M4 - P1: Enumeración de Active Directory con BloodHound y SharpHound

1. Instalación software

Máquina Windows: https://archive.org/details/es-es_windows_server_2019_x64_dvd

- **Sistema Operativo**: Windows Server 2019: *es_windows_server_2019_x64_dvd_3ce0fd9e.iso*
- **Arquitectura**: x64
- **Idioma**: Español (es-es)
- **Edición incluida**: Standard y Datacenter (en la instalación eliges)
- **Formato**: ISO "RTM" (Release to Manufacturing)
- **Build base**: **17763.1** (Octubre 2018 - *sin parches aplicados*)

Esta versión es completamente **válida** para usar como **Controlador de Dominio vulnerable** en entornos de laboratorio, ya que es compatible con:

- BloodHound / SharpHound
- Ataques de Kerberoasting
- Delegación (constrained/unconstrained)
- Errores de configuración comunes en AD
- Prácticas de enumeración, escalada y post-explotación

Evitar actualizar Windows más allá de esa versión para mantener las vulnerabilidades simulables.

Recomendaciones inmediatas

1. **No conectes el servidor a Internet permanentemente**
   - Desactiva Windows Update después de instalar (te paso cómo más abajo).
2. **Crea el entorno**:
   - IP fija: `10.0.0.190`
   - Dominio: `corp.local`
   - Usuarios: `admin1`, `usuario1`, `soporte1`
   - Cliente Windows 10 unido al dominio
3. **Después de la instalación**:
   - Crea un snapshot de la VM limpia
   - Instala herramientas administrativas (RSAT si necesitas)
   - Añade roles de AD DS y DNS

Desactivar actualizaciones automáticas

Abre PowerShell como administrador:

```
sconfig
```

Selecciona la opción:

```
5) Windows Update Settings
```

Elige:

```
Manual
```

1.1. Configuración Windows Server 2019 (DC)

Pasos completos y detallados para configurar Windows Server 2019 como un Controlador de Dominio (DC), usando el dominio `corp.local` y una IP estática `10.0.0.190`.

Este entorno será compatible con herramientas como **BloodHound**, **SharpHound**, y prácticas de post-explotación en entornos Active Directory.

Elemento	Valor
Sistema Operativo	Windows Server 2019 (VM)
Nombre del servidor	`DC01`
IP estática	`10.0.0.190`
Dominio	`corp.local`
Rol	Active Directory Domain Services + DNS
DNS local	10.0.0.190 (el propio DC)
Usuarios de prueba	`admin1`, `usuario1`, `soporte1`
Grupo de prueba	`SoporteTI`

1.1.1. Configurar IP Estática

Cambiar IP desde GUI:

1. Abrir **Panel de Control** > **Centro de redes y recursos compartidos**
2. Click en el adaptador de red > Propiedades
3. Seleccionar `Protocolo de Internet versión 4 (TCP/IPv4)` > Propiedades
4. Usar esta configuración:

Campo	Valor
Dirección IP	`10.0.0.190`
Máscara de subred	`255.255.255.0`
Puerta de enlace	`10.0.0.1`
DNS preferido	`10.0.0.190`

No poner DNS externo como `8.8.8.8` en esta máquina, ya que será tu propio DNS al instalar el rol de AD.

1.1.2. Cambiar el nombre del servidor

Desde PowerShell:

```
Rename-Computer -NewName "DC01" -Restart
```

O desde GUI: Panel de control > Sistema > Cambiar nombre del equipo.

1.1.3. Instalar el Rol de Active Directory

1. Abrir **Server Manager**
2. Click en **"Add roles and features"**
3. Avanzar hasta la sección **"Server Roles"**
4. Seleccionar:
   - `Active Directory Domain Services`
   - Se seleccionará automáticamente el rol de DNS. Aceptar.
5. Avanzar y hacer clic en **Install**
6. Esperar a que finalice (no reiniciar todavía)

1.1.4. Promover a Controlador de Dominio

1. En Server Manager, aparecerá un aviso:
   - `Promote this server to a domain controller` → Haz clic
2. Elegir:
   - `Add a new forest`
   - Nombre del dominio raíz: **`corp.local`**
3. Configurar:
   - Contraseña del modo de recuperación (DSRM): `P@ssw0rd123` *(guárdarla bien)*
4. Avanzar hasta el final y dar a **Install**
5. Se reiniciará automáticamente

1.1.5. Verifica que el dominio esté activo

Después del reinicio, ya se puede iniciar sesión como:

```
corp\Administrator
```

Verificar DNS:

```
nslookup corp.local
```

Debería resolver a `10.0.0.190`

1.1.6. Crear Usuarios y Grupos de Prueba

Abrir **"Active Directory Users and Computers" (ADUC)** desde Server Manager > Tools.

