Tasks List For POC

📌 PoC Task List: Linux Security - Exploitation & Hardening

🔷 Task 1: User & Permission Misconfigurations

✅ **Setup:**

- Create multiple users (`useradd`, `passwd`).
- Assign incorrect permissions to sensitive files (`chmod 777 /etc/shadow`).

✅ **Exploit:**

- Demonstrate how a low-privileged user can access **sensitive system files** (e.g., `/etc/passwd`, `/etc/shadow`).

✅ **Mitigation:**

- Fix permission issues using `chmod`, `chown`.
- Use `sudo` privileges properly (`visudo`).

📌 **Deliverables:**

- Report: Commands used, screenshots.
- Video: Demonstrating access and fix.

🔷 Task 2: Remote Access & SSH Hardening

✅ **Setup:**

- Enable SSH on a Linux machine.
- Allow root login (`PermitRootLogin yes`) and password authentication.

✅ **Exploit:**

- Perform a **brute-force attack** on SSH using `hydra` or `medusa`.

✅ **Mitigation:**

- Disable root login (`PermitRootLogin no`).
- Enable key-based authentication.
- Configure `fail2ban` to prevent brute-force attempts.

📌 **Deliverables:**

- Report: Commands, attack analysis, defense steps.
- Video: Brute-force attack + mitigation.

🔷 Task 3: Firewall & Network Security

✅ **Setup:**

- Install & configure a basic web server (`apache2`).
- Allow all traffic (`ufw disable`).

✅ **Exploit:**

- Use `nmap` and `netcat` to scan for open ports & services.
- Show how an attacker can discover **exposed services**.

✅ **Mitigation:**

- Restrict access using `ufw` (only allow **SSH & HTTP**).
- Implement **iptables** rules to block unnecessary traffic.

📌 **Deliverables:**

- Report: Open ports before & after hardening.
- Video: Scanning & firewall setup.

🔷 Task 4: SUID & Privilege Escalation

✅ **Setup:**

- Set the SUID bit on `/bin/bash` (`chmod u+s /bin/bash`).
- Create a script running with **root privileges** (`chmod 4755 root_script.sh`).

✅ **Exploit:**

- Use `find / -perm -4000 2>/dev/null` to identify **SUID misconfigurations**.
- Escalate privileges to **root** using `/bin/bash -p`.

✅ **Mitigation:**

- Remove unnecessary SUID permissions (`chmod -s /bin/bash`).
- Restrict script execution to specific users.

📌 **Deliverables:**

- Report: Vulnerability description, privilege escalation demo, mitigation steps.
- Video: Exploiting SUID misconfig + fixing it.

🔷 Task 5: Automated Security Auditing & Scripting

✅ **Setup:**

- Write a **Bash script** that:
  - Checks user login attempts (`last`, `auth.log`).
  - Detects running services (`systemctl list-units --type=service`).
  - Monitors disk usage (`df -h`).

✅ **Exploit:**

- Run the script to identify weak configurations.
- Demonstrate **how attackers can exploit misconfigurations** (e.g., old user accounts with weak passwords).

✅ **Mitigation:**

- Automate **system monitoring** via `cron`.
- Implement **security alerts** (e.g., email notification when unauthorized SSH login attempts occur).

📌 **Deliverables:**

- Report: Bash script + execution results.
- Video: Script demo + security fixes.

🔷 Task 6: Log Analysis & Intrusion Detection

✅ **Setup:**

- Enable system logging (`journalctl`, `/var/log/auth.log`).
- Simulate multiple **failed SSH login attempts**.

✅ **Exploit:**

- Analyze logs (`grep "Failed password" /var/log/auth.log`).

- Identify **brute-force attempts** & unauthorized access.

✅ **Mitigation:**

- Implement **fail2ban** to block repeated failed attempts.
- Set up **log monitoring automation** (`logwatch`, `rsyslog`).

📌 **Deliverables:**

- Report: Log analysis + mitigation steps.
- Video: Failed login detection + response.

📣 Submission Requirements:

Each PoC task must be submitted with:

📌 **A detailed report** (Commands used, screenshots, explanations).