Assignment-3

Course: Cybersecurity Fundamentals

University of Adelaide

Name: Karan Sethi

Student ID: a1745936

**Note - Final Answers are in blue font for ease of finding/marking**

**Answer 1**: The command on zenmap used to find out the information on host 10.0.0.17 was -

|  |
| --- |
| # nmap -T4 -O -A -v 10.0.0.17 |

The primary information after the scan using zenmap returned was -

|  |
| --- |
| PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 7.4 (protocol 2.0) | ssh-hostkey: | 2048 3e:e3:e5:21:84:e0:b8:2b:8a:cb:d6:cc:d2:88:a2:be (RSA) | 256 2e:3f:bd:86:ca:6e:60:91:df:f8:3d:0d:10:ed:a4:7d (ECDSA) |\_ 256 c2:f6:1e:4e:73:6a:96:3c:3d:77:16:62:94:45:d6:88 (ED25519) 53/tcp open domain ISC BIND 9.9.4 (RedHat Enterprise Linux 7) | dns-nsid: |\_ bind.version: 9.9.4-RedHat-9.9.4-73.el7\_6 80/tcp open http nginx 1.12.2 | http-methods: |\_ Supported Methods: GET HEAD |\_http-server-header: nginx/1.12.2 |\_http-title: Test Page **for** the Nginx HTTP Server on Fedora 443/tcp closed https Aggressive OS guesses: Linux 3.10 - 4.11 (94%), Linux 3.2 - 4.9 (91%), Linux 3.13 (90%), Linux 3.13 or 4.2 (90%), Linux 4.10 (90%), Linux 4.2 (90%), Linux 4.4 (90%), Asus RT-AC66U WAP (90%), Linux 3.10 (90%), Linux 3.11 - 3.12 (90%) No exact OS matches **for** host (test conditions non-ideal). Uptime guess: 21.682 days (since Fri Mar 1 22:07:29 2019) Network Distance: 1 hop TCP Sequence Prediction: Difficulty=262 (Good luck!) IP ID Sequence Generation: All zeros Service Info: OS: Linux; CPE: cpe:/o:redhat:enterprise\_linux:7  TRACEROUTE (using port 443/tcp) HOP RTT ADDRESS 1 34.38 ms 10.0.0.17  NSE: Script Post-scanning. Initiating NSE at 15:29 Completed NSE at 15:29, 0.00s elapsed Initiating NSE at 15:29 Completed NSE at 15:29, 0.00s elapsed |

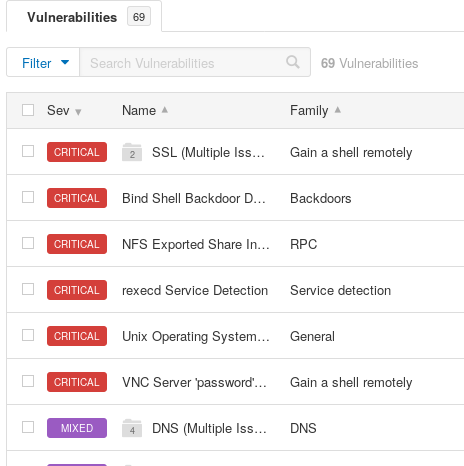
As we can see, the server is most likely running on the Red Hat Enterprise Linux 7 OS and using the following open listening ports and services -

