

Scanning Using Nmap and Wireshark

By Meenu Handa

<https://www.linkedin.com/in/handameenu/>

Warning

- This project is conducted strictly for educational and ethical research purposes.
- All testing, scanning, and analysis should be performed only on systems and networks you own or have explicit permission to test.
- Unauthorized activities may violate legal regulations and ethical guidelines.
- Always follow responsible disclosure practices, respect privacy laws, and ensure compliance with relevant cybersecurity policies.

Agenda

- Introduction about project
- Introduction about tools:
 - Introduction about Wireshark
 - Introduction about Nmap and NSE
- Project requirement and testing website/network
- Wireshark IDS/Firewall evade techniques
- NSE scanning for testing network/website
- Conclusion

Introduction

Network scanning is an essential technique used by security professionals to analyze network structures, detect open ports, and assess security defenses.

This project focuses on network scanning using Nmap and Wireshark, two powerful tools for network reconnaissance and traffic analysis. We will explore:

- Performing network scans using Nmap and its Nmap Scripting Engine (NSE) to gather information about target systems.
- Using Wireshark to analyze network traffic and test firewall evasion techniques.

Additionally, I will apply three NSE scripts for security testing and demonstrate three firewall evasion techniques using Wireshark. This hands-on approach will provide practical insights into ethical hacking, network security assessment, and the importance of proactive defense strategies.

Tools: Wireshark

Wireshark is a network protocol analyzer widely used for network troubleshooting, traffic analysis, and cybersecurity investigations. It helps in:

- **Packet Capture & Inspection** – Monitors and analyzes network packets in real time.
- **Protocol Analysis** – Supports various network protocols to decode and interpret data transmission.
- **Security Analysis** – Detects anomalies, suspicious traffic, and intrusion attempts.
- **Firewall Evasion Techniques** – Helps understand how attackers bypass security measures.
- **Performance Monitoring** – Identifies network bottlenecks and optimizes performance.

Tools: Nmap and NSE



Nmap (Network Mapper)

Nmap is a powerful open-source tool used for network discovery, security auditing, and vulnerability assessment. It helps cybersecurity professionals map networks, detect live hosts, and identify open ports and running services.

- Network Scanning – Discovers hosts, services, and open ports on a target network.
- OS and Service Detection – Identifies operating systems, versions, and running services.
- Vulnerability Assessment – Detects misconfigurations and security weaknesses.

NSE (Nmap Scripting Engine)

NSE enhances Nmap's capabilities by allowing customized automation for advanced security testing.

- Automates scanning tasks such as vulnerability detection and malware discovery.
- Uses specialized scripts for enumeration, brute-force attacks, and exploit detection.
- Flexible and customizable, making it a powerful tool for ethical hacking and security assessments.

Project Requirement and Testing Websites

- Virtual Environment Setup:
 - Oracle VirtualBox – Used for hosting virtual machines.
 - Kali Linux (10.0.0.228) – Security testing and penetration testing tools.
 - Windows Server 2022 (10.0.0.89) – Target system for testing.
- Testing Websites:
 - [Hack This Site](#) – Ethical hacking practice platform.

Wireshark IDS/Firewall evade techniques

Intrusion Detection Systems (IDS) and Firewalls are critical for network security, designed to detect and block malicious traffic. However, attackers use evasion techniques to bypass these security measures, making it challenging for security analysts to detect threats. This presentation explores few common IDS/Firewall evasion techniques, how they work, and how Wireshark can be used to detect them