Crear Grupo:

1. Botón derecho sobre **Users** > New > Group
2. Nombre: `SoporteTI`
3. Grupo global, tipo seguridad

Crear Usuarios:

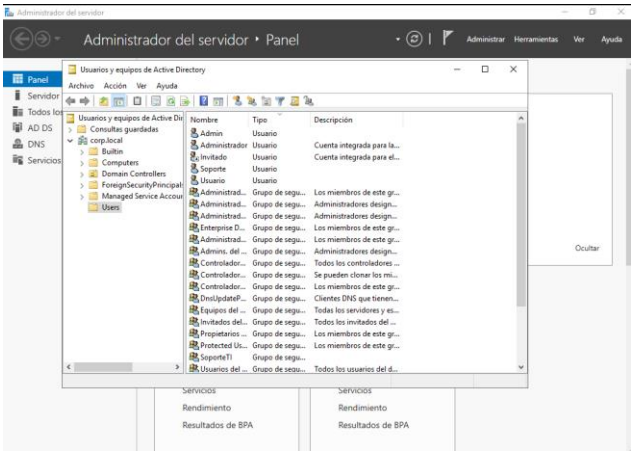
Creae los siguientes usuarios (clic derecho en `Users` > New > User):

Nombre	Usuario	Contraseña	Grupos
Admin	`admin1`	`P@ssw0rd123`	Domain Admins
Usuario	`usuario1`	`P@ssw0rd123`	Domain Users
Soporte	`soporte1`	`P@ssw0rd123`	Domain Users, SoporteTI

En todos los casos, desactivar la opción de "User must change password at next logon" y activar "Password never expires" (para simplificar pruebas).

Agregar `admin1` al grupo Domain Admins:

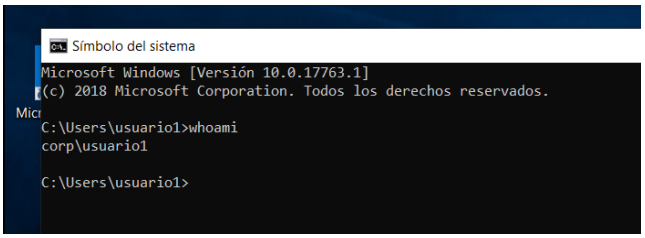
1. Abrir propiedades del usuario
2. Tab **Member Of** > Agregar `Domain Admins`



1.1.7. Cliente Windows 10 (opcional pero recomendado)

En tu VM con Windows 10:

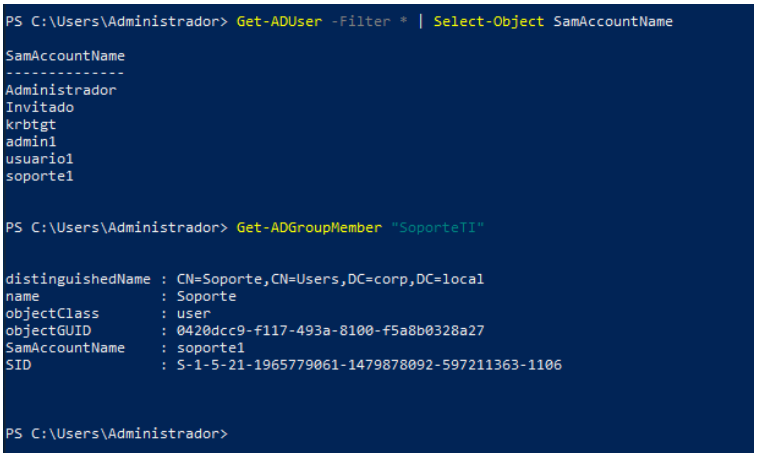
1. Asignar IP: `10.0.0.191`
2. DNS primario: `10.0.0.190`
3. Unir al dominio:
   - Click derecho en "Este equipo" > Propiedades > Cambiar configuración > Cambiar
   - Seleccionar: `Dominio` → `corp.local`
   - Usuario: `admin1`, pass: `P@ssw0rd123`
4. Reiniciar
5. Iniciar sesión como: `corp\usuario1`



1.1.8. Validaciones Finales

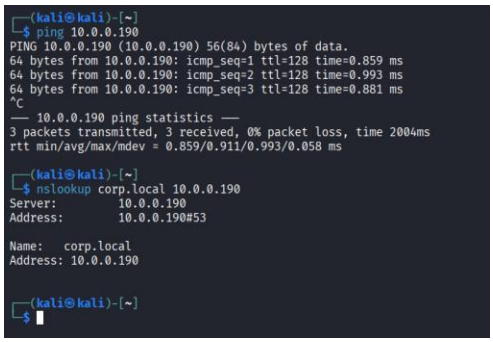
Desde el DC (PowerShell):

```
Get-ADUser -Filter * | Select-Object SamAccountName
Get-ADGroupMember "SoporteTI"
```



Desde Kali:

```
ping 10.0.0.190
nslookup corp.local 10.0.0.190
```



1.1.8. Snapshot recomendado

Una vez terminado todo esto, **crear un snapshot de la VM**. Así podrás regresar a este punto base antes de cada práctica con BloodHound, Mimikatz, o pruebas más invasivas.