1. Port 22 - OpenSSH 7.4
2. Port 53 - ISC BIND 9.9.4
3. Port 80 - nginx 1.12.2

Hence, the DNS server software and version on host 10.0.0.17 port 53 is -

ISC BIND with version 9.9.4

**Answer 2**: Based on the advanced full Nessus scan on 10.0.0.21 given in screenshot below -



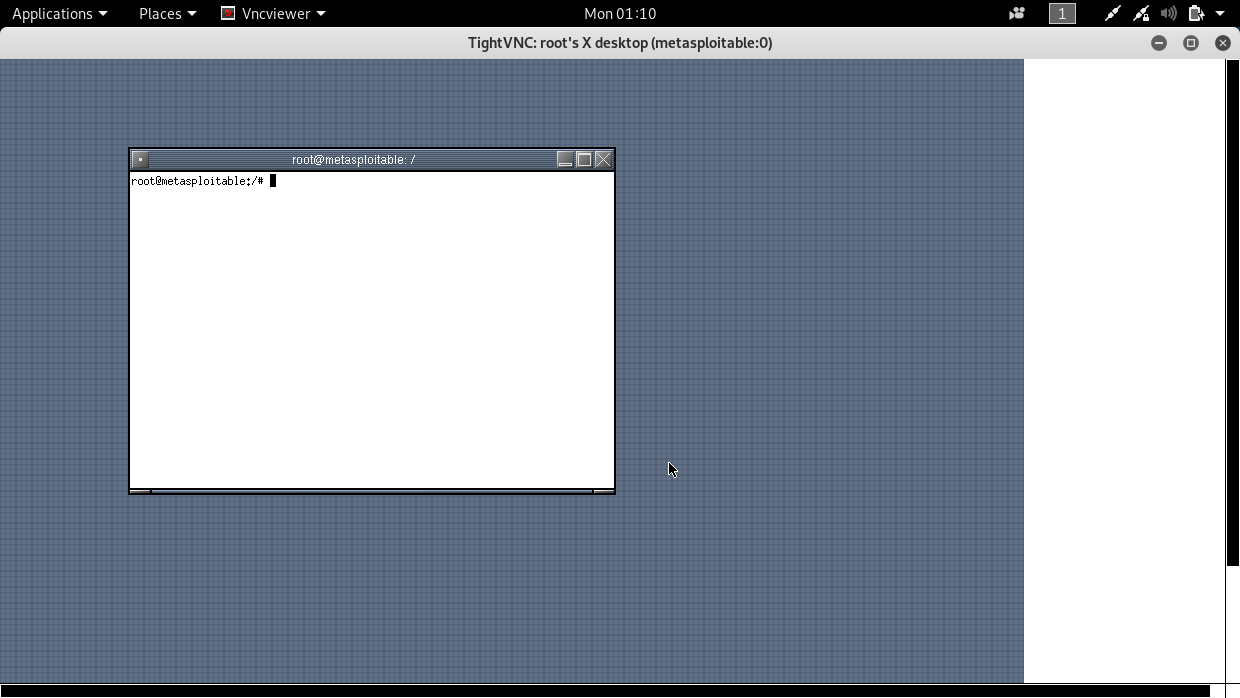
Multiple critical vulnerabilities were explored and it was tried to get a screenshot using a remote shell into the server using telnet and netcat. However none worked as its not possible to install software on the server to take a screenshot. One of the critical vulnerabilities in VNC having a simple password ‘password’ allowed to take a screenshot -

|  |
| --- |
| CRITICAL VNC Server 'password' Password **Description** The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system. |

The host 10.0.0.21 was connected using vnc viewing command -

‘vncviewer 10.0.0.21’ and as given by nessus ‘password’ was used as the password.

Screenshot of the GUI of the server is -



**Answer 3:** The following nmap command was run with 3 decoys to capture packets using wireshark and see spoofed packets along with real IP address to do a sweep on the 10.0.0.21 host randomly selected -