1. IP Address Decoy
2. Packet Fragmentation
3. Source Port Manipulation

Source Port Manipulation

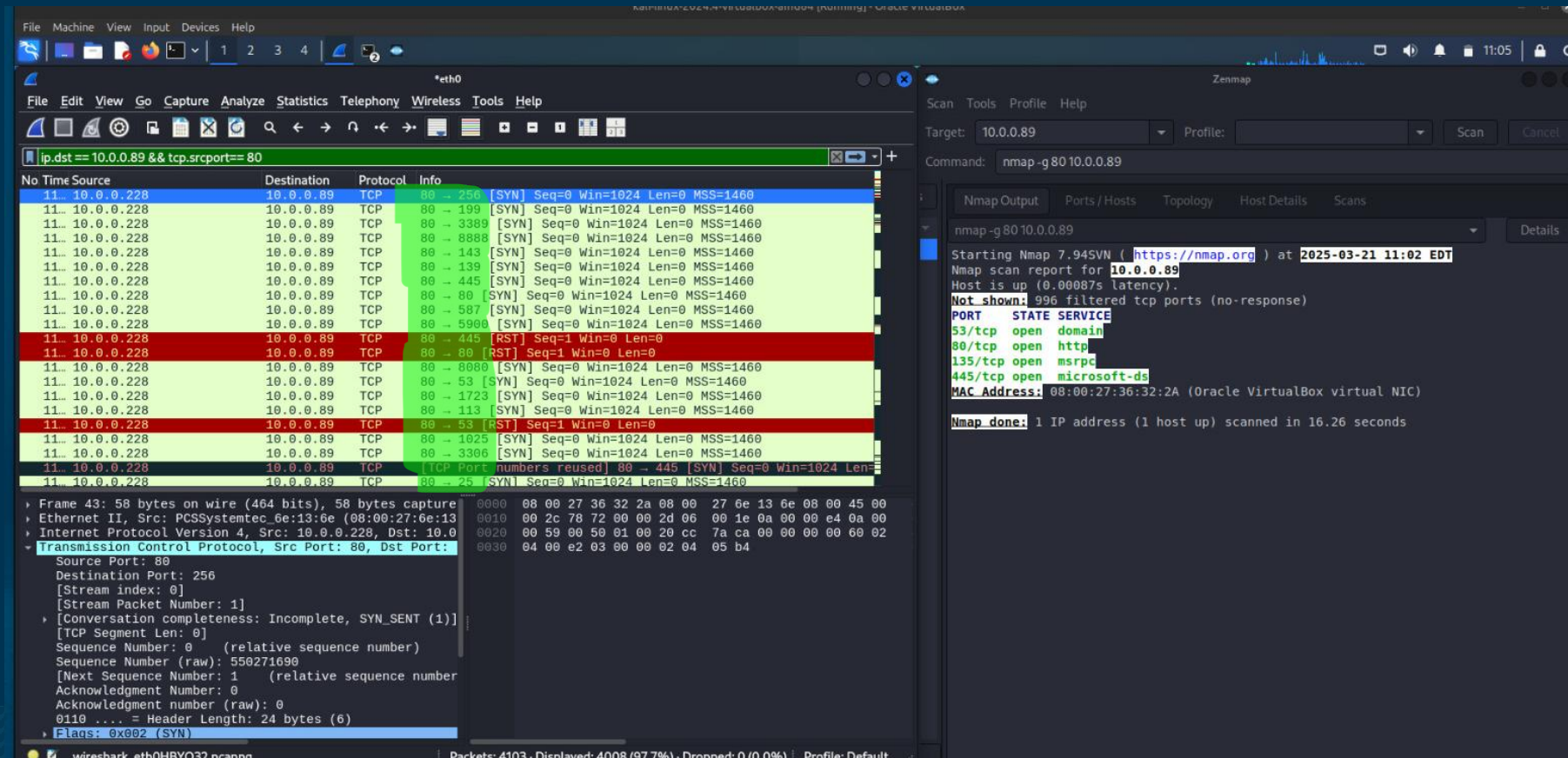
This technique is used to bypass firewall rules and IDS by sending packets from a specific source port that might be trusted by the target system. Some firewalls allow traffic from ports like 53 (DNS), 80 (HTTP), or 443 (HTTPS) while blocking others.

-g or --source-port option in Nmap to specify a custom source port when scanning a target.

Command: `nmap -g 80 10.0.0.89`

All the traffic will be routed from port 80 of Kali machine to all ports of target machine.

Highlighted section in screenshot shows traffic going from port 80 of source machine.



The screenshot displays two windows: Wireshark on the left and Nmap on the right.

