

Attack Surface Management Exercises - Educause CPPC 2023

version 1.5 4/27/2023

DNS Enumeration with dnsdumpster.com (5 minutes)

1. Open a web browser and go to <https://dnsdumpster.com/>.
2. Enter `cyberspacekittens.com` in the search bar.
3. Click **Search**
4. Review the results, do you see anything interesting?

```
TXT Records ** Find more hosts in Sender Policy Framework (SPF) configurations
"v=spf1 include:spf.efwd.registrar-servers.com ~all"

Host Records (A) ** this data may not be current as it uses a static database (updated monthly)

cyberspacekittens.com          34.216.187.82          AS16509 Amazon.com, Inc.
■■ ④ ④ ④ ④
HTTP: Apache/2.4.18 (Ubuntu)   ec2-34-216-187-82.us-west-
HTTPS: Apache/2.4.18 (Ubuntu)   2.compute.amazonaws.com
SSH: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8

neverfindthissubdomain.cyberspacekittens.com 34.216.187.82          AS16509 Amazon.com, Inc.
■■ ④ ④ ④ ④
HTTP: Apache/2.4.18 (Ubuntu)   ec2-34-216-187-82.us-west-
HTTPS: Apache/2.4.18 (Ubuntu)   2.compute.amazonaws.com
SSH: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
```

5. Now try the same thing with your organization!

Nmap (15 minutes)

Review the slide for this exercise for an IP address you can use for these exercises. You are also welcome to use an IP address/host/subnet that you have permission to scan.

`target` = replace with the IP address, hostname, subnet, etc that you are attempting to scan

`nmap target` - This will perform a TCP connect (if running as a regular user) or TCP SYN scan (if running as root/privileged) scan with default timeout settings against the top 1000 used TCP ports

```
[root@kali:~# nmap 192.168.2.159
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-09 20:58 EST
Nmap scan report for 192.168.2.159
Host is up (0.00073s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8089/tcp  open  unknown
MAC Address: 00:0C:29:98:F0:3F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.56 seconds
root@kali:~#
```

Common scan types

-sT	full TCP connect scan (default for non-privileged users)
-sS	SYN scan (default for privileged users)
-sU	SYN scan (default for privileged users)

Target input

DNS	<code>www.umd.edu</code>
CIDR blocks	<code>128.8.0.0/16</code>
IP Ranges	<code>10.0.0-255.1-254</code>
Multiple	<code>www.umd.edu 129.2.0.0/16</code>
From a file	<code>-iL filename</code> Example: <code>[kts@ITMC011285 scans % cat campus.txt 128.8.0.0/16 129.2.0.0/16 192.54.94.0/21 206.196.160.0/19</code>

Port selection -p

<code>-p22</code>	Scan port 22/tcp (SSH)
<code>-p1-65535</code>	Scan all 65,535 TCP ports
<code>-p U:53,111,137,T:21-25,80,139,8080</code>	Scan UDP ports 53, 111, 137 Scan TCP ports 21-15, 80, 139, 8080

Timing -T

`-T 0-5` - 0 = super slow, 5 = super fast

OS Fingerprinting -O

```
[root@kali:~# nmap -O 192.168.2.159
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-09 20:55 EST
Nmap scan report for 192.168.2.159
Host is up (0.00037s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8089/tcp  open  unknown
MAC Address: 00:0C:29:98:F0:3F (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.36 seconds
```

Service Versioning -sV

```
[root@kali:~# nmap -sV 192.168.2.159
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-09 21:20 EST
Nmap scan report for 192.168.2.159
Host is up (0.00038s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol
2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
8000/tcp  open  http     Splunkd httpd
8089/tcp  open  ssl/http Splunkd httpd (free license; remote login disabled)
MAC Address: 00:0C:29:98:F0:3F (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.98 seconds
```

