🦠 Lab#09: Malware Containment and Eradication

🎯 Objective

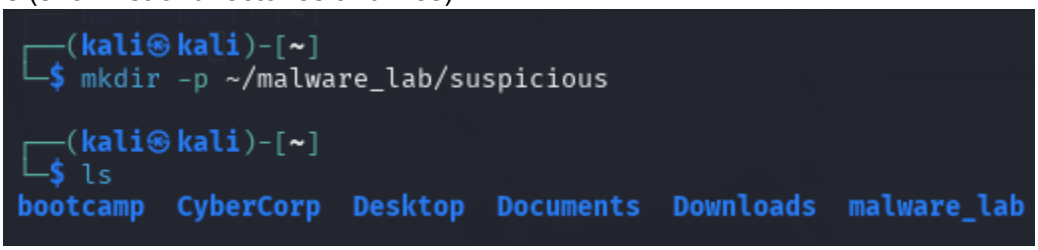
Simulate a detected malware file on a Linux system, isolate the threat, hash and preserve evidence, remove the file safely, and verify system integrity.

Step-by-Step Instructions / Summary

- **Step 1: Create a Simulated Malware File**
- **Step 2: Identify the Suspicious File**
- **Step 3: Isolate the File (Containment)**
- **Step 4: Hash the File (Preserve Evidence)**
- **Step 5: Compress & Archive File for Reporting**
- **Step 6: Remove the Malware Safely**
- **Step 7: Log Your Actions**
- **Step 8: Verify System Integrity (Basic)**

Step 1: Create a Simulated Malware File

Using this command to simulate a "malicious" file:
mkdir -p ~/malware_lab/suspicious
ls (show list of directories and files)



Creating the malicious file
echo "This file simulates a trojan downloader." > ~/malware_lab/suspicious/update_patch.bin

```
┌──(kali㉿kali)-[~]
└─$ echo "This file simulates a trojan downloader." > ~/malware_lab/suspicious/update_patch.bin

┌──(kali㉿kali)-[~]
└─$ cd malware_lab

┌──(kali㉿kali)-[~/malware_lab]
└─$ ls
suspicious
```

```
┌──(kali㉿kali)-[~/malware_lab/suspicious]
└─$ ls
update_patch.bin
```

Step 2: Identify the Suspicious File

Checking the type of file is created
file ~/malware_lab/suspicious/update_patch.bin

```
┌──(kali㉿kali)-[~]
└─$ file ~/malware_lab/suspicious/update_patch.bin
/home/kali/malware_lab/suspicious/update_patch.bin: ASCII text
```

Checks the file's metadata: permissions, creation data, and ownership

```
┌──(kali㉿kali)-[~]
└─$ stat ~/malware_lab/suspicious/update_patch.bin
  File: /home/kali/malware_lab/suspicious/update_patch.bin
  Size: 41              Blocks: 8          IO Block: 4096   regular file
Device: 8,1     Inode: 914180      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    kali)   Gid: ( 1000/    kali)
Access: 2025-07-29 22:06:52.495926255 -0400
Modify: 2025-07-29 22:02:34.849695445 -0400
Change: 2025-07-29 22:02:34.849695445 -0400
 Birth: 2025-07-29 22:02:34.849695445 -0400
```

Step 3: Isolate the File (Containment)

Make a directory to move the malicious file to isolate from other folders
sudo mkdir -p /quarantine

```
┌──(kali㉿kali)-[~]
└─$ sudo mkdir -p /quarantine

┌──(kali㉿kali)-[~]
└─$ ls /
```

```
proc  quarantine
```

Moving the malicious file to the quarantine directory
sudo mv ~/malware_lab/suspicious/update_patch.bin /quarantine/

```
┌──(kali㉿kali)-[~]
└─$ cd /quarantine

┌──(kali㉿kali)-[/quarantine]
└─$ ls
update_patch.bin
```

Step 4: Hash the File (Preserve Evidence)

Hashing the file to preserve the evidence:
This commands creates a file, the "tee" makes it so the user writes as root, the additional redirect ">" and "/dev/null" suppresses the duplicates output.