1.2.¿Qué sigue?

Con este entorno ya se puede:

- Iniciar sesión como `usuario1` (bajo privilegios) en la máquina cliente
- Ejecutar **SharpHound** desde PowerShell
- Importar datos en **BloodHound** desde Kali
- Analizar relaciones de privilegios, delegación, SPNs, etc.

Elemento	Estado esperado
Controlador de Dominio (DC) `corp.local` con usuarios creados	
Kali Linux (Atacante externo) IP: `192.168.56.110` + BloodHound	
Windows 10 unido al dominio Usuario: `usuario1` (permisos limitados)	

2. Recolección con SharpHound

Este paso es fundamental en la enumeración de Active Directory usando **BloodHound**, ya que **SharpHound.exe** es el componente que recoge toda la información de relaciones de privilegios, grupos, ACLs y sesiones en el dominio.

2.1. ¿Desde dónde se ejecuta?

Desde un equipo **unido al dominio**, donde se tenga acceso **como un usuario estándar** (por ejemplo, `usuario1`).

NO se necesita ser administrador para hacer una recolección básica. SharpHound está diseñado para recolectar información que cualquier usuario autenticado del dominio puede ver.

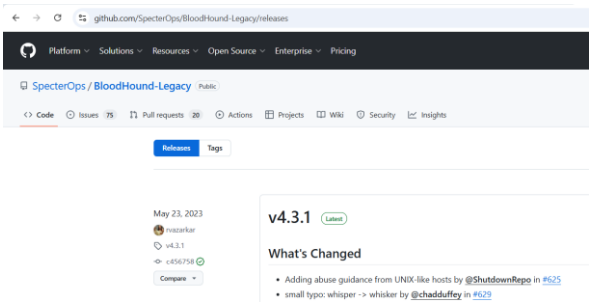
2.2. ¿Qué necesitas?

En el equipo Windows 10 que es cliente del dominio:

- `SharpHound.exe` (última versión del recolector)
- Sesión iniciada como `CORP\usuario1` (o cualquier usuario válido del dominio)
- PowerShell
- Permisos para leer dentro del dominio (con eso es suficiente)

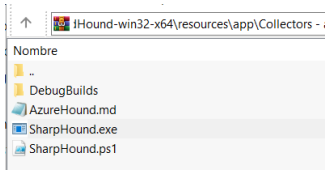
2.3. Descarga de SharpHound

1. Ir a GitHub oficial de BloodHound:
   https://github.com/BloodHoundAD/BloodHound/releases



2. Descargar el archivo ZIP de "Collectors":

   `SharpHound-vX.X.X.zip`



3. Extraer `SharpHound.exe` y copiarlo en una ruta accesible en el Windows cliente, por ejemplo:

   `C:\Users\Public\SharpHound.exe`

2.4. Ejecutar SharpHound desde PowerShell

Abrir PowerShell en el cliente Windows y ejecutar:

```
cd C:\Users\Public
.\SharpHound.exe -c all
```

```
PS C:\Users> cd .\Public\
PS C:\Users\Public> .\SharpHound.exe -c all
2025-09-25T23:49:33.7659810+02:00|INFORMATION|This version of SharpHound is compatible with the 4.3.1 Release of BloodH
und
2025-09-25T23:49:34.2397007+02:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, L
ggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2025-09-25T23:49:34.2730533+02:00|INFORMATION|Initializing SharpHound at 23:49 on 25/09/2025
2025-09-25T23:49:35.2977794+02:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for corp.local : DC01
corp.local
2025-09-25T23:49:35.5748415+02:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL,
ontainer, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2025-09-25T23:49:36.0488515+02:00|INFORMATION|Beginning LDAP search for corp.local
2025-09-25T23:49:36.0939447+02:00|INFORMATION|Producer has finished, closing LDAP channel
2025-09-25T23:49:36.1009180+02:00|INFORMATION|LDAP channel closed, waiting for consumers
2025-09-25T23:50:06.8223731+02:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 32 MB RAM
2025-09-25T23:50:26.2763048+02:00|INFORMATION|Consumers finished, closing output channel
2025-09-25T23:50:26.4228730+02:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2025-09-25T23:50:27.4348633+02:00|INFORMATION|Status: 95 objects finished (+95 1.862745)/s -- Using 40 MB RAM
2025-09-25T23:50:27.4359213+02:00|INFORMATION|Enumeration finished in 00:00:51.4101451
2025-09-25T23:50:27.6851584+02:00|INFORMATION|Saving cache with stats: 54 ID to type mappings.
 55 name to SID mappings.
 1 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2025-09-25T23:50:27.7348094+02:00|INFORMATION|SharpHound Enumeration Completed at 23:50 on 25/09/2025! Happy Graphing!
PS C:\Users\Public>
```