NSE - Nmap Scripting Engine

Format: `nmap --script=all,category,dir,script`

Examples:

```
nmap -p 443 --script ssl-enum-ciphers target
nmap -p 22 --script "ssh*" target
nmap -p 135-139,445 --script smb-os-discovery target
nmap -p 80,443 --script "ssl-*,http-*" target

nmap --script smb-os-discovery target
```

```
[root@kali:/home/kts# nmap --script smb-os-discovery 172.16.0.129
Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-10 22:15 EDT
Nmap scan report for 172.16.0.129
Host is up (0.00012s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3000/tcp  open  ppp
MAC Address: 00:0C:29:AB:97:66 (VMware)

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: enpm809q
|   NetBIOS computer name: ENPM809Q\x00
|   Domain name: \x00
|   FQDN: enpm809q
|_  System time: 2019-09-10T22:18:58-04:00

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

```
nmap -p443 --script ssl-enum-ciphers target
```

```
[kts@ITMC011285 scans % nmap -p 443 --script ssl-enum-ciphers www.umd.edu
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-16 13:33 EDT
Nmap scan report for www.umd.edu (18.160.46.99)
Host is up (0.013s latency).
Other addresses for www.umd.edu (not scanned): 18.160.46.81 18.160.46.7 18.160.46.53
rDNS record for 18.160.46.99: server-18-160-46-99.iad55.r.cloudfront.net

PORT      STATE SERVICE
443/tcp  open  https
| ssl-enum-ciphers:
|_ TLSv1.2:
|   ciphers:
|     TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|     TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|     TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|     TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecdh_x25519) - A
|     TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519) - A
|   compressors:
|     NULL
|   cipher preference: server
|_ TLSv1.3:
|   ciphers:
|     TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|     TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|     TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|   cipher preference: server
|_ least strength: A

Nmap done: 1 IP address (1 host up) scanned in 1.93 seconds
```

Output of nmap results

-oN (Normal output)	<pre>Nmap scan report for maxgigapop.net (206.196.178.94) Host is up (0.016s latency). Not shown: 497 filtered tcp ports (no-response) PORT STATE SERVICE 53/tcp open domain 80/tcp open http 443/tcp open https</pre>
-oG (“grepable” output) (not this will not work with NSE script output)	<pre>[kts@ITMC011285 scans % grep 206.196.178.94 scan.txt Host: 206.196.178.94 (maxgigapop.net) Status: Up Host: 206.196.178.94 (maxgigapop.net) Ports: 80/open/tcp//http///, 443/open/tcp//https/// Ignored State: filtered (498)]</pre>

Manipulating -oG output:

```
grep 445/open results
grep 445/open results | awk '{ print $2}'
```

```
[root@kali ~]# grep "445/open" port445.txt
Host: 172.16.0.130 ()  Ports: 445/open/tcp//microsoft-ds///
Host: 172.16.0.212 ()  Ports: 445/open/tcp//microsoft-ds///

[root@kali ~]# grep "445/open" port445.txt | awk '{print $2}'
172.16.0.130
172.16.0.212
```

Shodan (15 minutes)

Feel free to create an account on Shodan, there is a free option that allows you to unlock more searching and more details on results. If you register with a university email account (something ending in .edu) you may get a free upgrade to a full paid account.

1. Open a web browser and navigate to <https://shodan.io>
2. Start searching for anything you can think of, some examples below (and you can combine multiple search terms into one search)
 - i. Domain names (umd.edu)
 - ii. IP addresses
 - iii. Services (ssh, telnet)
 - iv. Banners (“OpenSSH”, “SSH-1.99”)
 - v. **net:XX.XX.XX.XX/YY**
 - vi. **port:443**
 - vii. **http.status:200**
 - viii. **org:Stanford**
 - ix. **country:US**
 - x. **title:cyberspacekittens**
 - xi. **apache city:“College Park”** (Find Apache servers in College Park)
 - xii. **cisco country:”JP”** (Find cisco equipment in Japan)
 - xiii. **“default password”** (Find devices with the default login credentials still set.)
 - xiv. **ssh -port:22** (Find SSH on non-standard ports)
 - xv. **vuln: CVE-2023-27350** (Search for a specific vulnerability - Paid/Academic accounts only)

Information on Shodan's search query options:

<https://help.shodan.io/the-basics/search-query-fundamentals>

truffleHog (5 minutes)

Available from <https://github.com/dxa4481/truffleHog>

Install with: **pip install truffleHog**

Searches through git repositories, commit histories, and branches for secrets. This is an effective way to discover secrets that were accidentally committed like API keys, SSH keys, passwords, etc.

Example:

```
trufflehog https://github.com/cyberspacekittens/dnscat2
```

Did we find anything interesting?

