🔴🐶 BloodHound

Intro

BloodHound is a tool that is based on the theory of **graphs**, and what it is going to do is collect all the information from our Domain and objects to represent them in the form of Graphs 📊 with the relationships between them. This tool is probably going to be the most important, in terms of **information collection** in Active Directory Environment. This goes even beyond just collecting information, but also, it is a tool that will help us carry out **vulnerability analysis.**

The components of this tool will be two:

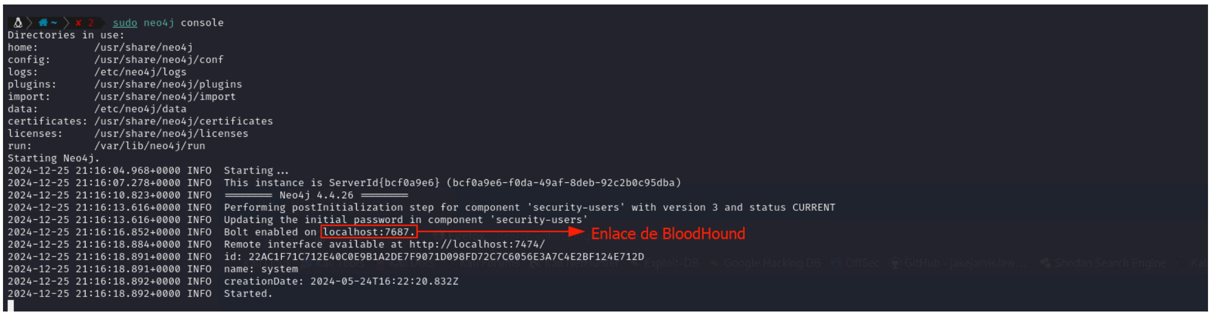
1) **The main application:** This is what we are going to use to represent the objects of our domain and the relationships through a graphical interface. And they can be installed on any operating system, but in this case we will use Linux
2) **Collectors:** These are some modules that Blood Hound offers us that we have to run on the machine that we have linked to Dominio so that it collects the information and then generates a .zip file that we will take to the machine where we have BloodHound, pass it to its interface and that will graphically represent the result. This collector is called **sharphound.** And if we want to use it against Windows systems we can use the **sharphound.ps1**

📥⚙️ Installation and Configuration

We can install this simply by performing:
sudo apt install bloodhound.

BloodHound is based on a non-relational database called **neo4j.** And to perform the initial configuration we only have to do from the command console:

sudo neo4j console



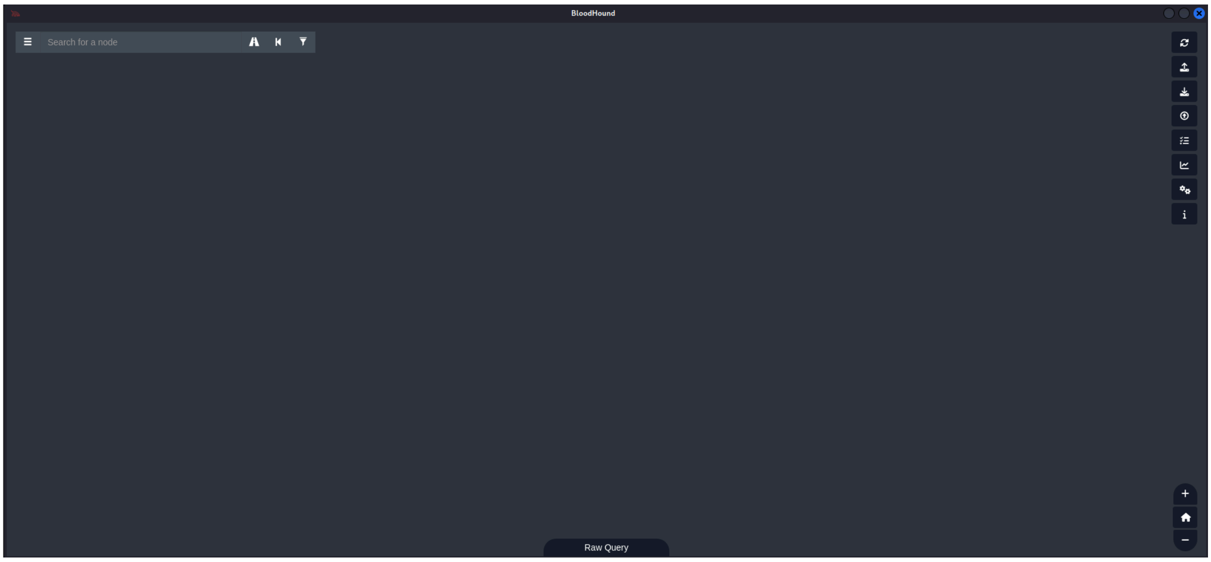
```
~ ✘ 2  sudo neo4j console
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.
2024-12-25 21:16:04.968+0000 INFO  Starting...
2024-12-25 21:16:07.278+0000 INFO  This instance is ServerId{bcf0a9e6} (bcf0a9e6-f0da-49af-8deb-92c2b0c95dba)
2024-12-25 21:16:10.823+0000 INFO  ======== Neo4j 4.4.26 ========
2024-12-25 21:16:13.616+0000 INFO  Performing postInitialization step for component 'security-users' with version 3 and status CURRENT
2024-12-25 21:16:13.616+0000 INFO  Updating the initial password in component 'security-users'
2024-12-25 21:16:16.852+0000 INFO  Bolt enabled on localhost:7687.
2024-12-25 21:16:18.884+0000 INFO  Remote interface available at http://localhost:7474/
2024-12-25 21:16:18.891+0000 INFO  id: 22AC1F71C712E40C0E9B1A2DE7F9071D098FD72C7C6056E3A7C4E2BF124E712D
2024-12-25 21:16:18.891+0000 INFO  name: system
2024-12-25 21:16:18.892+0000 INFO  creationDate: 2024-05-24T16:22:20.832Z
2024-12-25 21:16:18.892+0000 INFO  Started.
```