Este parámetro lanza **todos los métodos de recopilación**, incluyendo:

Método	Descripción breve
`ACL`	Permisos sobre objetos de AD (control total, escritura, etc.)
`Session`	Usuarios conectados a equipos
`Group Membership`	Miembros de grupos locales y globales
`LocalAdmin`	Quién es administrador local de las máquinas
`Trusts`	Relaciones de confianza entre dominios (si las hay)
`LoggedOn`	Sesiones activas detectables
`RDP Session`	Conexiones RDP activas o guardadas
`DCOnly`	Info específica del controlador de dominio
`ObjectProps`	Propiedades adicionales (SPNs, delegación, etc.)

2.5. Archivos generados

Después de unos minutos, dependiendo del tamaño del dominio, SharpHound generará un archivo ZIP como:

```
YYYYMMDDHHMMSS_BloodHound.zip
```

Este archivo contendrá varios `.json.gz`, como:

- `computers.json.gz`
- `users.json.gz`

- `sessions.json.gz`
- `acl.json.gz`
- `group_membership.json.gz`
- etc.

Este archivo ZIP es el que tiene que importar directamente en BloodHound.

2.6. Transferencia del archivo a Kali Linux

Desde Kali, usar `scp` para llevar el archivo:

```
scp usuario@10.0.0.191:/Users/Public/20250925112233_BloodHound.zip ~/bloodhound-data/
```

Reemplazar:

- `usuario`: el usuario que se tiene en el cliente Windows (ej. `usuario1`)
- `10.0.0.191`: IP de la máquina cliente Windows
- `~/bloodhound-data/`: carpeta en Kali donde se guardará el archivo

Alternativas si se está en red interna sin SSH:

- Compartir la carpeta vía `Samba`
- Usar `python -m http.server` para servirlo desde Windows, o
- Copiar por USB

Recomendaciones de Seguridad en entornos reales

En entornos de producción o simulados más avanzados, se puede usar:

- `-c stealth` solo algunos recolectores ya que evita detección
- `--loop` para mantener recolección continua
- `-o` para elegir un nombre personalizado para el ZIP

3. Análisis con BloodHound

3.1. Inicia Neo4j (base de datos)

```
sudo apt install neo4j
```

Opción A — Si se instaló Neo4j desde apt en Kali:

Abrir una terminal y arrancar el servicio en modo consola, útil para ver logs:

```
sudo systemctl start neo4j
sudo systemctl status neo4j    # para verificar que esté activo
```

```
# o para ver logs en tiempo real
journalctl -u neo4j -f
```

Opción B — si se prefire el modo console ya instalado:

```
sudo neo4j console
```

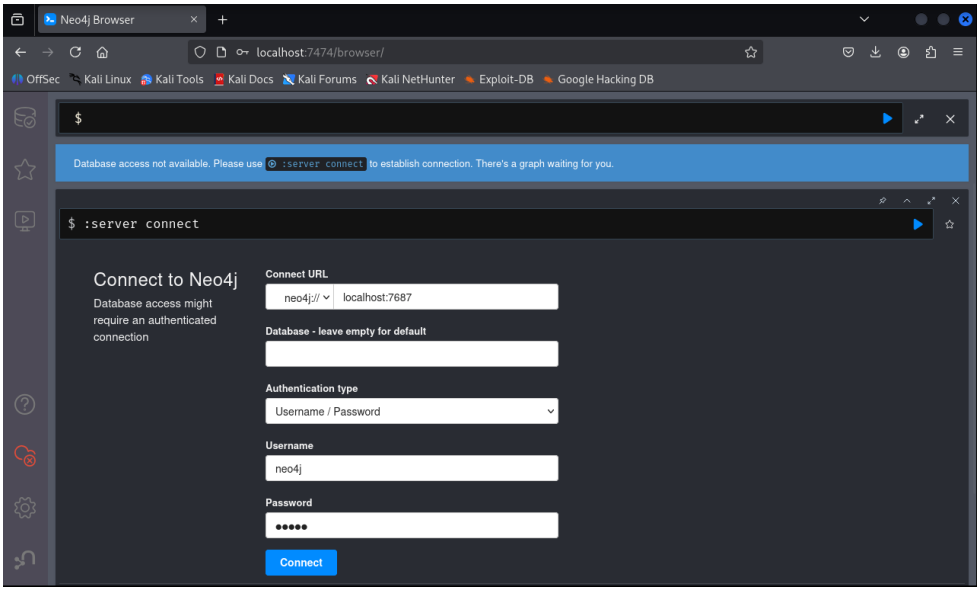
```
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.

2025-09-25 21:56:47.227+0000 INFO  Starting...
2025-09-25 21:57:05.351+0000 INFO  This instance is ServerId{55086e21} (55086e21-17a1-4936-aa2a-19885b5a63fb)
2025-09-25 21:57:09.911+0000 INFO  ======== Neo4j 4.4.26 ========
2025-09-25 21:57:13.385+0000 INFO  Initializing system graph model for component 'security-users' with version -1 and status
UNINITIALIZED
2025-09-25 21:57:13.394+0000 INFO  Setting up initial user from defaults: neo4j
2025-09-25 21:57:13.396+0000 INFO  Creating new user 'neo4j' (passwordChangeRequired=true, suspended=false)
2025-09-25 21:57:13.446+0000 INFO  Setting version for 'security-users' to 3
2025-09-25 21:57:13.452+0000 INFO  After initialization of system graph model component 'security-users' have version 3 and s
tatus CURRENT
2025-09-25 21:57:13.472+0000 INFO  Performing postInitialization step for component 'security-users' with version 3 and statu
s CURRENT
2025-09-25 21:57:14.236+0000 INFO  Bolt enabled on localhost:7687.
2025-09-25 21:57:16.345+0000 INFO  Remote interface available at http://localhost:7474/
2025-09-25 21:57:16.351+0000 INFO  id: 86CDDECECBB573D5C05780249311D0F3CF208F40B63E1FD8750B231B2C961425
2025-09-25 21:57:16.352+0000 INFO  name: system
2025-09-25 21:57:16.352+0000 INFO  creationDate: 2025-09-25T21:57:11.06Z
2025-09-25 21:57:16.353+0000 INFO  Started.
```