Wireshark: The packet list shows a series of TCP SYN packets from 10.0.0.228 to 10.0.0.89. The packet details for the selected packet (Frame 43) show the source port as 80 and the destination port as 256. The packet bytes show the TCP header with the SYN flag set.

Nmap: The command line shows `nmap -g 80 10.0.0.89`. The output shows the scan results for 10.0.0.89, indicating that the host is up and listing open ports: 53/tcp (domain), 80/tcp (http), 135/tcp (msrpc), and 445/tcp (microsoft-ds).

IP Address Decoy

This technique is used to evade IDS and firewalls by making it difficult for defenders to trace the source of a scan. Nmap's `-D` option allows to generate fake IP addresses alongside real IP, confusing security tools that rely on packet logs.

Command:

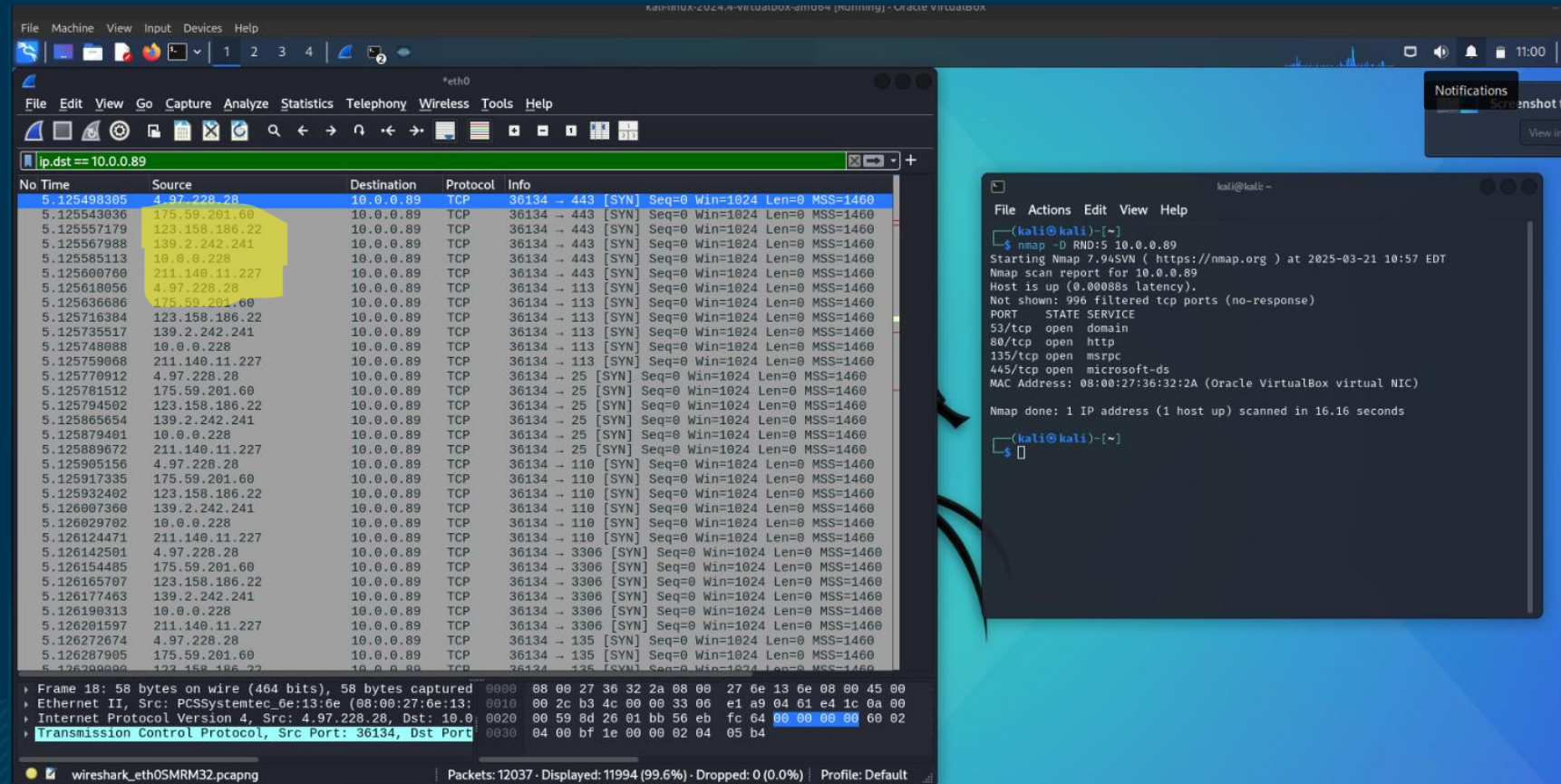
- `nmap -D RND: 5 10.0.0.89`

RND: 5 will generate 5 random fake IP addresses along with target IP address .