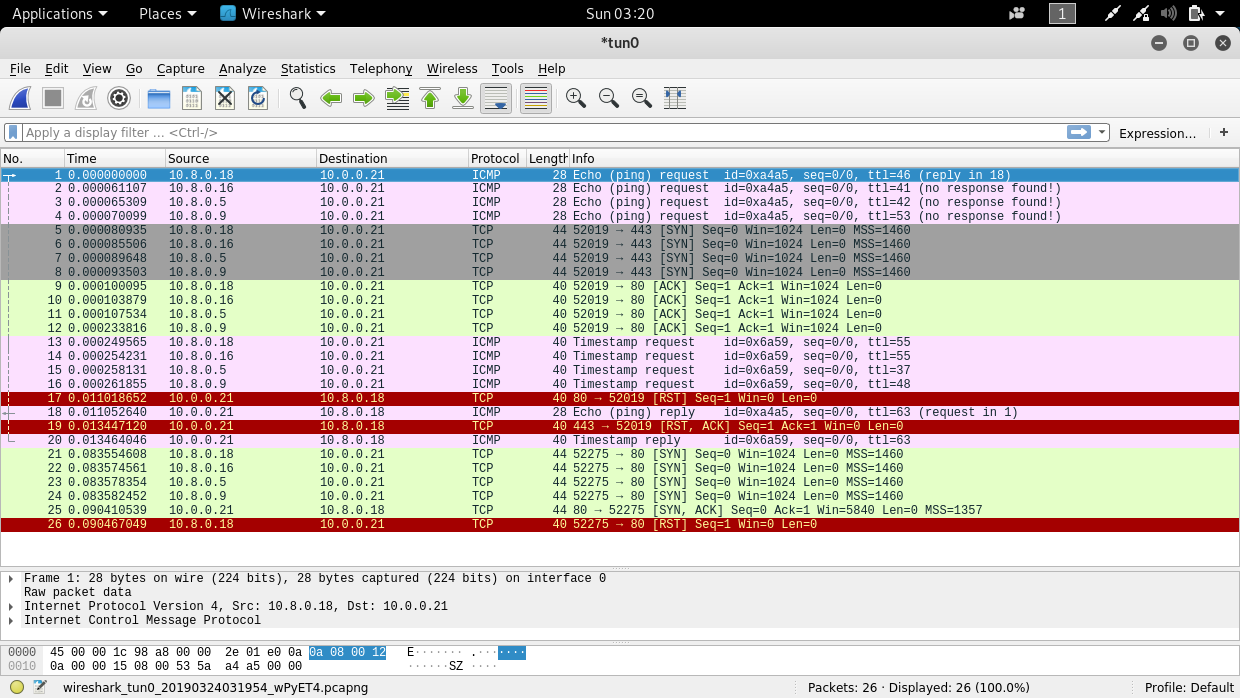
|  |
| --- |
| # nmap -v -n -p 80 -D10.8.0.16,10.8.0.5,10.8.0.9 10.0.0.21 |

Hosts currently online on the same network were discovered using the nmap command -

|  |
| --- |
| # nmap -sn 10.8.0.0/24 |

Some of these hosts were used as decoys - 10.8.0.16, 10.8.0.5, 10.8.0.9

The following wireshark screenshot shows spoof packets from 3 other ips along with original ip of my own computer -



While this can be defeated through router path tracing, response-dropping, and other active mechanisms, it is generally an effective technique for hiding IP address for a black hat hacker.

With -D option it appears to the remote host that the host(s) we specify as decoys are scanning the target network too, as we can see in wireshark. Thus their intruder detection system might report 5-10 port scans from unique IP addresses, but they won’t know which IP was scanning them and which were innocent decoys.

Reference used - <https://www.cyberciti.biz/tips/nmap-hide-ipaddress-with-decoy-ideal-scan.html>

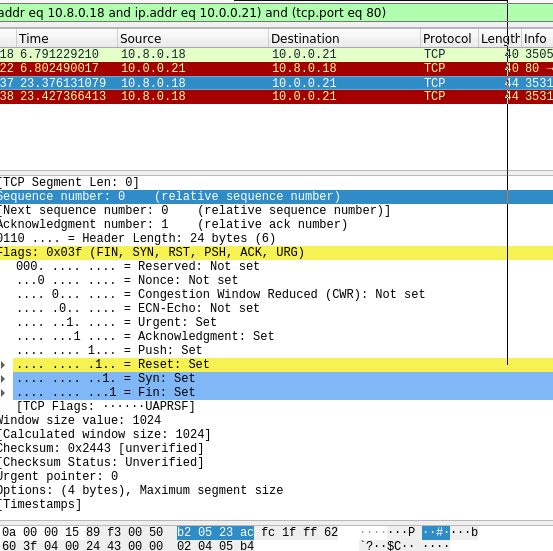
**Answer 4**:

The command used for all flags was -

|  |
| --- |
| # nmap -p 80 -v --scanflags FINSYNRSTPSHACKURG 10.0.0.21 |

If we do the ALL flags option with nmap, it gives 8 flags set to 1.

