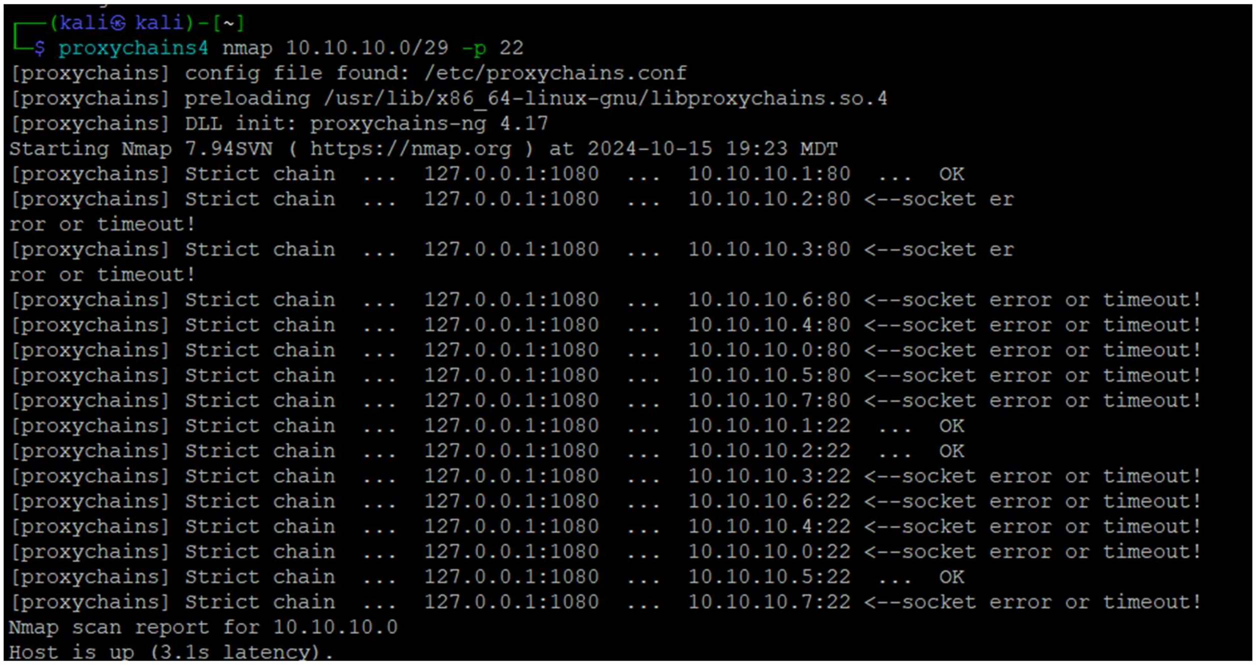
Internal Discovery and Lateral Movement

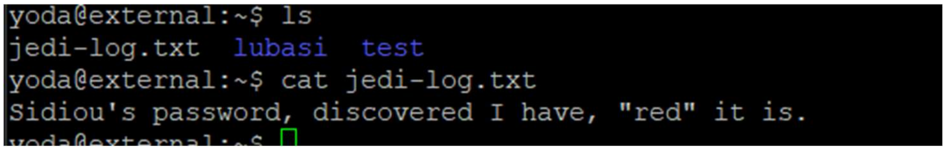
Internal Discovery

MITRE ATT&CK (Discovery) - Using proxychains we ran a nmap scan of yoda host machine to discover any machine on the same private network.



```
┌──(kali㉿ kali)-[~]
└─$ proxychains4 nmap 10.10.10.0/29 -p 22
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 19:23 MDT
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.1:80  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.2:80 <--socket er
ror or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.3:80 <--socket er
ror or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.6:80 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.4:80 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.0:80 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.5:80 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.7:80 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.1:22  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.2:22  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.3:22 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.6:22 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.4:22 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.0:22 <--socket error or timeout!
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.5:22  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.10.10.7:22 <--socket error or timeout!
Nmap scan report for 10.10.10.0
Host is up (3.1s latency).
```

The next part was searching for credentials that would later help us in the attack. After a quick search through the files, we find a file named jedi-log.txt. Finding credentials falls under the credential access (unsecured credentials). Below is a screenshot of the file and text.