```
[kts@kali:~$ trufflehog https://github.com/cyberspacekittens/dnscat2
~~~~~
Reason: High Entropy
Date: 2018-01-13 21:58:04
Hash: 403a412ba15c765680547cb69c16570915957e2c
Filepath: server/controller/csk.config
Branch: origin/master
Commit: Config Update
@@ -1,33 +0,0 @@
-dnscat.config
- "awsSecretKey": "28dunBEhc374473Hdkql3kvdk881AYvne349KD3"
- "awsAccessId": "AKIAJFHD345JFENC34FD"
- "client_secret": "JFHe43fkwdjen3r8fjd3Dje8"
- "secretkey": "cyberspacekittenssupersecurepassword"
-
-----BEGIN RSA PRIVATE KEY-----
-MIIEpAIBAAKCAQEaWU/01pfXi0RTEzhu46rXRNAPwfY3zWMiZDnwu1sZBmedVR0n
-fx+4nPAb2dzy+/qp+DjMGs/iLfdtHC6U/9Arvh9Ni0HEfoqCoIzPG0oqyqkcut/e
-fAHxjLvRjwsoCrfcND4gtTZ29/r8mixbCa3LayDD4HZHz7C1ZMi6DajF6h0Bf1i3
-X02PBj0aybP/2GYCLc7Zpgb3+jXU6J4beuk3bA0nfZNQwpJec844I98EUN2auDaX
-0tYmunEKmt2JAuaA1vSOMC2VmM0Cu13cWRh1jMu4+mf0xgx28Lo2tpWIFabJ61T2
-HNNVnmUWTy51DoWnvF91rnOxmva48GwVLdtvHBgQIDAQABoIBAAENk3LbzuPDAqTX
-KNt6oc0RMpTG55Tp1lUfb61FmMRNkjE9gGqRmIraUATkzDoNkoHcnGvG+B9x+pkt
-s8gU9TgK6Zw4ir55uK5zs+iZ1fPwqf5mm8qnJA61MzYJRiWQKLXsJLd3/XvqVRft
-5+0Mk1AaFjCtbd62wp/i7AiJA9L7pQpJq/yCS8LjTtVrajXB9Pido/rhVLjwe3ov
-1c59NaxU0I1641RIwRVSBhrfmfe+S86IaRml8IXmpkjsyvSdxDvP0BA2LUwemrIG
-WhusZbsVd5pMP7kKyyjagTspPjm5gwpaAyHiDptFiEJck12VNLo36vKB1E9RdTZ/1
-NK0YRKUCgYEAMU/XdFEyj3HWBn9W+e58yY5afmpH6wbw0z0plvwax25R4giJ/FsB
-fJWEBNzSR/BdFFR6/C/+AwvXQ2AmYjrmGusIp8yS/2HK95vWIK00iYqSgxc6XRRk
-f7fpSSkL177D7DSJ7pvh3V/pJ1BXDjLjzJowmyN9FAj+H5uE5FEPtvsCgYEAIbxZ
-oCMxH7d9gj1M9BRKqT+BFUyMvk+1oFIdgcZ5YuAfPQbpHIsu7mb9g14t17a+Ygu1B
-92GP1z9UFmn8y2r6xJ/uxRgjL3V3D7EQmfM8jQIU35Ma7vhHA6p1Q9fHx34+0mjB
-570XVq5Ujzjyo8LT3kSQXSgmKu26ovAgq2iGLMCgYB1pw4tHBCKviy3USA7uM
-KEAVzZLHpUl5qPoig170m230LSSK60v117mcqgbxT/AsPIJvtaIr4KwP1wkc3ap6
-B2aSminMTO+fW2+6jTuM/3fQSpNqGMeS3f12ngl2H3lzPiKXPIOBJDS+IqP5T+d3
-RyRy+wIgEwzPWuZvEq7psQKBgQDGMrZc6iyHy0KXMA2nkM/kaqTztidm6a2D+0B
-HVl+0+VIYDuz+s5Xkih2H1zY9EvIgtMxyRp30/fJXc2VcWfSFDFLyPWSbMuJGake9
-4jca9yHxo5zGEH2iokMiwlzVP1SLHNGar0hC2XAvx6GJfUGlGabPvJ9XKJ0pQt51
-/MJK2wKBgQCFln5YGqRONBX20JV2EygA/EdbgFFCjwHsQ5+rcBzyBgQWTVKk1aHI
-X50vPX40HvmnPhsLhb6cHSwi3iHXNNyKQtua1ae05se4wxJNoCBuAa54eGDd/mjf
-ErtB8pFer2nL7JraiXb/ILT83rYiTq2iN+DyC0d0yesik0vMpK Ct2Q==
-----END RSA PRIVATE KEY-----
```