Screenshot with all 6 flags set to 1 and sequence number 0-



**Answer 5**:

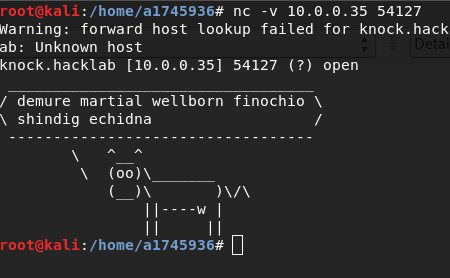
The command used to find the network service between the range 20000 and 60000 on 10.0.0.35 was -

|  |
| --- |
| # nmap -sS -p 20000-60000 -n -Pn 10.0.0.35 |

The command used to connect to the port 54127 on 10.0.0.35 using netcat was -

|  |
| --- |
| # nc -v 10.0.0.35 54127 |

The secret found in the screenshot -



|  |
| --- |
| / demure martial wellborn finochio \ \ shindig echidna /  ----------------------------------  \ ^\_\_^  \ (oo)\\_\_\_\_\_\_\_  (\_\_)\ )\/\  ||----w |  || || |
|  |

**Answer 6**:

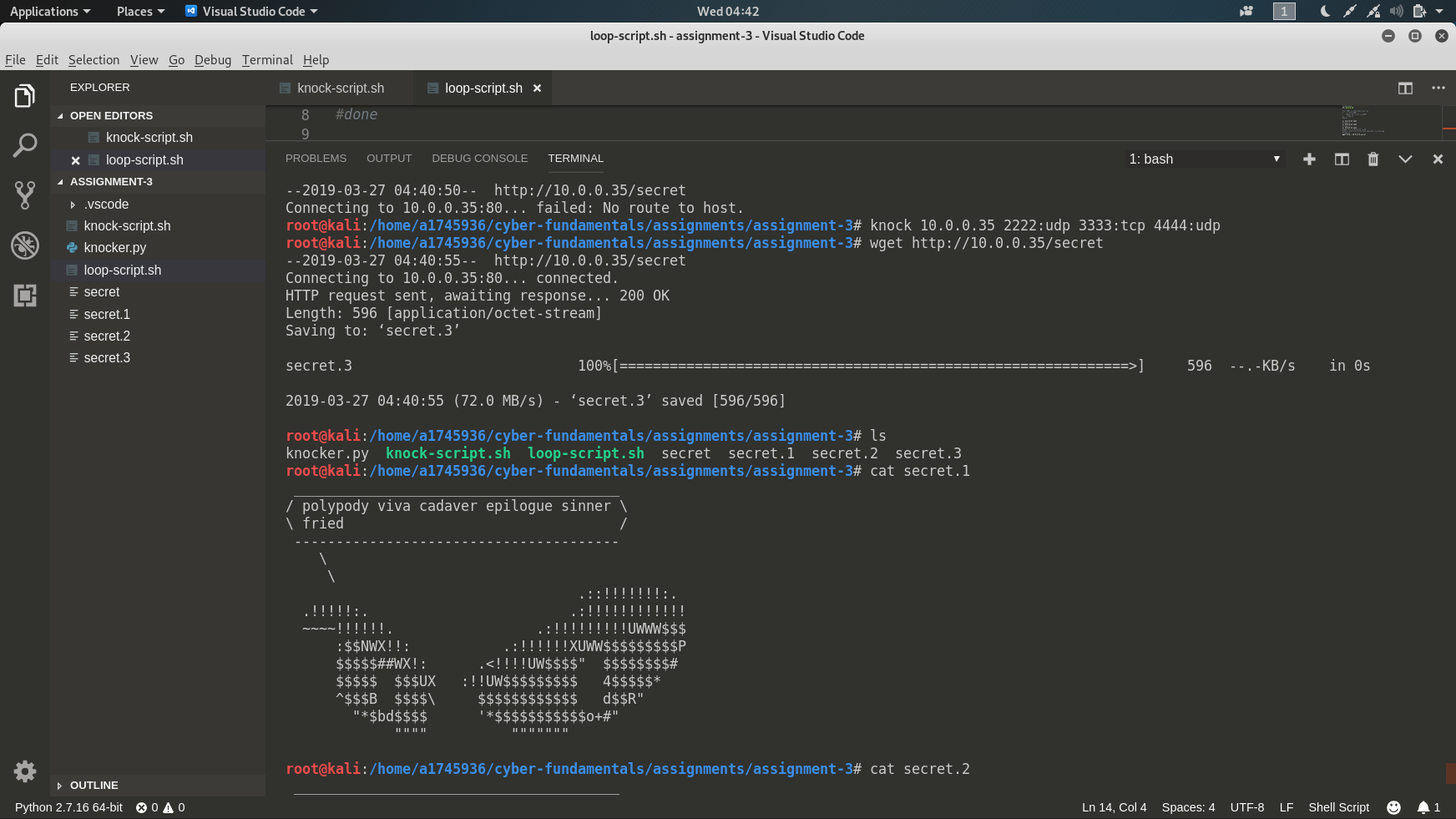
Multiple commands to send packets to 10.0.0.35 were tried like nc, nmap and hping3 inside scripts with loops and sleep timers, based on some internet references. However after understanding port knocking and how to access a port from a paper on the subject -

