AYI NEDJIMI Consultants

Windows Server 2025 - Guide Pratique Cybersecurite

Catégorie : Forensics Lecture : 8 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

Méthodologie complète d Analyse Forensique des Logs IIS, DNS et AD DS sous. Expert en cybersécurité et intelligence artificielle. Guide technique...

🔍 Analyse Forensique des Logs IIS, DNS et AD DS sous Windows Server 2025

Méthodologie complète de triage d'infrastructure Windows Server 2025 : analyse forensique des logs IIS, DNS, Active Directory avec scripts PowerShell, corrélation multi-sources et détection avancée d'attaques. La reponse aux incidents et l'analyse forensique requierent une expertise technique pointue et une methodologie rigoureuse. Les equipes DFIR sont confrontees a des defis croissants : volumes de donnees massifs, techniques d'evasion complexees et environnements hybrides cloud. Cet article fournit un guide technique complet avec des procedures detaillees et des exemples concrets pour les professionnels de l'investigation numerique. Méthodologie complète d Analyse Forensique des Logs IIS, DNS et AD DS sous. Expert en cybersécurité et intelligence artificielle. Guide technique... Ce guide couvre les aspects essentiels de windows server 2025 forensics : méthodologie structurée, outils recommandés et retours d'expérience opérationnels. Les professionnels y trouveront des recommandations directement applicables.

Introduction

L'analyse forensique des infrastructures Windows Server 2025 représente un défi majeur pour les équipes de sécurité et les analystes forensiques. La complexité croissante des attaques ciblant les infrastructures Active Directory, couplée à la multiplication des vecteurs d'attaque via les services web et DNS, nécessite une approche méthodologique rigoureuse pour le triage et l'investigation.

Windows Server 2025 introduit de nouvelles fonctionnalités de journalisation et d'audit qui, bien exploitées, permettent une reconstruction précise de la chronologie des événements lors d'un incident de sécurité. L'objectif de cet article est de fournir une méthodologie complète pour l'analyse forensique des trois composants critiques : Internet Information Services (IIS), le service DNS Windows, et Active Directory Domain Services (AD DS).

Vos preuves numériques seraient-elles recevables devant un tribunal ?

1. Architecture de Journalisation dans Windows Server 2025

1.1 Vue d'Ensemble du Système de Journalisation

Windows Server 2025 implémente une architecture de journalisation multicouche basée sur Windows Event Log (WEL) et Event Tracing for Windows (ETW). Cette architecture permet une collecte granulaire des événements système avec plusieurs niveaux de verbosité configurables.

Le système de journalisation s'articule autour de plusieurs composants :

Event Log Service (eventlog) : Service central responsable de la gestion des journaux d'événements Windows. Il gère les canaux de journalisation, la rotation des logs, et l'accès concurrent aux fichiers EVTX.

Windows Event Collector (WEC) : Permet la centralisation des événements depuis plusieurs serveurs vers un collecteur central, utilisant le protocole WS-Management pour le transport sécurisé des événements.

Event Tracing for Windows (ETW) : Framework de traçage en temps réel permettant la capture d'événements haute fréquence avec un impact minimal sur les performances. ETW est particulièrement crucial pour l'analyse des activités IIS et DNS.

1.2 Formats de Fichiers et Structures de Données

Les journaux Windows Server 2025 utilisent principalement trois formats de fichiers :

Format EVTX : Format binaire propriétaire utilisé pour les Event Logs Windows. Structure basée sur des chunks de 64KB contenant des enregistrements XML compressés. Chaque enregistrement contient un header avec timestamp, EventID, et metadata, suivi du payload XML contenant les données de l'événement. Pour approfondir, consultez Modèles de Rapports.

Format W3C Extended Log : Utilisé par IIS pour les logs d'accès web. Format texte configurable permettant la sélection des champs à journaliser. Chaque ligne représente une requête HTTP avec des champs délimités par des espaces.