- `nmap -D <fake IP1> <fake IP2> <fake IP3> <target IP Address>`

In this if we want to specify fake IP addresses along with target IP address.

Highlighted section in screenshot represent 5 different source IP addresses along with actual source IP address.



The screenshot displays two windows: Nmap scan results and a Wireshark packet capture. The Nmap window shows a scan of 10.0.0.89 with 5 random decoy IP addresses (RND: 5) and the target IP address. The results show 1 IP address (1 host up) scanned in 16.16 seconds. The Wireshark window shows a packet capture of the scan, with the source IP address highlighted as 10.0.0.89.

Nmap Scan Results:

```

kali@kali:~$ nmap -D RND: 5 10.0.0.89
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-21 10:57 EDT
Nmap scan report for 10.0.0.89
Host is up (0.00088s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:36:32:2A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 16.16 seconds
  
```

Wireshark Packet Capture:

No	Time	Source	Destination	Protocol	Info
5	125498305	4.97.228.28	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125543936	175.59.201.60	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125557179	123.158.186.22	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125567988	139.2.242.241	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125585113	10.0.0.228	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125600760	211.140.11.227	10.0.0.89	TCP	36134 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125618056	4.97.228.28	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125636686	175.59.201.60	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125716384	123.158.186.22	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125735517	139.2.242.241	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125748088	10.0.0.228	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125759068	211.140.11.227	10.0.0.89	TCP	36134 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125770912	4.97.228.28	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125781512	175.59.201.60	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125794502	123.158.186.22	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125805654	139.2.242.241	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125879401	10.0.0.228	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125889672	211.140.11.227	10.0.0.89	TCP	36134 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125905156	4.97.228.28	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125917335	175.59.201.60	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	125932402	123.158.186.22	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126007360	139.2.242.241	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126029702	10.0.0.228	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126124471	211.140.11.227	10.0.0.89	TCP	36134 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126142501	4.97.228.28	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126154485	175.59.201.60	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126165707	123.158.186.22	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126177463	139.2.242.241	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126190313	10.0.0.228	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126201597	211.140.11.227	10.0.0.89	TCP	36134 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126272674	4.97.228.28	10.0.0.89	TCP	36134 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126287905	175.59.201.60	10.0.0.89	TCP	36134 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	126300000	123.158.186.22	10.0.0.89	TCP	36134 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 18: 58 bytes on wire (464 bits), 58 bytes captured
 Ethernet II, Src: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e), Dst: 10.0.0.89
 Internet Protocol Version 4, Src: 4.97.228.28, Dst: 10.0.0.89
 Transmission Control Protocol, Src Port: 36134, Dst Port: 443

