

## LINUX FILE SYSTEM ANALYSIS

---

### AIM:

To explore and apply live forensic file system analysis techniques on a compromised Linux environment. This includes investigating users, system logs, binaries, permissions, and digital artefacts to reconstruct the attack timeline and identify evidence of compromise.

### PROCEDURE:

1. Isolate the compromised system and load clean binaries via USB for trusted analysis.
2. Modify the ``PATH`` and ``LD_LIBRARY_PATH`` to ensure only clean binaries are used.
3. Investigate suspicious uploads and artifacts under ``/var/www/html/``.
4. Extract metadata, timestamps, and file integrity using tools like ``stat``, ``exiftool``, and checksum utilities.
5. Identify and investigate unusual user accounts, group IDs, and ``sudoers`` entries.
6. Review user history and SSH configurations for backdoors.
7. Examine SUID binaries, unverified executables, and detect rootkits.

## TASK 1 – INTRODUCTION

- Introduced the importance of live file system forensic analysis in Linux environments.
- Emphasized the goal of identifying digital artefacts and compromise indicators.
- Clarified that remediation should not be done on live systems during initial analysis.
- Highlighted the focus on detecting unauthorized access, data tampering, and rootkits.
- Stressed the relevance of understanding logs, users, file structures, and permissions.
- Recommended restoring from backups after analysis, not reusing compromised systems.

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}

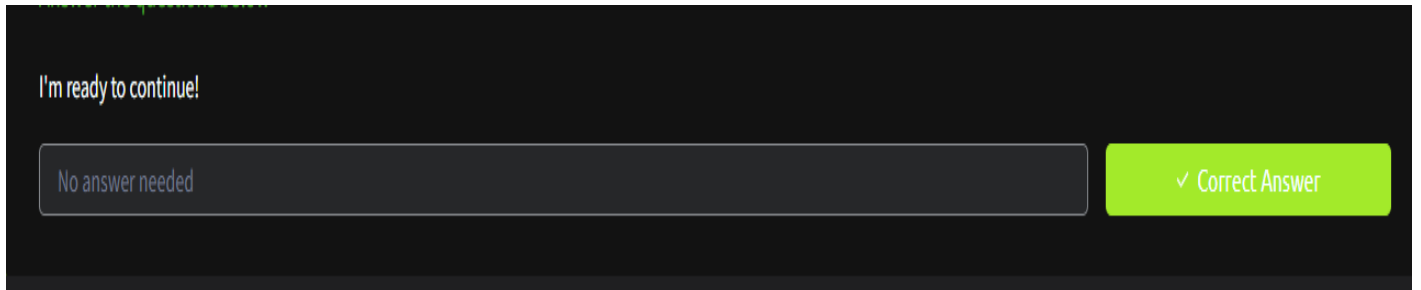
✓ Correct Answer

💡 Hint

## TASK 2 – INVESTIGATION SETUP

- Mounted a USB containing clean Debian-based binaries and libraries on the compromised system.
- Copied `/bin`, `/sbin`, `/lib`, and `/lib64` folders to `/mnt/usb` for a trusted toolset.
- Updated `PATH` and `LD_LIBRARY_PATH` to prioritize clean binaries for forensic commands.
- Ensured the environment uses only verified binaries to avoid tampered results.

- Verified clean environment setup using the check-env script.
- Provided a secure and controlled setup for conducting further forensic analysis.



### TASK 3 – FILES, PERMISSIONS & TIMESTAMPS

- Detected uploaded web shell `b2c8e1f5.phtml` via upload vulnerability.
- Found and analyzed reverse shell binary `reverse.elf`.
- Retrieved its metadata (MIME type), timestamps (`stat`), and computed hashes (MD5 & SHA256).
- Verified indicators via VirusTotal for malware classification.

- Practiced `find` command to identify files created by user `bob`.

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

#### TASK 4 – USERS AND GROUPS

- Used `/etc/passwd`, `getent`, and `cat /etc/group` to identify suspicious users.
- Discovered backdoor UID 0 user.
- Identified group with GID 46.
- Inspected `/etc/sudoers` file to find binaries accessible to Jane.

- Observed that Jane could use `/sbin/ifconfig` with `sudo`.

Answer the questions below

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

✓ Correct Answer

💡 Hint

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

✓ 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?

✓ Correct Answer

## TASK 5 – USER DIRECTORIES & SSH ACCESS

- Explored hidden files in home directories such as `.bash_history` and `.ssh/authorized_keys`.
- Found a backdoor SSH key in Jane's `authorized_keys`.
- Discovered flag in Jane's bash history.
- Located a hidden flag in Bob's home directory.

- Extracted modification timestamp for Jane's `~/.ssh/authorized_keys` using `stat`.

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 & EXECUTABLES

- Used `find` and `debsums` to identify unauthorized root-owned binaries and config file modifications.
- Used `md5sum` and `strings` for integrity and behavior analysis.
- Identified altered system config files.
- Found attacker-created binary in `/var/tmp/bash` with suspicious MD5 hash.

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

## TASK 7 – ROOTKIT DETECTION

- Ran ``chkrootkit`` and detected a suspicious ``.sh`` script.
- Used ``rkhunter`` to scan for deeper system integrity checks.
- Confirmed UID 0 account anomaly through rkhunter summary.

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

## RESULT:

Successfully identified indicators of compromise, backdoor accounts, and manipulated binaries. Demonstrated capability to use live forensics methodology in incident response and Linux system compromise investigations.