Format ETL : Format binaire pour les traces ETW. Contient des événements haute résolution avec timestamps précis au niveau microseconde. Nécessite des outils spécifiques comme WPA (Windows Performance Analyzer) ou tracerpt pour l'analyse.

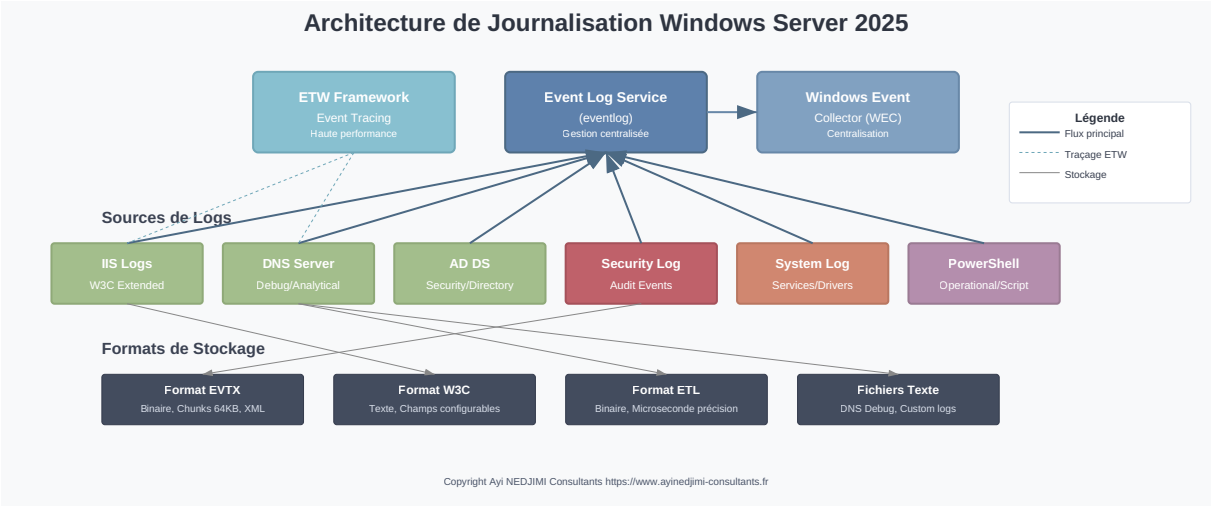

Illustration 1 : Architecture de Journalisation Windows Server 2025

1.3 Mécanismes de Rotation et Rétention

La gestion de la rétention des logs est critique pour l'analyse forensique. Windows Server 2025 implémente plusieurs mécanismes :

Rotation basée sur la taille : Les fichiers EVTX ont une taille maximale configurable (par défaut 20MB pour Security, 16MB pour System). Une fois atteinte, le système peut soit écraser les anciens événements (circular logging) soit archiver le fichier.

Rotation temporelle : IIS supporte la rotation quotidienne, hebdomadaire ou mensuelle des logs. Les fichiers sont nommés avec un pattern incluant la date (ex: u_ex20250315.log).

Archive automatique : Configuration possible via Group Policy pour l'archivage automatique des logs vers un partage réseau ou un système de stockage centralisé.

Artefact	Localisation	Information extraite
Registre	SYSTEM, SAM, SOFTWARE	Configuration, comptes, services
Event Logs	Security, System, Application	Connexions, erreurs, alertes
Prefetch	C:\Windows\Prefetch	Programmes executes et timestamps
MFT	$MFT sur volume NTFS	Fichiers crees, modifies, supprimes

Notre avis d'expert
La reconstruction de timeline est l'art le plus sous-estimé de la forensique numérique. Corréler les horodatages entre fichiers système, journaux d'événements, artefacts réseau et traces applicatives permet de reconstituer le scénario exact d'une compromission.