Si Neo4j no arranca por falta de memoria, aumentar el heap en `/etc/neo4j/neo4j.conf` o ajustar `dbms.memory.heap.initial_size` / `dbms.memory.heap.max_size`.

3.2. Acceder a la interfaz web de Neo4j por primera vez

1. En Kali abrir: `http://localhost:7474`
2. Usuario por defecto: `neo4j`
   Contraseña por defecto: `neo4j` , pedirá cambiarla en el primer login.
3. Definir una contraseña ‘password’



3.3. Inicia BloodHound

En Kali:

```
./BloodHound --disable-gpu
```

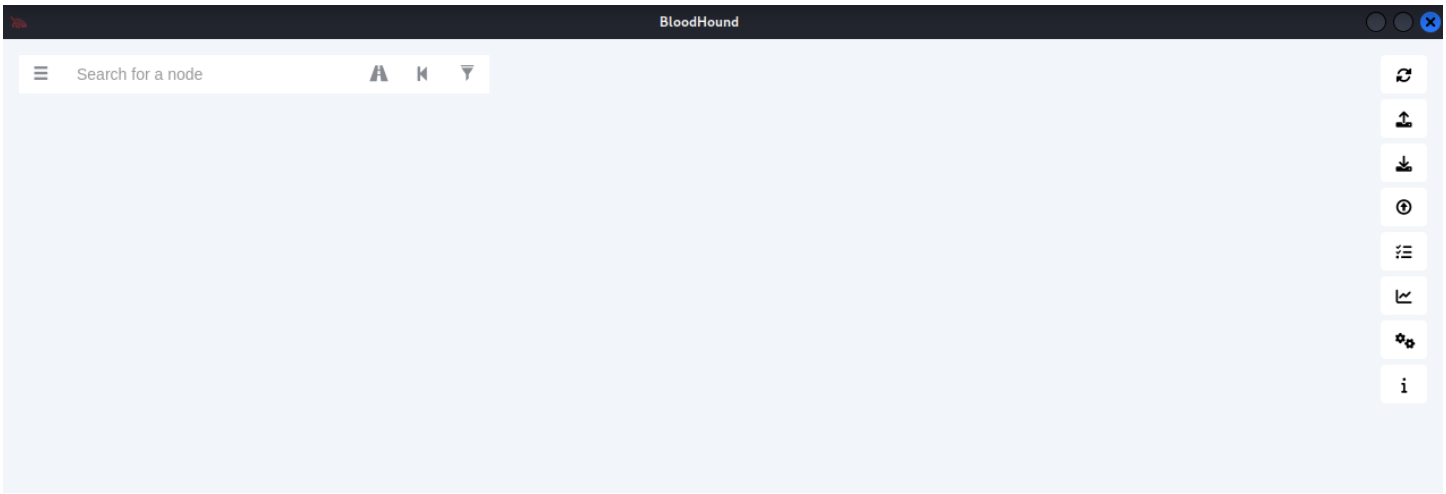
Se abrirá la interfaz GUI de BloodHound (Electron). Si se prefiere la app nativa, lanzarla desde el menú.

