.
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|_    Description: Will prevent the browser from MIME-sniffing a response aw
ay from the declared content-type.
110/tcp   open   pop3
143/tcp   open   imap
443/tcp   open   https
| http-security-headers:
|   Strict_Transport_Security:
|     HSTS not configured in HTTPS Server
|   X_XSS_Protection:
|     Header: X-XSS-Protection: 1; mode=block
|     Description: The browser will prevent the rendering of the page when X
SS is detected.
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|_    Description: Will prevent the browser from MIME-sniffing a response aw
ay from the declared content-type.
465/tcp   open   smtps
587/tcp   open   submission
993/tcp   open   imaps
995/tcp   open   pop3s
3306/tcp  open   mysql
49999/tcp closed unknown
50000/tcp closed ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 24.66 seconds
```



Target: ctf365.com

Profile:

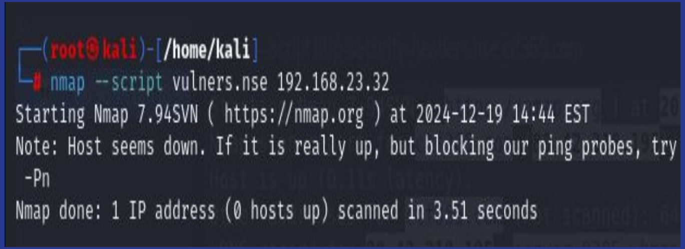
Command: nmap --script http-security-headers.nse ctf365.com

```
nmap --script http-security-headers.nse ctf365.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:36 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.11s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 986 filtered tcp ports (no-response)
PORT      STATE  SERVICE
20/tcp    closed ftp-data
21/tcp    open   ftp
26/tcp    open   rsftp
80/tcp    open   http
| http-security-headers:
|   X_XSS_Protection:
|     Header: X-XSS-Protection: 1; mode=block
|     Description: The browser will prevent the rendering of the page when XSS is detected.
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|_    Description: Will prevent the browser from MIME-sniffing a response away from the declared cont
type.
110/tcp   open   pop3
143/tcp   open   imap
443/tcp   open   https
| http-security-headers:
|   Strict_Transport_Security:
|     HSTS not configured in HTTPS Server
|   X_XSS_Protection:
|     Header: X-XSS-Protection: 1; mode=block
|     Description: The browser will prevent the rendering of the page when XSS is detected.
|   X_Content_Type_Options:
|     Header: X-Content-Type-Options: nosniff
|_    Description: Will prevent the browser from MIME-sniffing a response away from the declared cont
type.
465/tcp   open   smtps
587/tcp   open   submission
993/tcp   open   imaps
995/tcp   open   pop3s
3306/tcp  open   mysql
```

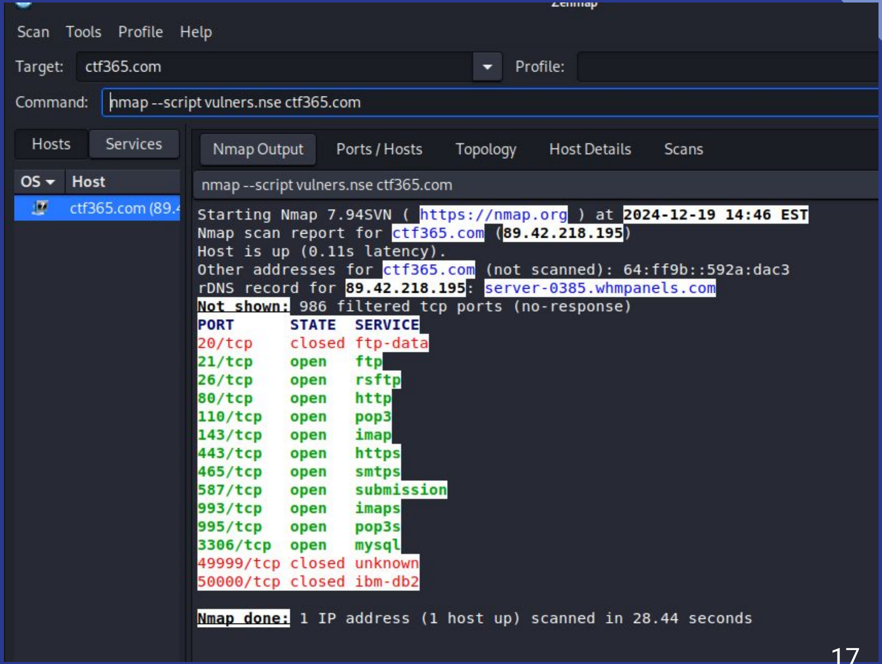
3. Vulners.nse

For each available CPE the script prints out known vulns and correspondent CVSS scores.

- Works only when some software version is identified for an open port
- Take all the known CPEs for that software.
- Make a request to remote servers to learn whether any known vulns exist for that CPE.



```
┌──(root㉿kali)-[/home/kali]
└─# nmap --script vulners.nse 192.168.23.32
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:44 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try
 -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.51 seconds
```



Target: ctf365.com

Profile:

Command: nmap --script vulners.nse ctf365.com

```
nmap --script vulners.nse ctf365.com

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-19 14:46 EST
Nmap scan report for ctf365.com (89.42.218.195)
Host is up (0.11s latency).
Other addresses for ctf365.com (not scanned): 64:ff9b::592a:dac3
rDNS record for 89.42.218.195: server-0385.whmpanels.com
Not shown: 986 filtered tcp ports (no-response)
PORT      STATE  SERVICE
20/tcp    closed ftp-data
21/tcp    open   ftp
26/tcp    open   rsftp
80/tcp    open   http
110/tcp   open   pop3
143/tcp   open   imap
443/tcp   open   https
465/tcp   open   smtps
587/tcp   open   submission
993/tcp   open   imaps
995/tcp   open   pop3s
3306/tcp  open   mysql
49999/tcp closed unknown
50000/tcp closed ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 28.44 seconds
```

Conclusion

- Network scanning with Wireshark and Nmap, enhanced by the Nmap Scripting Engine (NSE), provides robust tools for analyzing network security and identifying vulnerabilities. By leveraging Nmap's comprehensive scanning capabilities alongside Wireshark's in-depth packet analysis, we can uncover potential security gaps, monitor network traffic, and improve overall protection strategies.
- The integration of NSE scripts significantly enhances Nmap's versatility, enabling automated vulnerability detection and customized scans tailored to specific needs. Meanwhile, Wireshark delivers granular packet-level insights, making it easier to track network behavior and identify unusual activity.
- This project underscores the critical role of advanced scanning and analysis tools in strengthening cybersecurity measures. It also highlights the importance of ethical practices and legal compliance to ensure responsible and effective use of these powerful technologies.

Reference

1. https://nmap.org/book/nse-usage.html#nse-script-selection
2. https://www.tecmint.com/use-nmap-script-engine-nse-scripts-in-linux/
3. https://nmap.org/book/man-nse.html
4. McLeman, C., A technical investigation into port scanning using Nmap.
5. https://ieeexplore.ieee.org/abstract/document/9002531

THANK YOU