2. Analyse des Logs IIS

2.1 Emplacements et Configuration des Logs IIS

Les logs IIS dans Windows Server 2025 sont stockés par défaut dans :

```
%SystemDrive%\\inetpub\\logs\\LogFiles\\
```

Chaque site web possède son propre répertoire nommé W3SVC suivi de l'ID du site :

```
C:\\inetpub\\logs\\LogFiles\\W3SVC1\\   (Site par défaut)
C:\\inetpub\\logs\\LogFiles\\W3SVC2\\   (Second site)
```

2.2 Champs Critiques pour l'Analyse Forensique

Les champs suivants sont essentiels pour l'investigation : Pour approfondir, consultez OWASP Top 10 pour les LLM : Guide Remédiation 2026.

- **c-ip (Client IP)** : Adresse IP source de la requête. Attention aux proxys et load balancers qui peuvent masquer l'IP réelle. Utiliser le champ X-Forwarded-For si disponible.
- **cs-username** : Identifiant de l'utilisateur authentifié. Vide pour les accès anonymes. Critical pour tracer les actions d'un compte compromis.
- **cs-method et cs-uri-stem** : Méthode HTTP et chemin de la ressource. Permet d'identifier les tentatives d'exploitation (SQL injection, path traversal).
- **sc-status et sc-win32-status** : Codes de retour HTTP et Windows. Les codes 4xx indiquent des erreurs client (401: non autorisé, 403: interdit, 404: non trouvé). Les codes 5xx signalent des erreurs serveur potentiellement liées à des attaques.
- **time-taken** : Temps de traitement en millisecondes. Les requêtes anormalement longues peuvent indiquer des attaques par déni de service ou des tentatives d'exploitation.

2.3 Techniques d'Analyse Avancées

2.4 Détection d'Activités Malveillantes

Identification de Web Shells : Les web shells génèrent des patterns caractéristiques dans les logs IIS :

- Requêtes POST répétées vers des fichiers ASPX/PHP/JSP nouvellement créés
- User-Agent inhabituels ou absents
- Longues chaînes en base64 dans les paramètres
- Accès depuis des IPs internes après compromission initiale

Analyse des tentatives d'exploitation Exchange :

```
# ProxyLogon/ProxyShell detection
$exploitPatterns = @{
    ProxyLogon = "/owa/auth/x.js"
    ProxyShell = "/autodiscover/autodiscover.json"
    ProxyNotShell = "/autodiscover/autodiscover.json.*@.*Powershell"
}

$exploitAttempts = foreach($pattern in $exploitPatterns.GetEnumerator()) {
    $logs | Where-Object { $_."cs-uri-stem" -match $pattern.Value } |
        Select-Object @{N="ExploitType";E={$pattern.Key}}, DateTime, "c-ip", "cs-uri-
stem", "sc-status"
}
```

3. Analyse des Logs DNS

3.1 Configuration et Emplacements des Logs DNS

Windows Server 2025 offre plusieurs niveaux de journalisation DNS :

Debug Logging : Journalisation détaillée des requêtes et réponses DNS

```
Emplacement par défaut : %SystemRoot%\\System32\\DNS\\DNS.log
Format : Texte avec champs délimités par espaces
Taille maximale : Configurable (500MB par défaut)
```

Analytical Logging (ETW) : Journalisation haute performance via ETW

```
Canal : Microsoft-Windows-DNSServer/Analytical
Format : EVTX
Emplacement : %SystemRoot%\\System32\\Winevt\\Logs\\Microsoft-Windows-
DNSServer%4Analytical.evtx
```

📌 **Event IDs Critiques DNS :**

- **150** : Zone transfer initiated
- **151** : Zone transfer completed
- **152** : Zone transfer failed
- **541** : DNS query received
- **545** : DNS response sent