3.4. Conectar BloodHound a Neo4j

En la ventana de login de BloodHound:

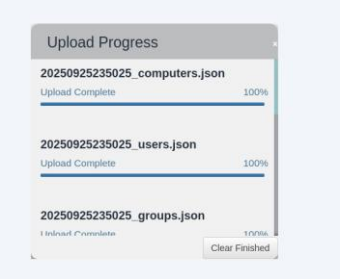
- URI: `bolt://localhost:7687` (o `bolt://<ip>:7687` si Neo4j en otra máquina)
- Username: `neo4j`
- Password: la que se creó anteriormente
- Click **Login**

Si aparece error: revisar que Neo4j esté corriendo y que el puerto 7687 esté accesible.



3.5. Importar el archivo `.zip` generado por SharpHound

1. En la GUI de BloodHound, arriba a la izquierda hacer clic en **Upload Data** o arrastrar el `.zip` al área de la app.
2. Seleccionar el archivo `YYYYMMDD_HHMM_BloodHound.zip` generado por SharpHound.
3. Esperar a que termine la importación. Apareceran logs en la parte inferior y, cuando acabe, el grafo se cargará automáticamente.



Si la importación falla, comprobar:

- Que el archivo `.zip` no esté corrupto.
- Que no haya versiones incompatibles entre SharpHound y BloodHound.

3.6. Familiarizarse con la GUI y los elementos del grafo

En la interfaz se mostrarán nodos y aristas:

- **Nodos (colores/íconos distintos):**
  - Users (personas)
  - Computers (equipos)
  - Groups (grupos)
  - OUs, Domains, GPOs, etc.
- **Relaciones (aristas):**
  - `MemberOf` - miembro de un grupo
  - `AdminTo` - admin en un equipo o entidad
  - `HasSession` - sesión activa en equipo
  - `HasSPN` / `SPN` - servicio con SPN (Kerberoastable)
  - `AllowedToDelegate` / `TrustedForDelegation` - delegación
  - `CanRDP` - RDP permitido
  - `GenericAll` / `GenericWrite` - permisos sobre objetos (ACLs)

Pasar el cursor sobre nodos/aristas para ver propiedades (ej. `samaccountname`, `lastlogon`, `objectid`, `adminCount`).

3.7. Queries predefinidas: cómo usarlas y qué significan

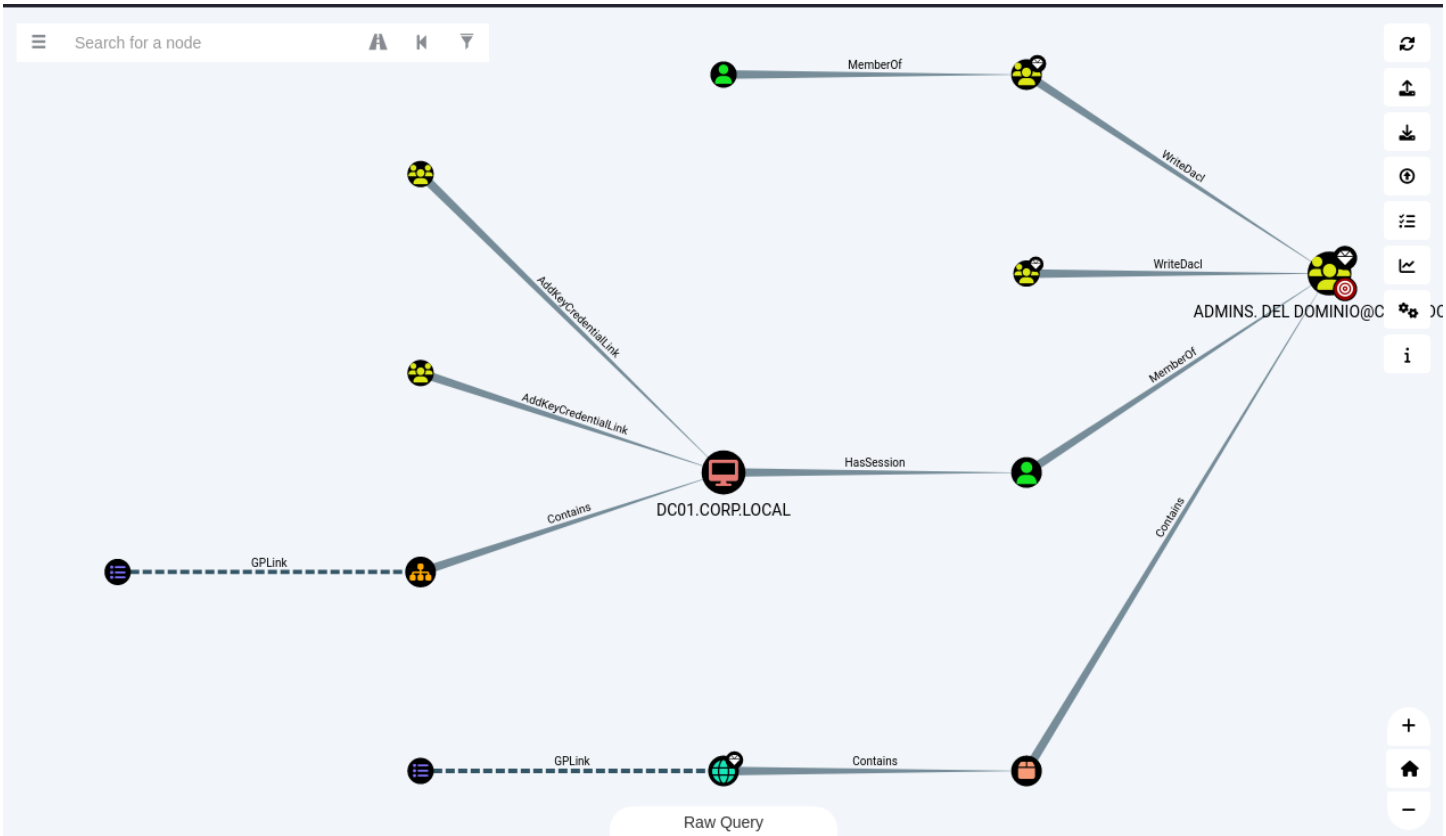
BloodHound incluye un listado de *prebuilt queries* (menú izquierdo). Las más útiles para la práctica:

a) **Shortest path to Domain Admins**

- **Qué hace:** calcula el camino más corto desde un usuario o conjunto de usuarios hasta el grupo `Domain Admins`.
- **Por qué importa:** revela rutas de escalada (ej. Usuario → miembro de GrupoX → Grupo con AdminTo sobre DC → Domain Admins).

- **Interpretación:** cada salto es un vector de escalada (membresía, ACL, sesiones, delegación). Identifica cuentas y recursos críticos en la ruta.

Acción: Ejecuta la query globalmente o selecciona un usuario y pide "Shortest path to Domain Admins". Toma nota de usuarios intermedios y permisos concretos (p. ej. `GenericAll` sobre un grupo).



b) **Users with Delegation Rights**

- **Qué hace:** lista cuentas de usuario/servicio que tienen permiso de delegación (trusted for delegation, etc.).
- **Por qué importa:** la delegación mal configurada permite movimiento lateral e impersonación.
- **Interpretación:** localiza máquinas que aceptan tickets para cuentas delegadas o cuentas de servicio con `trustedForDelegation=true`.

Mitigación: comprobar `trustedForDelegation`, aplicar constrained delegation o eliminar delegación innecesaria.

c) **Kerberoastable Users**

- **Qué hace:** muestra usuarios con SPNs (Service Principal Names) asignados — objetivos de Kerberoasting.

- **Por qué importa:** cuentas con SPN permiten solicitar TGS en modo cifrado susceptible a cracking offline.
- **Interpretación:** identifica cuentas de servicio con privilegios elevados (service accounts). Prioriza cracking o mitigación (migrar a managed service accounts, poner contraseñas fuertes).

d) **Unconstrained Delegation**

- **Qué hace:** muestra equipos marcados como `TrustedForDelegation` sin restricciones (unconstrained).
- **Por qué importa:** si un equipo está en unconstrained delegation, un atacante con control del equipo puede obtener tickets en nombre de usuarios que se autentiquen ahí, incluso Domain Admins.
- **Interpretación:** localiza hosts críticos (p. ej. servidores con RDP o servicios) que deben ser corregidos.

e) **Groups with Admin Rights on Computers**

- **Qué hace:** lista grupos que tienen privilegios administrativos en máquinas concretas.
- **Por qué importa:** si un usuario pertenece a uno de esos grupos puede escalar a administrador local en esas máquinas; desde allí, utilizar passthrough o sacarle provecho.
- **Interpretación:** identifica grupos con excesivos permisos locales — revisar membresías.

3.8. Ejecución práctica: ejemplo de flujo de análisis

1. Ejecutar **"All Domain Admins"** para ver el objetivo.
2. Ejecutar **"Shortest path to Domain Admins"** y revisar los caminos.
   - Si ves una arista `MemberOf` seguida de `AdminTo` en un equipo, eso es un vector de escalada.
3. Para cada usuario intermedio inspecciona el nodo:
   - ¿Tiene `adminCount=1`? ¿SPNs? ¿Último logon? ¿Tiene `pwdLastSet` antiguo?
4. Ejecutar **"Kerberoastable Users"**:
   - Lista usuarios con SPN; exporta para intentar `GetUserSPNs.py` desde Kali si estás autorizando pruebas.
5. Ejecutar **"Unconstrained Delegation"**:
   - Si aparecen hosts, marca como críticos y documenta.
6. Ejecutar **"Users with Delegation Rights"** y **"Groups with Admin Rights on Computers"** para cerrar el círculo.