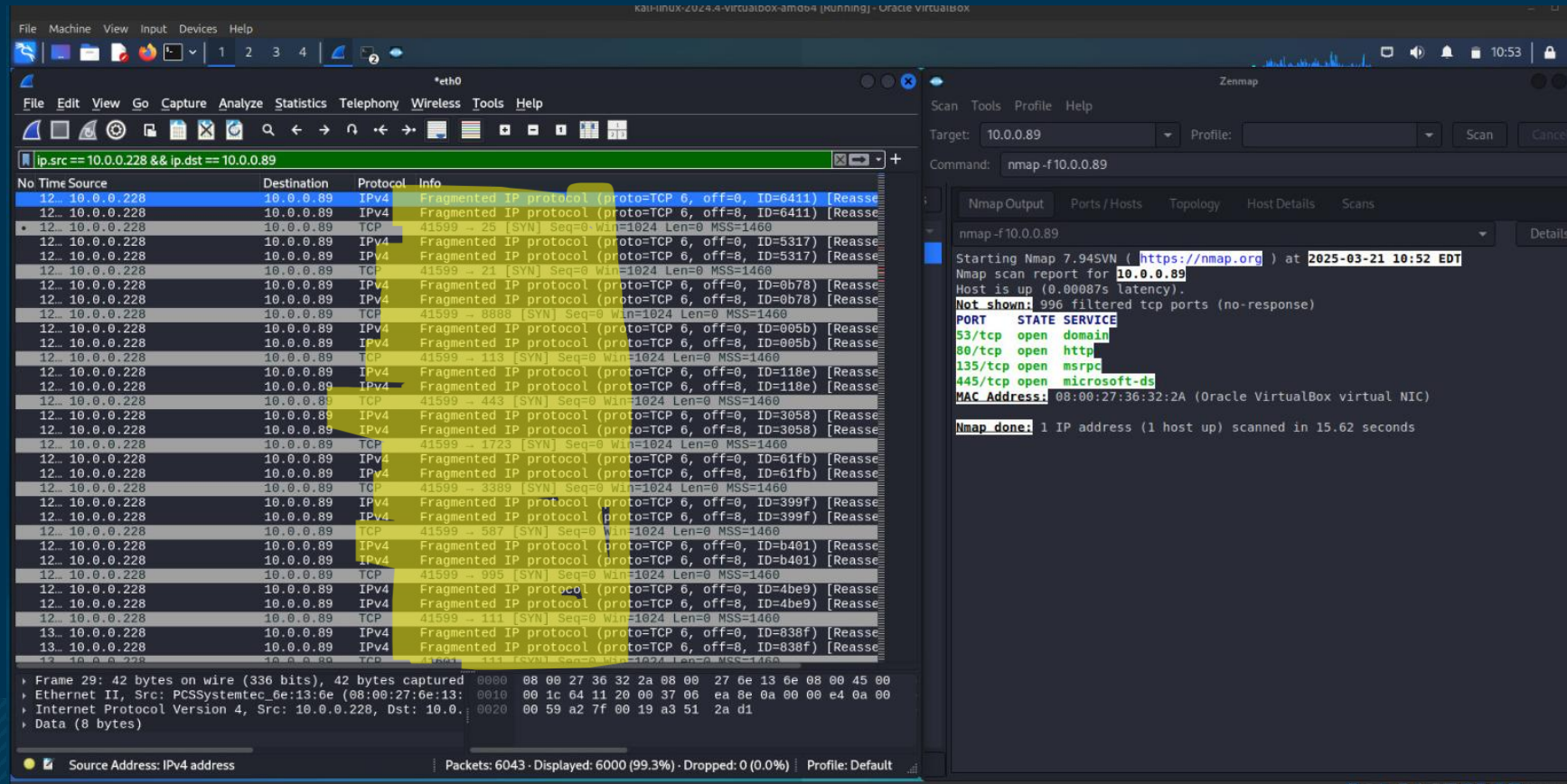
Packet Fragmentation

By splitting scan packets into smaller fragments, some security systems have trouble reconstructing fragmented packets which makes it a useful evasion method. Normally, network packets are sent in full, but with `-f` nmap splits packets into smaller chunks, making detection harder.

Command:

`nmap -f 10.0.0.89`

Normally, network packets are sent in full, but with `-f` nmap splits packets into smaller chunks, making detection harder.



The screenshot shows a Kali Linux virtual machine environment. The top window is Wireshark, displaying a list of network packets captured on the interface `*eth0`. The filter is `ip.src == 10.0.0.228 && ip.dst == 10.0.0.89`. The packet list shows numerous fragmented IP packets, all marked as 'Reassembled'. The bottom window is Nmap, showing the scan results for the target IP `10.0.0.89`. The scan was performed using the command `nmap -f 10.0.0.89`. The results indicate that the host is up and that 996 filtered TCP ports were not shown. The Nmap output is as follows:

```

Nmap scan report for 10.0.0.89
Host is up (0.00007s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:36:32:2A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 15.62 seconds
  
```

Nmap NSE scanning for testing network/website

Nmap Scripting Engine (NSE) is a powerful feature of Nmap that allows users to automate network scanning, vulnerability detection, and service enumeration using scripts. It extends Nmap's functionality beyond basic port scanning, making it a valuable tool for penetration testers, security analysts, and system administrators. Nmap comes with a library of pre-installed NSE scripts, categorized into Discovery, Vulnerability, Exploit, Authentication.