3.2 Analyse des Requêtes DNS Suspectes

3.3 Corrélation avec les Événements de Zone Transfer

Les transferts de zone non autorisés représentent un risque majeur :

```
# Analyse des événements de zone transfer
$zoneTransfers = Get-WinEvent -FilterHashtable @{
    LogName = 'DNS Server'
    ID = @(6001, 6002, 6003)  # Zone transfer events
} | ForEach-Object {
    $xml = [xml]$_.ToXml()
    [PSCustomObject]@{
        TimeCreated = $_.TimeCreated
        EventID = $_.Id
        ZoneName = $xml.Event.EventData.Data[0].'#text'
        RequestingServer = $xml.Event.EventData.Data[1].'#text'
        TransferType = $xml.Event.EventData.Data[2].'#text'
        Result = $xml.Event.EventData.Data[3].'#text'
    }
}

# Vérification des serveurs autorisés
$authorizedServers = @("192.168.1.10", "192.168.1.11")
$unauthorizedTransfers = $zoneTransfers | Where-Object {
    $_.RequestingServer -notin $authorizedServers
}
```

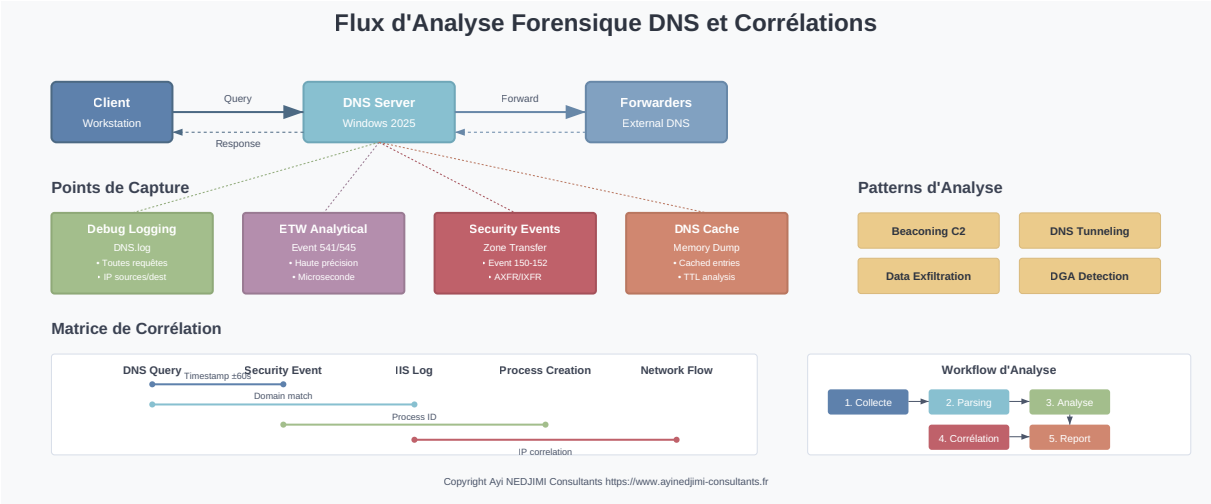

Illustration 2 : Flux d'Analyse Forensique DNS et Matrice de Corrélation

Cas concret

L'analyse forensique de NotPetya (2017) a révélé que le malware utilisait le mécanisme de mise à jour du logiciel comptable ukrainien M.E.Doc comme vecteur de distribution initiale. La reconstruction de la timeline d'infection a montré que la propagation mondiale s'était faite en moins de 45 minutes via EternalBlue.

Disposez-vous d'un kit de forensique prêt à l'emploi en cas de compromission ?

4. Analyse des Logs Active Directory Domain Services

4.1 Sources de Logs AD DS

Active Directory génère des événements dans plusieurs canaux :

Security Log : Événements d'authentification et d'autorisation Pour approfondir, consultez Anti-Forensics.

```
Emplacement : %SystemRoot%\\System32\\Winevt\\Logs\\Security.evtx
Taille par défaut : 128MB (augmenter à 4GB minimum pour forensics)
Events critiques : 4624-4625 (Logon), 4720-4733 (Gestion comptes), 4768-4771 (Kerberos)
```