Enlace de BloodHound

Then we enter the link so that we can access the graphical interface of the database from the browser.

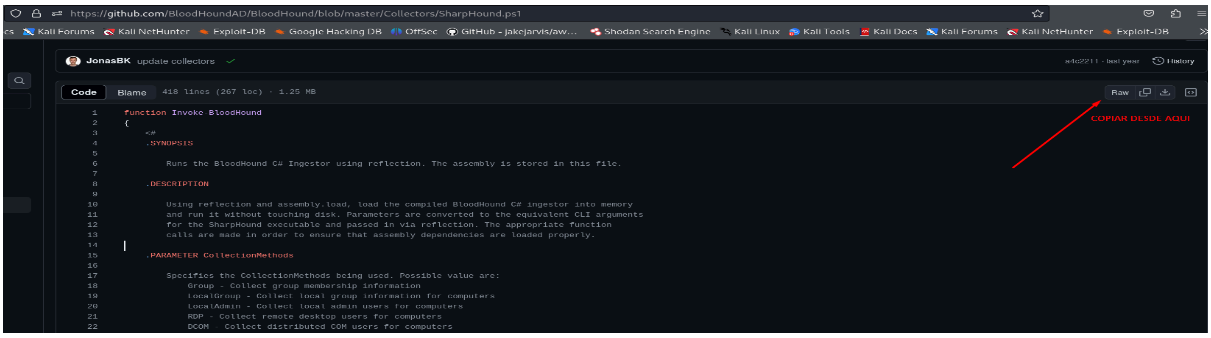
Once inside, all we have to do is change the default password, which is **neo4j** (that is also the default user) put the password we want (or leave it as is my life **:***). The database **neo4j** will be running in the **port 7687** from our local system, as you can see in the image above.

With this we only have to call the framework by placing it in our console **bloodhound** from the terminal, or by searching for the framework in the start menu, we log in and that's it 👌 .



🐕 Hunting with bloodhound

The practical use of Bloodhound is as simple as going to the repository of **GitHub** from our Kali machine (Windows defender won't let us enter this repository), Go to **Collectors,** and once there open **sharphound.ps1,** view in **Raw** from **GitHub** and from the Raw visualization we copy the code and create a file with extension **.ps1** on our Windows machine Active directory.


https://github.com/BloodHoundAD/BloodHound/blob/master/Collectors/SharpHound.ps1
cs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec GitHub - jakejarvis/aw... Shodan Search Engine Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB
JonasBK update collectors
a4c2211 · last year
History
Code
Blame
418 lines (267 loc) · 1.25 MB
Raw

COPIAR DESDE AQUI

```
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
        for the SharpHound executable and passed in via reflection. The appropriate function
        calls are made in order to ensure that assembly dependencies are loaded properly.

    .PARAMETER CollectionMethods

        Specifies the CollectionMethods being used. Possible value are:
            Group - Collect group membership information
            LocalGroup - Collect local group information for computers
            LocalAdmin - Collect local admin users for computers
            RDP - Collect remote desktop users for computers
            DCOM - Collect distributed COM users for computers
```