Ref: <https://pen-testing.sans.org/resources/papers/gcih/knock-bro-116990>,

The following commands worked -

|  |
| --- |
| # knock 10.0.0.35 2222:udp 3333:tcp 4444:udp # wget http://10.0.0.35/secret |

Here is a screenshot of the terminal showing the commands and the secret -



As seen, the secret revealed at <http://10.0.0.35/secret> was (slightly distorted due to pasting) -

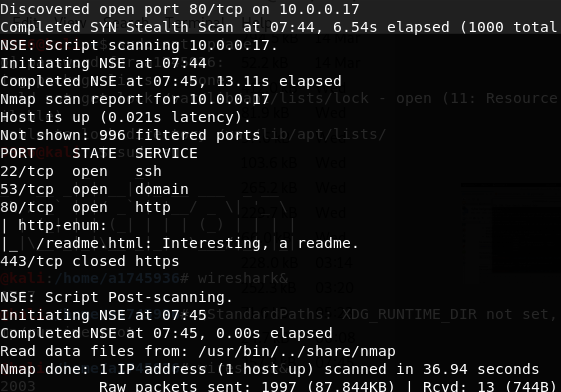
|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / polypody viva cadaver epilogue sinner \ \ fried /  ---------------------------------------  \  \  .::!!!!!!!:.  .!!!!!:. .:!!!!!!!!!!!!  ~~~~!!!!!!. .:!!!!!!!!!UWWW$$$  :$$NWX!!: .:!!!!!!XUWW$$$$$$$$$P  $$$$$##WX!: .<!!!!UW$$$$" $$$$$$$$#  $$$$$ $$$UX :!!UW$$$$$$$$$ 4$$$$$\*  ^$$$B $$$$\ $$$$$$$$$$$$ d$$R"  "\*$bd$$$$ '\*$$$$$$$$$$$o+#"  """" """"""" |

**Answer 7**:

Using the http-enum script with nmap the command was -

|  |
| --- |
| # nmap -v --script http-enum.nse 10.0.0.17 |

The main output was -

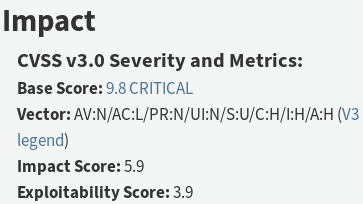


The file was opened in the browser using the url 10.0.0.17/readme.html and the contents of the file was -

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / faraway hallow souse plebeian barking \ \ groove / --------------------------------------- \ / \ //\ \ |\\_\_\_/| / \// \\ /0 0 \\_\_ / // | \ \ / / \/\_/ // | \ \ @\_^\_@'/ \/\_ // | \ \ //\_^\_/ \/\_ // | \ \ ( //) | \/// | \ \ ( / /) \_|\_ / ) // | \ \_\ ( // /) '/,\_ \_ \_/ ( ; -. | \_ \_\.-~ .-~~~^-. (( / / )) ,-{ \_ `-.|.-~-. .~ `. (( // / )) '/\ / ~-. \_ .-~ .-~^-. \ (( /// )) `. { } / \ \ (( / )) .----~-.\ \-' .~ \ `. \^-. ///.----..> \ \_ -~ `. ^-` ^-\_ ///-.\_ \_ \_ \_ \_ \_ \_}^ - - - - ~ ~-- ,.-~ /.-~ |

**Answer 8**:

The CVE ID for drupalgeddon2 was looked up and found to be CVE-2018-7600. The CVSS v3 information was looked up on the NIST NVD and following information was found -



So the vector string is AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H and the base score is 9.8.

As for the software and version having this vulnerability and being affected by it, the CVE description is enough -

Drupal before 7.58, 8.x before 8.3.9, 8.4.x before 8.4.6, and 8.5.x before 8.5.1 allows remote attackers to execute arbitrary code because of an issue affecting multiple subsystems with default or common module configurations.