BloodHound Exercise (30 minutes)

BloodHound (<https://github.com/BloodHoundAD/BloodHound>) is a tool designed to help attackers and defenders graphically map out the relationships in an AD environment, many of which are often hidden and unintended. This can help us as an attacker map out the fastest path between the account access we currently have to getting Domain Administrator. It is also useful to show us who to target.

BloodHound works on most operating systems out there but we will use the packages that come for Kali.

1. On your Kali VM open a terminal and run `sudo apt-get install bloodhound` which will install BloodHound and all of the packages it needs to run. This may take a few minutes to download and install everything.
2. BloodHound runs on Neo4j, a graph database management system. To run BloodHound we need to start Neo4j with `sudo neo4j console`. You'll see some information messages and then a notice that neo4j is running at `http://localhost:7474/`

```
root@kali:~# neo4j console
Active database: graph.db
Directories in use:
  home:          /usr/share/neo4j
  config:        /usr/share/neo4j/conf
  logs:          /usr/share/neo4j/logs
  plugins:       /usr/share/neo4j/plugins
  import:        /usr/share/neo4j/import
  data:          /usr/share/neo4j/data
  certificates: /usr/share/neo4j/certificates
  run:           /usr/share/neo4j/run
Starting Neo4j...
WARNING: Max 1024 open files allowed, minimum of 40000 recommended. See the Neo4j manual.
2019-10-21 20:48:03.416+0000 INFO ===== Neo4j 3.5.3 =====
2019-10-21 20:48:03.457+0000 INFO Starting...
2019-10-21 20:48:08.763+0000 INFO Bolt enabled on 127.0.0.1:7687.
2019-10-21 20:48:12.123+0000 INFO Started.
2019-10-21 20:48:14.534+0000 INFO Remote interface available at http://localhost:7474/
```

3. In your Kali VM open up a web browser and load `http://localhost:7474/` which is the web UI for Neo4j. It will redirect you to the login prompt, the password for the neo4j user is `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

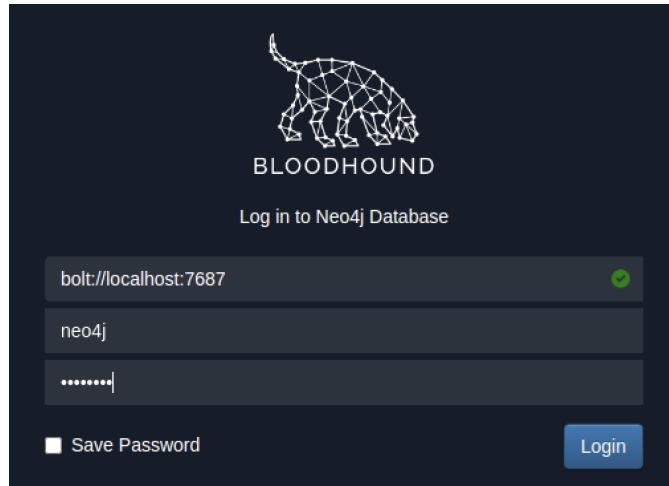
Password:

Connect

4. You'll need to select a new password, I'll use **password**.

5. Now that Neo4j is running we can start Bloodhound. Inside your Kali VM open a terminal (different from the one you started neo4j in if you started it a Terminal in Kali) and run **bloodhound**. You will see a new window pop up for BloodHound, login with the credentials you set in the previous step and then click **Login**.

Note: If the window pops up and is just blank/white type **Control+R** to refresh the window and you should see the login page for BloodHound.

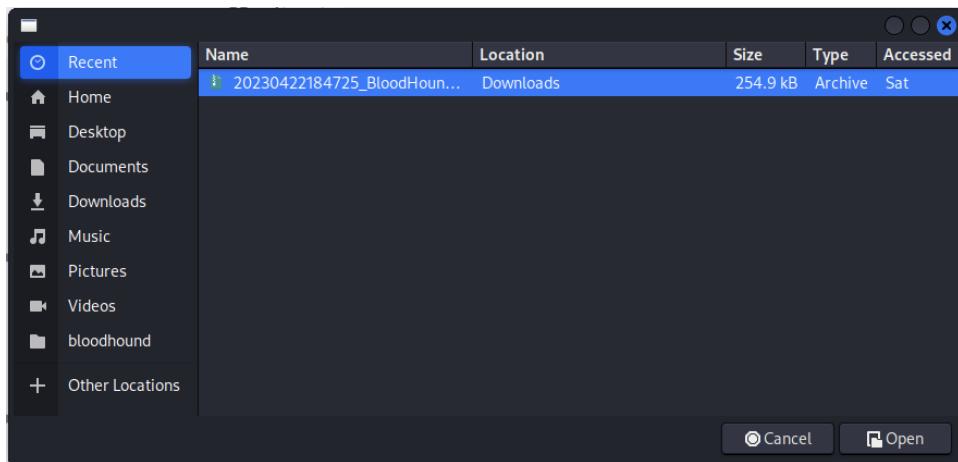


6. Once logged in we have a blank screen. Not very useful. Now we need to load in the results from SharpHound. For this example we will use a sample dataset generated by a tool called BadBlood which is designed to generate a complete Active Directory domain dataset with a number of vulnerabilities inside of it.

7. Download the AD data set from onto your Kali VM by running `wget https://github.com/kts262/ASM/raw/main/20230422184725_BloodHound.zip` from a Terminal/SSH session in your Kali VM.
8. Load this data set into BloodHound by selecting the **Upload Data** button on the top right of BloodHound.

Upload Data 

9. Next select the ZIP file from step 7 and click **Open**.



10. You'll see an Upload Progress window that will show the status of the upload/import process. You can exit out of this window when it is finished.
11. Click **Refresh Database Stats** on the lower left side of the Database Info tab and you should see a large number of Relationships, ACLs, and if you scroll down Users, Groups, Computers, etc.
12. Click the **Analysis** tab to see some of the prebuilt queries.

Search for a node

Database Info **Node Info** **Analysis**

Pre-Built Analytics Queries

Domain Information

- Find all Domain Admins
- Map Domain Trusts
- Find Computers with Unsupported Operating Systems

Dangerous Privileges

- Find Principals with DCSync Rights
- Users with Foreign Domain Group Membership
- Groups with Foreign Domain Group Membership
- Find Computers where Domain Users are Local Admin
- Find Computers where Domain Users can read LAPS passwords
- Find All Paths from Domain Users to High Value Targets
- Find Workstations where Domain Users can RDP
- Find Servers where Domain Users can RDP
- Find Dangerous Privileges for Domain Users Groups
- Find Domain Admin Logons to non-Domain Controllers

Kerberos Interaction

Start with “Find all Domain Admins” and then you can click on one of them to get more information about that specific user account.