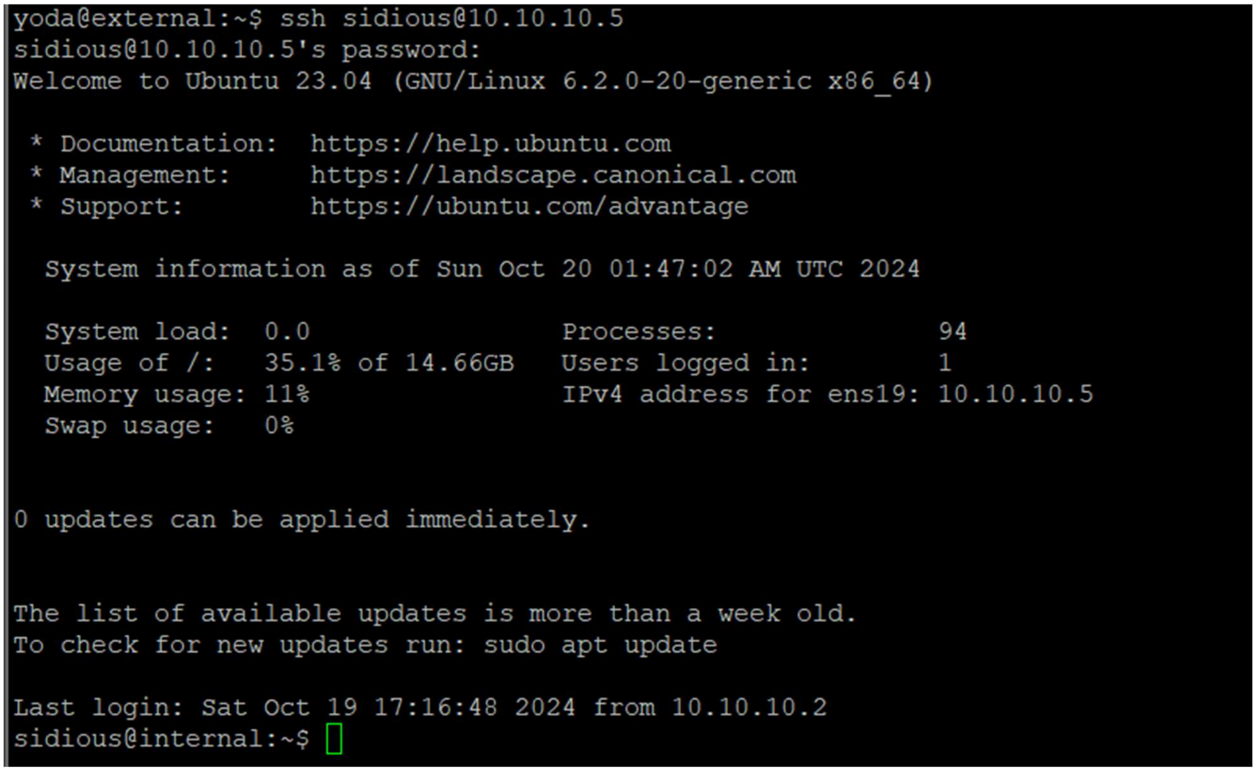
```
yoda@external:~$ ls
jedi-log.txt  lubasi  test
yoda@external:~$ cat jedi-log.txt
Sidiou's password, discovered I have, "red" it is.
yoda@external:~$ 
```

The nmap technique is a common that goes out and sends a request to communicate with the machine through a specific port. If there is a reply, then we know that the machine exists and is up and running. These tactics help us focus our efforts and what machine may be vulnerable to attack. Of course, the threat actors in these scenarios are hackers trying to gain access to the machines.

A few recommendations to help stop these attacks or detect and attack from happening would be using intrusion detection systems that will alert you when multiple signals are being sent to multiple addresses in a system. Also using SIEM (Security information and Event Management) system allows security teams to monitor and detect threats by recognizing attack patterns, malicious activity, and forensic response. That last thing I would suggest is performing file audits on different systems to make sure files are not storing open text credentials.