This powershell script will also be on the attacking machine where we have installed **bloodhound**, since installing this tool also downloads all the repo information, which includes this powershell script. This script is located in directory: **/usr/lib/bloodhound/resources/app/Collectors/SharpHound.ps1**

We can copy the script and take it wherever we want.
He script of **sharphound.ps** We have to take it to our domain machine or to the computer that we have compromised (see file sharing 📤 **File Transfer With Python Server)** and execute the following from PowerShell on said Windows machine:

1) **. .\Nombredelscrip.ps1 (from the path where we created the file)**
2) **Invoke-BloodHound -CollectionMethod All**

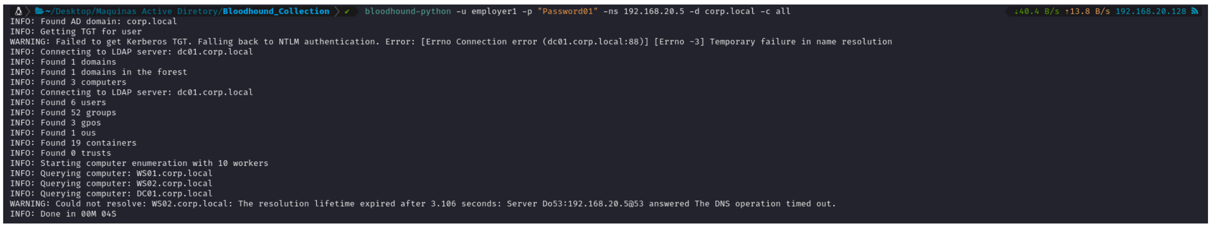
This will generate a File **.zip** that we will transfer to the machine where we have installed **BloodHound**, following the method we prefer. And once executed **BloodHound** With the mouse cursor we pass the file **.zip** and we will now have access to all the mapping that has been done to the entire Domain server.

🐍🐕‍🦺bloodhound-python

Sometimes the antivirus may block the script **SharpHound.ps1.** If this were to happen, there is another method to achieve this from our Kali machine with the utility **bloodhound-python**.

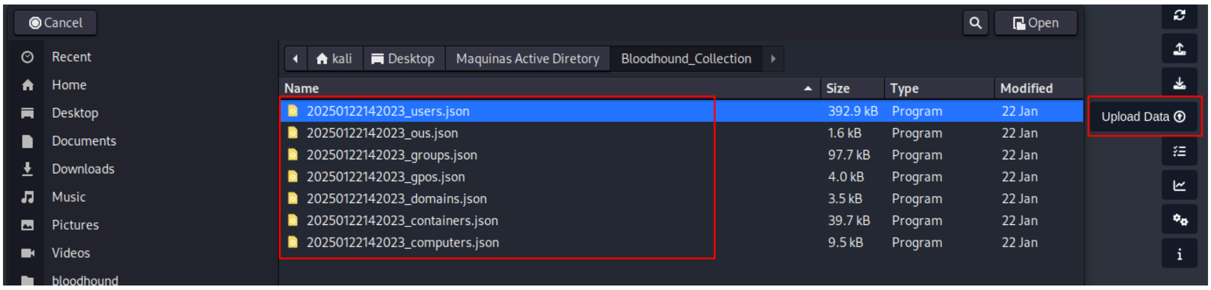
This utility will allow us to enumerate the domain by passing it a series of parameters, including **the credentials of a domain user** that we have compromised or that has been passed to us for auditing, the IP address of the domain controller and the domain name. The command would look like this:

bloodhound-python -u userAD -p "Password" -ns 10.10.20.5 -d dominio.local -c all --dns-tcp



```
~/Desktop/Maquinas Active Diretory/Bloodhound_Collection ✓  bloodhound-python -u employer1 -p "Password01" -ns 192.168.20.5 -d corp.local -c all      ↓40.4 B/s ↑13.8 B/s 192.168.20.128
INFO: Found AD domain: corp.local
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc01.corp.local:88)] [Errno -3] Temporary failure in name resolution
INFO: Connecting to LDAP server: dc01.corp.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 3 computers
INFO: Connecting to LDAP server: dc01.corp.local
INFO: Found 6 users
INFO: Found 52 groups
INFO: Found 3 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: WS01.corp.local
INFO: Querying computer: WS02.corp.local
INFO: Querying computer: DC01.corp.local
WARNING: Could not resolve: WS02.corp.local: The resolution lifetime expired after 3.106 seconds: Server Do53:192.168.20.5@53 answered The DNS operation timed out.
INFO: Done in 00M 04S
```

This will generate a series of .JSON files which are what we will pass bloodhound through **Upload Data**.