Directory Service Log : Opérations LDAP et réplication

```
Emplacement : %SystemRoot%\\System32\\Winevt\\Logs\\Directory Service.evtx
Events critiques : 1644 (LDAP searches), 2889 (LDAP signing), 1102 (Audit log cleared)
```

AD Database (NTDS.dit) : Base de données Active Directory

```
Emplacement : %SystemRoot%\\NTDS\\ntds.dit
Transaction logs : %SystemRoot%\\NTDS\\edb*.log
Checkpoint : %SystemRoot%\\NTDS\\edb.chk
```

4.2 Événements Critiques pour l'Investigation

4.3 Analyse de la Réplication et DCSync

⚠️ Détection d'Attaques DCSync

Les attaques DCSync permettent l'extraction des hashes de mots de passe en simulant un contrôleur de domaine. Elles exploitent les droits de réplication `ms-DS-Replication-Get-Changes` et `ms-DS-Replication-Get-Changes-All`.

```
# Events de réplication suspects
$dcSyncEvents = Get-WinEvent -FilterHashtable @{
    LogName = 'Security'
    ID = @(4662, 4624)  # DS Access et Logon events
} | Where-Object {
    $_.Message -match "ms-DS-Replication-Get-Changes" -or
    $_.Message -match "ms-DS-Replication-Get-Changes-All" -or
    $_.Message -match "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2" -or
    $_.Message -match "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2"
}

# Analyse détaillée des droits de réplication
$replicationRights = Get-ADObject -Filter * -Properties nTSecurityDescriptor |
    ForEach-Object {
        $acl = $_.nTSecurityDescriptor
        $aces = $acl.Access | Where-Object {
            $_.ActiveDirectoryRights -match "ExtendedRight" -and
            ($_.ObjectType -eq "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2" -or
             $_.ObjectType -eq "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2")
        }
        if ($aces) {
            [PSCustomObject]@{
                ObjectDN = $_.DistinguishedName
                Permissions = $aces | ForEach-Object {
                    "$($_.IdentityReference) - $($_.ActiveDirectoryRights)"
                }
            }
        }
    }
```

4.4 Timeline Reconstruction

La reconstruction de timeline est cruciale pour comprendre la progression d'une attaque. Voici un framework complet d'agrégation multi-sources :

```
# Agrégation multi-sources pour timeline
$timeline = @()

# Events AD
$adEvents = Get-WinEvent -FilterHashtable @{
    LogName = @('Security', 'System', 'Application', 'Directory Service')
    StartTime = (Get-Date).AddDays(-7)
} | Select-Object TimeCreated, LogName, Id, Message

# Logs IIS
$iisLogs = Get-ChildItem "C:\\inetpub\\logs\\LogFiles\\W3SVC*\\*.log" |
    Where-Object { $_.LastWriteTime -gt (Get-Date).AddDays(-7) } |
    ForEach-Object {
        Get-Content $_.FullName | Where-Object { $_ -notmatch "^#" } |
        ConvertFrom-Csv -Delimiter " " -Header @("date","time","s-ip","cs-method","cs-uri-
stem","cs-uri-query","s-port","cs-username","c-ip","cs-User-Agent","cs-Referer","sc-
status","sc-substatus","sc-win32-status","time-taken") |
        ForEach-Object {
            [PSCustomObject]@{
                TimeCreated = [DateTime]::Parse($_.date + " " + $_.time)
                Source = "IIS"
                Details = "$($_.c-ip) - $($_.cs-method) $($_.cs-uri-stem) - $($_.sc-
status)"
            }
        }
    }

# Logs DNS
$dnsLogs = Get-Content "C:\\Windows\\System32\\DNS\\DNS.log" -ErrorAction SilentlyContinue
|
    Where-Object { $_ -match "^\\d{1,2}/\\d{1,2}/\\d{4}" } |
    ForEach-Object {
        if ($_ -match "^(\\S+)\\s+(\\S+).+\\[(\\S+)\\].+Query.+for\\s+(.+)") {
            [PSCustomObject]@{
                TimeCreated = [DateTime]::Parse($matches[1] + " " + $matches[2])
                Source = "DNS"
                Details = "$($matches[3]) queried $($matches[4])"
            }
        }
    }

# Consolidation et tri chronologique
$timeline = $adEvents + $iisLogs + $dnsLogs |
    Sort-Object TimeCreated |
    Select-Object TimeCreated, Source, @{N="Event";E={
        if ($_.Id) { "EventID: $($_.Id)" }
        elseif ($_.Details) { $_.Details }
        else { $_.Message -replace "\`r\`n", " " | ForEach-Object { $_.Substring(0,
[Math]::Min($_.Length, 100)) } }
    }}

# Export pour analyse
$timeline | Export-Csv -Path "C:\\Forensics\\Timeline_$(Get-Date -Format
'yyyyMMdd_HHmmss').csv" -NoTypeInformation
```