sudo sha256sum update_patch.bin | sudo tee hash_update_patch.txt > /dev/null

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo sha256sum update_patch.bin | sudo tee hash_update_patch.txt > /dev/null
```

Check if the hash file is created:
ls -l hash_update_patch.txt
cat hash_update_patch.txt

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo sha256sum update_patch.bin | sudo tee hash_update_patch.txt > /dev/null

┌──(kali㉿kali)-[/quarantine]
└─$ ls -l  hash_update_patch.txt
-rw-r--r-- 1 root root 83 Jul 29 22:27 hash_update_patch.txt

┌──(kali㉿kali)-[/quarantine]
└─$ cat hash_update_patch.txt
fe86b6b629c09b44c98e1e95626521abff2b39cd19644b0726f721aa2b8eda8a  update_patch.bin
```

Step 5: Compress & Archive File for Reporting

This command packages the malware sample and the hash together for reporting to a threat intel team or malware lab.
sudo tar -czvf update_patch_quarantined.tar.gz update_patch.bin hash_update_patch.txt

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo tar -czvf update_patch_quarantined.tar.gz update_patch.bin hash_update_patch.txt
update_patch.bin
hash_update_patch.txt

┌──(kali㉿kali)-[/quarantine]
└─$ ls
hash_update_patch.txt  update_patch.bin  update_patch_quarantined.tar.gz
```

Step 6: Remove the Malware Safely

WIth this command it overwrites and deletes the file, making recovery difficult. A safe malware removal option.

shred -u update_patch.bin

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo shred -u update_patch.bin

┌──(kali㉿kali)-[/quarantine]
└─$ ls
hash_update_patch.txt  update_patch_quarantined.tar
```

Step 7: Log Your Actions

Creating a log file of actions that happened
nano ~/malware_lab/response_log.txt

```
┌──(kali㉿kali)-[/quarantine]
└─$ nano ~/malware_lab/response_log.txt
```

```
File  Actions  Edit  View  Help
  GNU nano 8.3
- Suspicious file discovered: update_patch.bin
- Moved to /quarantine
- SHA-256 hash generated
- Archived file and hash
- Malware safely deleted using shred
```

```
┌──(kali㉿kali)-[/quarantine]
└─$ cat  ~/malware_lab/response_log.txt
- Suspicious file discovered: update_patch.bin
- Moved to /quarantine
- SHA-256 hash generated
- Archived file and hash
- Malware safely deleted using shred
```

Step 8: Verify System Integrity (Basic)

Scan home directory for malicious files:
find ~ -name "*.sh"

```
┌──(kali㉿kali)-[/quarantine]
└─$ find ~ -name "*.sh"

┌──(kali㉿kali)-[/quarantine]
└─$
```

Checks for any hidden files
ls -la ~ | grep "^\."

```
┌──(kali㉿kali)-[/quarantine]
└─$ ls -la ~ | grep "^\."

┌──(kali㉿kali)-[/quarantine]
└─$
```

Installing and running an Antivirus
sudo apt install clamav -y

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo apt install clamav -y
Installing:
  clamav

Installing dependencies:
  clamav-base  clamav-freshclam  libclamav12

Suggested packages:
  libclamunrar  clamav-doc  libclamunrar11

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 1291
  Download size: 15.1 MB
  Space needed: 69.5 MB / 3,737 MB available
```

sudo clamscan -r ~/

```
┌──(kali㉿kali)-[/quarantine]
└─$ sudo clamscan -r ~/
LibClamAV Error: cli_loaddbdir: No supported database files found in /var/lib/clamav
ERROR: Can't open file or directory

----------- SCAN SUMMARY -----------
Known viruses: 0
Engine version: 1.4.2
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 0.005 sec (0 m 0 s)
Start Date: 2025:07:29 23:00:15
End Date:   2025:07:29 23:00:15
```

🧰 Tools & Skills Used

Tools used:

- Kali Linux or Ubuntu VM
- Terminal access with sudo rights
- Tools used: sha256sum, file, stat, ls, rm, tar

Skills used

- Containment
- Evidence preservation (hashing, logging)
- File system analysis
- Safe malware removal

🧠 Reflection & Takeaways

This lab taught me the importance of containment and evidence preservation. The first step is not to delete the malware first, but to move it to a safe "quarantine" directory to stop the potential harm of the malware spreading and isolating from important files. I later use the process of hashing and logging. The process of hashing (sha256sum) is used to document the unique digital footprint to ensure its integrity of the file. Afterwards, the "shred -u” command is used which differs from the usual “rm” command by making recovery nearly impossible and emphasizes the principle of “secure eradication.” This concludes my lab.