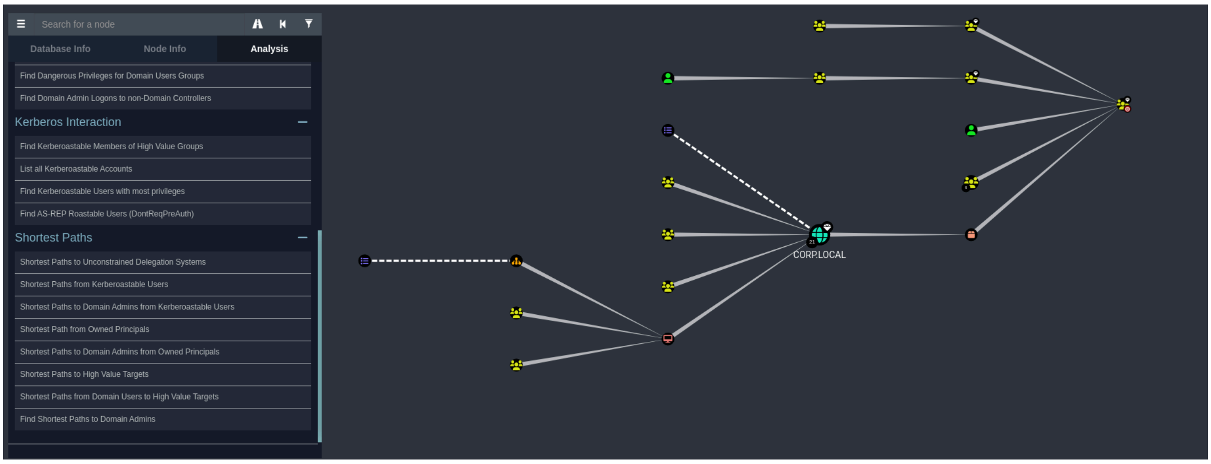


Name	Size	Type	Modified
20250122142023_users.json	392.9 kB	Program	22 Jan
20250122142023_ous.json	1.6 kB	Program	22 Jan
20250122142023_groups.json	97.7 kB	Program	22 Jan
20250122142023_gpos.json	4.0 kB	Program	22 Jan
20250122142023_domains.json	3.5 kB	Program	22 Jan
20250122142023_containers.json	39.7 kB	Program	22 Jan
20250122142023_computers.json	9.5 kB	Program	22 Jan

It is important to use the command from a particular folder, so as not to become a mess when we pass the files to bloodhound.

🐕‍🦺 Bloodhound in action

With all that information that we have uploaded we can see all the domain information **on *graphs form***:


Search for a node
Database Info
Node Info
Analysis
Find Dangerous Privileges for Domain Users Groups
Find Domain Admin Logons to non-Domain Controllers
Kerberos Interaction
Find Kerberoastable Members of High Value Groups
List all Kerberoastable Accounts
Find Kerberoastable Users with most privileges
Find AS-REP Roastable Users (DontReqPreAuth)
Shortest Paths
Shortest Paths to Unconstrained Delegation Systems
Shortest Paths from Kerberoastable Users
Shortest Paths to Domain Admins from Kerberoastable Users
Shortest Path from Owned Principals
Shortest Paths to Domain Admins from Owned Principals
Shortest Paths to High Value Targets
Shortest Paths from Domain Users to High Value Targets
Find Shortest Paths to Domain Admins
CORP.LOCAL

Now we can use bloodhound to enumerate all the domains, groups, objects, users... We can see all this information in a graphic and organized way.

And If you look closely at **BloodHound**, there are **three sidebar** options that are critical to navigating and understanding the information collected within the Active Directory graph.

1) 📂 **Database Info:** shows us general information about the current database that BloodHound is using.

2) 🧑‍💼**Node Info:** This option appears when we select a node (for example, a user, a group or a computer) within the graph and give us specific details about that node like properties of that object, direct relationships it has with other nodes and more

3) 🧠 **Analysis:** This is where the magic happens. This section includes automated analysis tools to help you identify ***common attack paths.***

As the lessons progress, we will see how we can use these Bloodhound functions when performing enumeration and attacks in AD environments. Without further ado, this lesson dedicated to bloodhound ends here.