In the next slides, we will execute several Nmap scripts to gather valuable information:

1. MSRPC Enumeration Script
2. HTTP Title Script
3. NetBIOS Information Script

MSRPC Enumeration Script

The msrpc-enum script is used to enumerate Microsoft RPC (Remote Procedure Call) services on a target system. Queries an MSRPC endpoint mapper for a list of mapped services and displays the gathered information.

Command: `nmap --script msrpc-enum.nse <target>`

This script helps

- Identifies Microsoft RPC services running on the target.
- Lists available RPC endpoints and their associated programs.
- Helps assess misconfigurations or security flaws in Windows systems.

```
(root@kali)-[/home/kali]
# nmap --script msrpc-enum.nse 10.0.0.89
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-25 09:21 EDT
Nmap scan report for 10.0.0.89
Host is up (0.00032s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:36:32:2A (Oracle VirtualBox virtual NIC)

Host script results:
|_msrpc-enum: NT_STATUS_ACCESS_DENIED

Nmap done: 1 IP address (1 host up) scanned in 16.87 seconds
```

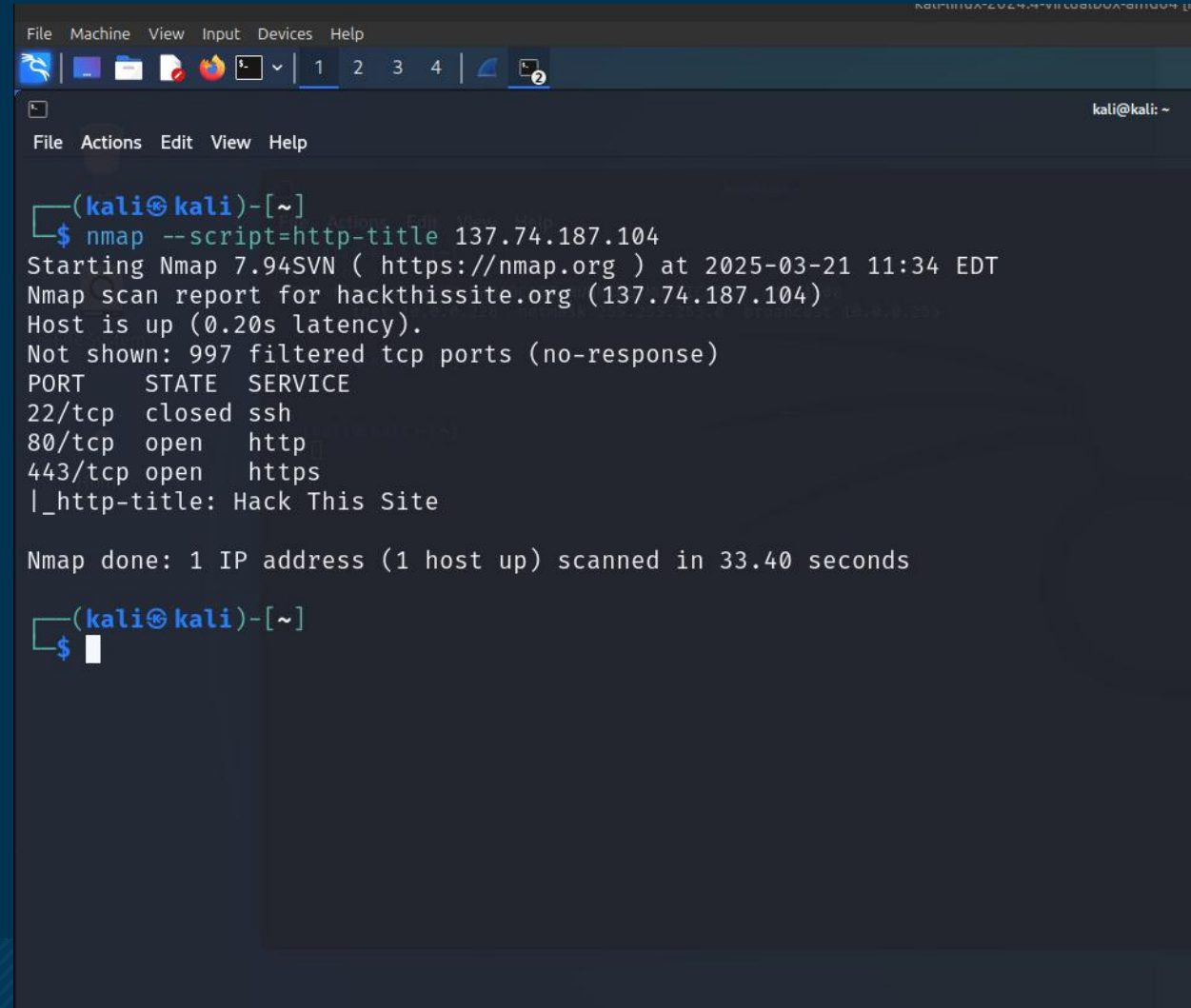
Http title Script

The HTTP Title Script (http-title) extracts the title of a webpage from its HTTP response. This helps identify web applications running on a target.

Command: `nmap --script=http-title <target>`

How it works:

- Sends an HTTP request to the target.
- Retrieves and displays the **title** of the webpage.
- Useful for identifying **web applications** and their purpose.



```
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ nmap --script=http-title 137.74.187.104
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-21 11:34 EDT
Nmap scan report for hackthissite.org (137.74.187.104)
Host is up (0.20s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    open  http
443/tcp    open  https
|_http-title: Hack This Site

Nmap done: 1 IP address (1 host up) scanned in 33.40 seconds

(kali@kali)-[~]
$
```


NetBIOS Information Script