5. Techniques de Corrélation Avancées

5.1 Corrélation Multi-Sources

La corrélation entre différentes sources de logs permet d'identifier des patterns d'attaque complexes. Voici une fonction avancée de corrélation temporelle :

```
# Fonction de corrélation temporelle
function Find-CorrelatedEvents {
    param(
        [DateTime]$ReferenceTime,
        [Int]$WindowSeconds = 60,
        [String[]]$LogNames = @('Security', 'System', 'Application')
    )

    $startTime = $ReferenceTime.AddSeconds(-$WindowSeconds)
    $endTime = $ReferenceTime.AddSeconds($WindowSeconds)

    $correlatedEvents = @{}

    # Windows Events
    $correlatedEvents['WindowsEvents'] = Get-WinEvent -FilterHashtable @{
        LogName = $LogNames
        StartTime = $startTime
        EndTime = $endTime
    } -ErrorAction SilentlyContinue

    # IIS Logs
    $iisPath = "C:\\inetpub\\logs\\LogFiles\\W3SVC1\\*.log"
    $correlatedEvents['IISLogs'] = Get-Content $iisPath -ErrorAction SilentlyContinue |
        Where-Object {
            $_ -notmatch "^#" -and
            $_ -match "^(\\S+)\\s+(\\S+)"
        } | ForEach-Object {
            $fields = $_ -split '\\s+'
            $logTime = [DateTime]::Parse($fields[0] + " " + $fields[1])
            if ($logTime -ge $startTime -and $logTime -le $endTime) {
                $_
            }
        }

    # DNS Logs
    $dnsPath = "C:\\Windows\\System32\\DNS\\DNS.log"
    $correlatedEvents['DNSLogs'] = Get-Content $dnsPath -ErrorAction SilentlyContinue |
        Where-Object {
            $_ -match "^(\\d{1,2}/\\d{1,2}/\\d{4})\\s+(\\S+)"
        } | ForEach-Object {
            if ($_ -match "^(\\S+)\\s+(\\S+)") {
                $logTime = [DateTime]::Parse($matches[1] + " " + $matches[2])
                if ($logTime -ge $startTime -and $logTime -le $endTime) {
                    $_
                }
            }
        }

    return $correlatedEvents
}

# Exemple d'utilisation pour investigation d'incident
$suspiciousLogin = Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4624} |
    Where-Object { $_.Message -match "NTLM" } |
    Select-Object -First 1

$correlated = Find-CorrelatedEvents -ReferenceTime $suspiciousLogin.TimeCreated
-WindowSeconds 300
```