3.9. Uso de consultas personalizadas (Cypher) — ejemplos prácticos

BloodHound permite ejecutar Cypher directamente en la barra de consultas. Algunos ejemplos útiles:

- Encontrar usuarios con SPN**:**

```
MATCH (u:User) WHERE u.hasspn = true RETURN u.sAMAccountName, u.name, u.memberOf
LIMIT 50;
```

- Buscar equipos con delegación unconstrained**:**

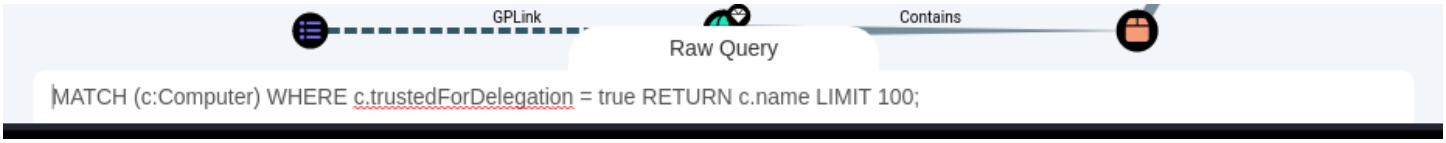
```
MATCH (c:Computer) WHERE c.trustedForDelegation = true RETURN c.name LIMIT 100;
```

- Camino corto desde un usuario hasta Domain Admins (ejemplo conceptual)**:**

```
MATCH (u:User {sAMAccountName:"usuario1"}) , (g:Group {name:"DOMAIN ADMINS"})
MATCH p=shortestPath((u)-[*..8]-(g))
RETURN p;
```

  Ajusta `sAMAccountName` y `name` a tu dominio, comprueba propiedades de tus nodos.

Si no estás cómodo con Cypher, usa las queries predefinidas ya que suelen resolver la mayoría de necesidades.



3.10. Interpretación y documentación de hallazgos

Para cada hallazgo importante se documenta:

- **Hallazgo:** e.g. `Host05` con `Unconstrained Delegation`
- **Impacto:** riesgo de escalada a Domain Admin si se compromete Host05
- **Evidencia:** captura de la pantalla de BloodHound / propiedades (node detail)
- **Recomendación:** quitar `TrustedForDelegation`, mover servicios a constrained delegation, revisar cuentas afectadas
- **Prioridad:** alta/media/baja

Ejemplo para Kerberoasting:

- Hallazgo: `svc_web` tiene SPN `HTTP/web.corp.local`
- Impacto: ticket TGS crackeable → posibilidad de obtener hash de contraseña
- Recomendación: usar Managed Service Accounts / contraseñas fuertes / auditar uso de SPNs

3.11. Exportar y reportar resultados

- **Exportar tablas:** en la vista de query se puede exportar resultados CSV.
- **Capturas:** hacer screenshots de rutas (Shortest Path) y nodos relevantes.
- **Export Graph:** se puede usar la función de exportación (graph export) o guardar imágenes para incluir en el informe.
- **Informes:** estructura sugerida en tu práctica: descripción, metodología (SharpHound params), hallazgos (con evidencias), recomendaciones técnicas y prioridades.

3.12. Buenas prácticas al usar BloodHound en laboratorio

- Usar cuentas de prueba; evitar usar cuentas reales/prod.
- Mantener snapshots antes de pruebas invasivas.
- Emplear el modo `-c stealth` en SharpHound si se simula detección o pruebas con EDRs.
- Anotar versiones de SharpHound/BloodHound/Neo4j en el informe (compatibilidad).

3.13. Ejemplos rápidos de mitigaciones vinculadas a cada query

- **Shortest path to Domain Admins:** eliminar membresías innecesarias, revisar ACLs `GenericAll`.
- **Users with Delegation Rights / Unconstrained Delegation:** usar constrained delegation o eliminar delegaciones.
- **Kerberoastable Users:** rotación de contraseñas, Managed Service Accounts, limitar SPNs.
- **Groups with Admin Rights on Computers:** reducir grupos con admins locales, usar LAPS para contraseñas locales.

4. Mitigaciones Recomendadas

Riesgo Detectado	Mitigación Sugerida
Usuarios con GenericAll sobre objetos	Revisar delegaciones y aplicar principio de menor privilegio
Delegación sin restricciones (Unconstrained)	Migrar a delegación restringida o eliminar donde no sea necesario
SPNs en cuentas de usuario (Kerberoast)	Usar cuentas administradas o cambiar a cuentas de servicio sin privilegios altos
GPOs mal configuradas	Auditar permisos en GPO y asegurar que solo administradores las modifiquen
Contraseñas locales reutilizadas	Implementar **LAPS** para rotación automática y única por equipo