The nbstat script retrieves NetBIOS name table information from a target system. It is useful for network reconnaissance and Windows environment enumeration.

Command: `nmap --script nbstat.nse <target>`

Nbstat:

- Queries the NetBIOS Name Service (NBNS) on UDP port 137.
- Retrieves hostnames, workgroups, and domain names.
- Helps identify network shares, misconfigurations, and potential attack vectors.

```
(root@kali)-[/home/kali]
# nmap --script nbstat.nse 10.0.0.89
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-25 09:24 EDT
Nmap scan report for 10.0.0.89
Host is up (0.00028s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:36:32:2A (Oracle VirtualBox virtual NIC)

Host script results:
| nbstat: NetBIOS name: WIN-I900SLUNDJ1, NetBIOS user: <unknown>, NetBIOS
| Names:
|   WIN-I900SLUNDJ1<20>  Flags: <unique><active>
|   WIN-I900SLUNDJ1<00>  Flags: <unique><active>
|   WORKGROUP<00>       Flags: <group><active>
|   WORKGROUP<1e>       Flags: <group><active>
|   WORKGROUP<1d>       Flags: <unique><active>
|_  \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>

Nmap done: 1 IP address (1 host up) scanned in 16.02 seconds
```

Conclusion

In this project, Wireshark and Nmap NSE are utilized to analyze network traffic, test security defenses, and identify vulnerabilities.

- Wireshark helped to capture and analyze network packets, allow to detect anomalies, inspect communication patterns, and evaluate firewall/IDS evasion techniques.
- Nmap NSE provided powerful script-based scanning capabilities, enabling the collection of detailed service information, detection of vulnerabilities, and automation of security assessments.
- Various Nmap commands executed during the testing process, including msrpc-enum, http-title, and nbstat, helped explore open ports, identify running services, determine software versions, and assess potential security risks in target systems.

Reference

- [Firewall/IDS Evasion and Spoofing | Nmap Network Scanning](#)
- [Nmap Scripting Engine \(NSE\) | Nmap Network Scanning](#)

Thank You