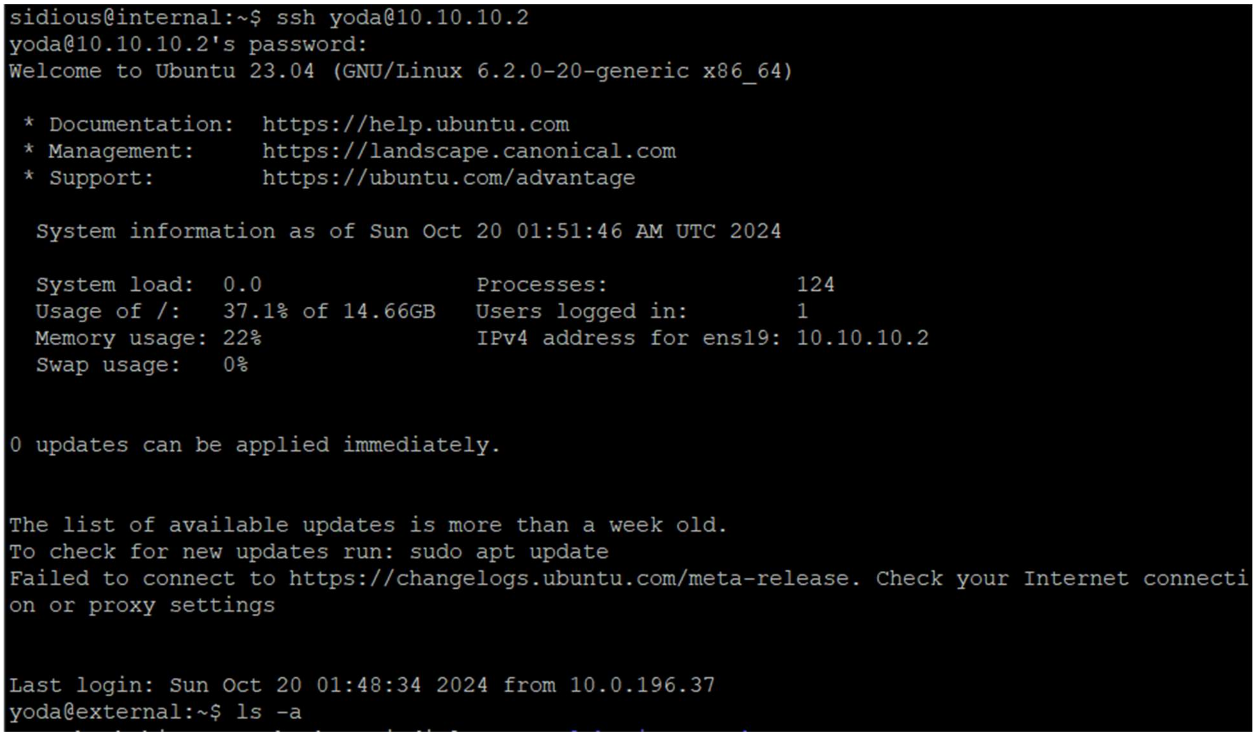
Lateral Movement

For these machines, the later movement tactics were pretty straight forward with using ssh(secure shell). When we did our scan, we could see that each machine had shh services running on each machine. For the lateral movement in MITRE ATT&CK we will be using the simple subsection of remote services. In our discovery we found Sidiou's password. We will be using this to remotely connect to the machines. The first machine we focused on was 10.10.10.5 as shown below



```
yoda@external:~$ ssh sidious@10.10.10.5
sidious@10.10.10.5's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct 20 01:47:02 AM UTC 2024

  System load:  0.0               Processes:             94
  Usage of /:   35.1% of 14.66GB   Users logged in:       1
  Memory usage: 11%                IPv4 address for ens19: 10.10.10.5
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Oct 19 17:16:48 2024 from 10.10.10.2
sidious@internal:~$ 
```

Once that was done, we moved onto machine 10.10.10.2 using the same method. However, using Sidious credentials did not work however used another already known credentials, yoda. With these credentials were able to once again gain access to the machine. This is also shown below.



```
sidious@internal:~$ ssh yoda@10.10.10.2
yoda@10.10.10.2's password:
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct 20 01:51:46 AM UTC 2024

  System load:  0.0               Processes:             124
  Usage of /:   37.1% of 14.66GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for ens19: 10.10.10.2
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connecti
on or proxy settings


Last login: Sun Oct 20 01:48:34 2024 from 10.0.196.37
yoda@external:~$ ls -a
```

The lateral movement tactic is very dangerous because it gives the threat actors (i.e. hackers, cybercriminals, nation-state actors) more access to resources across multiple machines. Having more access to machines means more information access and more chances to find more credentials. Another thing to consider is that the different machines may have different weaknesses that could eventually lead to the threat actor gaining elevated privileges taking control of the whole system.

While working on these machines a few things came to mind in terms of preventing people from moving laterally. One of the first things was not giving people access to multiple machines which will prevent a person from using someone else's credentials to gain access to multiple machines. My next recommendation is that of creating different virtual networks so that, if able, you can make is so only machines that need to communicate with each other are able to but are not able to communicate to other machines on the network. Again, make sure that there aren't any openly stored credentials in directories, this was big problem that really make lateral movement in this scenario.