5.2 Analyse Comportementale et Détection d'Anomalies

📊 Framework de Détection d'Anomalies

```
function Detect-AnomalousActivity {
    param(
        [String]$BaselinePath = "C:\\Forensics\\Baseline.json",
        [Int]$ThresholdMultiplier = 3
    )

    # Chargement de la baseline
    if (Test-Path $BaselinePath) {
        $baseline = Get-Content $BaselinePath | ConvertFrom-Json
    } else {
        # Création de baseline
        $baseline = @{
            AverageLoginsPerHour = 50
            CommonServiceAccounts = @('svc_backup', 'svc_sql', 'svc_web')
            NormalDNSQueryRate = 1000
            CommonWebPaths = @('/default.aspx', '/api/health', '/login')
        }
    }

    # Analyse des activités courantes
    $currentHour = (Get-Date).AddHours(-1)

    # Analyse des logons
    $recentLogons = Get-WinEvent -FilterHashtable @{
        LogName = 'Security'
        ID = 4624
        StartTime = $currentHour
    } | Measure-Object

    if ($recentLogons.Count -gt ($baseline.AverageLoginsPerHour * $ThresholdMultiplier)) {
        Write-Warning "Anomalie détectée : Nombre de logons anormal ($
($recentLogons.Count) vs baseline $($baseline.AverageLoginsPerHour))"
    }

    # Analyse des comptes de service
    $serviceAccountLogons = Get-WinEvent -FilterHashtable @{
        LogName = 'Security'
        ID = 4624
        StartTime = $currentHour
    } | Where-Object {
        $xml = [xml]$_.ToXml()
        $targetUser = $xml.Event.EventData.Data | Where-Object {$_.Name -eq
'TargetUserName'} | Select-Object -ExpandProperty '#text'
        $targetUser -like "svc_*" -and $targetUser -notin $baseline.CommonServiceAccounts
    }

    if ($serviceAccountLogons) {
        Write-Warning "Anomalie détectée : Utilisation de compte de service non référencé"
        $serviceAccountLogons | ForEach-Object {
            $xml = [xml]$_.ToXml()
            $targetUser = $xml.Event.EventData.Data | Where-Object {$_.Name -eq
'TargetUserName'} | Select-Object -ExpandProperty '#text'
            Write-Warning "  - Compte suspect : $targetUser"
        }
    }
}
```

5.3 Reconstruction de Chaîne d'Attaque (Kill Chain)

L'analyse selon le modèle Cyber Kill Chain permet de reconstituer les phases d'une attaque complexee. Le script suivant implémente une détection automatique des 7 phases :

6. Outils et Automatisation

6.1 Script Principal d'Extraction Forensique

6.2 Analyse Automatisée avec SIGMA Rules

L'implémentation de règles SIGMA permet une détection standardisée des patterns d'attaque connus. Voici un framework d'intégration SIGMA pour Windows : Pour approfondir, consultez Comparatif Outils DFIR.

```
# Implémentation de SIGMA rules pour Windows
function Test-SigmaRule {
    param(
        [String]$RulePath,
        [String]$EventLogPath
    )

    # Parsing de la règle SIGMA (format YAML simplifié)
    $rule = Get-Content $RulePath -Raw | ConvertFrom-Yaml

    $detection = $rule.detection
    $selection = $detection.selection

    # Construction de la requête
    $filter = @{
        Path = $EventLogPath
    }

    if ($selection.EventID) {
        $filter['ID'] = $selection.EventID
    }

    # Recherche des événements correspondants
    $matches = Get-WinEvent -FilterHashtable $filter -ErrorAction SilentlyContinue |
Where-Object {
        $xml = [xml]$_.ToXml()
        $match = $true

        foreach ($criterion in $selection.Keys) {
            if ($criterion -ne 'EventID') {
                $value = $xml.Event.EventData.Data |
                    Where-Object {$_.Name -eq $criterion} |
                    Select-Object -ExpandProperty '#text'

                if ($value -notmatch $selection[$criterion]) {
                    $match = $false
                    break
                }
            }
        }
        $match
    }

    return [PSCustomObject]@{
        Rule = $rule.title
        Severity = $rule.level
        MatchCount = $matches.Count
        Matches = $matches
    }
}
```

7. Recommandations et Bonnes Pratiques

7.1 Configuration Préventive

Pour optimiser la capacité d'investigation forensique, les configurations suivantes sont recommandées :