Search for a node

Database Info **Node Info** **Analysis**

KENDALL_MORAN@DM.COM

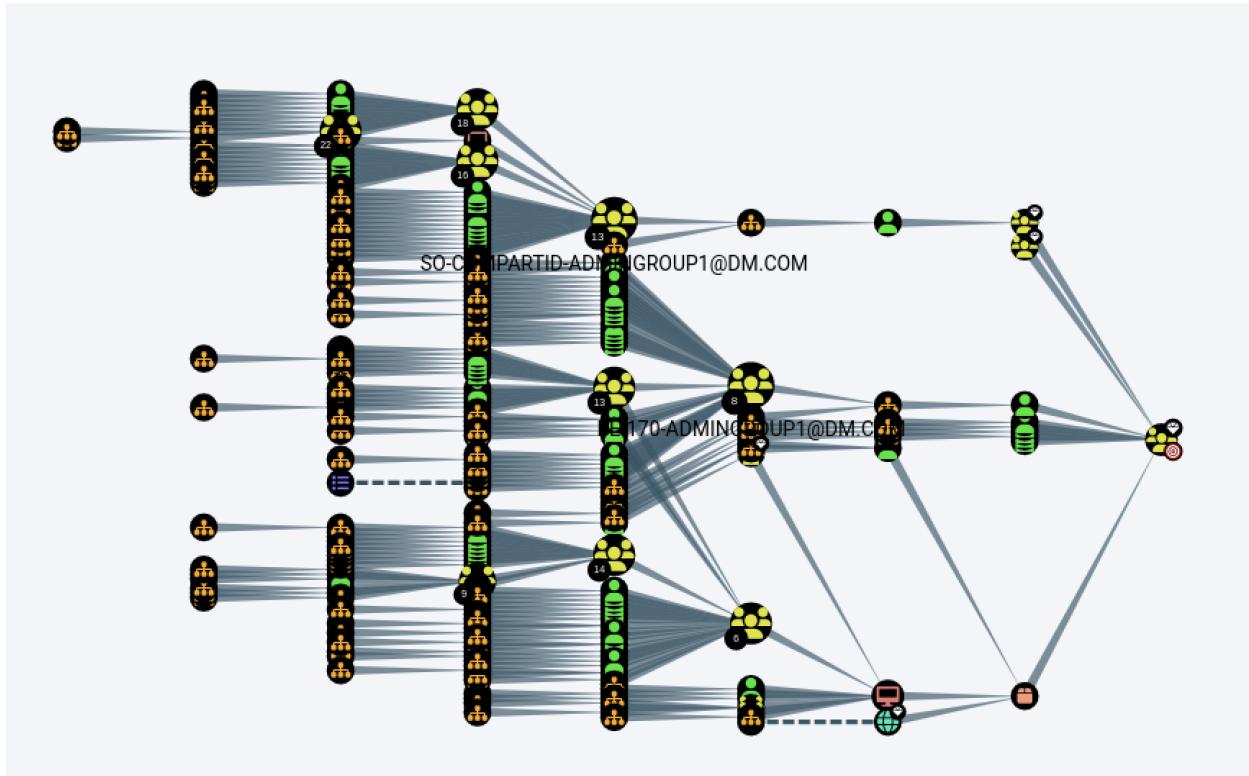
OVERVIEW

Sessions	0
Sibling Objects in the Same OU	16
Reachable High Value Targets	9
Effective Inbound GPOs	1
See user within Domain/OU Tree	

NODE PROPERTIES

Display Name	KENDALL_MORAN
Object ID	S-1-5-21-1606558351-3154278411-2097635151-1741
Password Last Change	Sat, 22 Apr 2023 19:38:09 GMT
Last Logon	0
Last Logon (Renewal)	Never

13. From here feel free to select some other queries. **Find Shortest Path to Domain Admin** is always an interesting one to show the fastest method to get to Domain Admin.



Note: some of these queries will not return any results, in this case the same data was not configured in a way to show any results for those queries. (In a production environment this is typically a good thing!)