7.2 Centralisation et Archivage

La mise en place d'une architecture de centralisation des logs est cruciale pour les grandes infrastructures. Windows Event Forwarding (WEF) permet la collecte centralisée sans infrastructure SIEM lourde :

```
# Configuration Windows Event Forwarding (WEF)
wecutil cs ForensicsSubscription.xml
```

7.3 Documentation et Rapport

La documentation rigoureuse est essentielle pour la validité légale de l'investigation. Un framework de génération de rapport forensique complet doit inclure :

- **Informations du cas** : Nom, numéro, examinateur, date, système analysé
- **Chain of Custody** : Hashes SHA-256 de tous les fichiers d'évidence
- **Findings Summary** : Résultats critiques, activités suspectes, timeline consolidée, IOCs
- **Recommandations** : Mesures correctives et préventives
- **Annexes techniques** : Logs complets, scripts utilisés, méthodologie détaillée

> **Ressources open source associées :**
>
> - SuperTimelineBuilder — Générateur de super timeline (C++)
> - ETWThreatHunter — Threat hunter ETW (C++)
> - forensics-windows-fr — Dataset forensics Windows (HuggingFace)

Questions frequentes

Comment mener une investigation forensique sur un systeme compromis ?

Une investigation forensique debute par la preservation des preuves via une image disque et un dump memoire, suivie de l'analyse des artefacts systeme (registres, journaux d'evenements, fichiers prefetch), la reconstruction de la timeline d'activite et la correlation des indicateurs de compromission pour identifier la source et l'etendue de l'attaque.

Quels sont les outils essentiels pour l'analyse forensique ?

Les outils essentiels pour l'analyse forensique incluent Volatility pour l'analyse memoire, Autopsy et FTK pour l'analyse disque, KAPE et Velociraptor pour la collecte automatisee, Plaso pour la creation de timelines, ainsi que des outils de triage comme Eric Zimmerman's tools pour l'analyse des artefacts Windows.

Pourquoi la chaine de custody est-elle importante en forensique ?

La chaine de custody garantit l'integrite et l'admissibilite des preuves numeriques en documentant chaque etape de manipulation, de la collecte a la presentation. Sans une chaine de custody rigoureuse, les preuves peuvent etre contestees juridiquement et perdre leur valeur probante.

Pour approfondir, consultez les ressources officielles : SANS White Papers, NVD - NIST et ANSSI.

Sources et références : SANS SIFT · MITRE ATT&CK

Conclusion

L'analyse forensique des infrastructures Windows Server 2025 requiert une approche méthodologique rigoureuse combinant expertise technique approfondie et utilisation d'outils spécialisés. La corrélation entre les différentes sources de logs (IIS, DNS, AD DS) permet de reconstituer avec précision la chronologie des événements lors d'un incident de sécurité.

📌 Points Clés à Retenir :

1. **Configuration préventive** : Une configuration adéquate de l'audit et de la rétention des logs est essentielle pour disposer des données nécessaires lors d'une investigation.
2. **Approche multi-sources** : La corrélation entre IIS, DNS et AD DS révèle des patterns d'attaque invisibles lors de l'analyse isolée de chaque source.
3. **Automatisation** : L'utilisation de scripts PowerShell et de règles SIGMA permet d'automatiser la détection et d'accélérer le triage lors d'incidents majeurs.
4. **Documentation rigoureuse** : La traçabilité et la documentation détaillée sont cruciales pour la validité légale et la reproductibilité de l'investigation.
5. **Évolution continue** : Les techniques d'attaque évoluant constamment, la veille technologique et l'adaptation des méthodologies d'investigation sont indispensables.

Cette méthodologie, appliquée de manière systématique, permet aux équipes de sécurité de répondre efficacement aux incidents, d'identifier les vecteurs de compromission, et de mettre en place les mesures correctives appropriées pour renforcer la posture de sécurité de l'infrastructure.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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