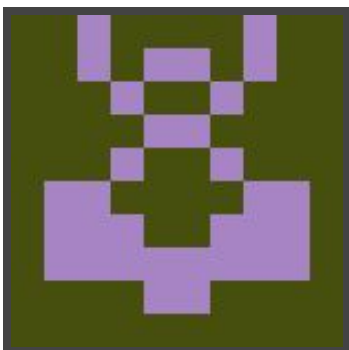




Hack The Box  
PEN-TESTING LABS



# Brainfuck

17<sup>th</sup> October 2017 / Document No D17.100.24

Prepared By: Alexander Reid (Arrexel)

Machine Author: ch4p

Difficulty: **Hard**

Classification: Official



## SYNOPSIS

Brainfuck, while not having any one step that is too difficult, requires many different steps and exploits to complete. A wide range of services, vulnerabilities and techniques are touched on, making this machine a great learning experience for many.

### Skills Required

- Intermediate knowledge of Linux
- Basic understanding of RSA cryptography

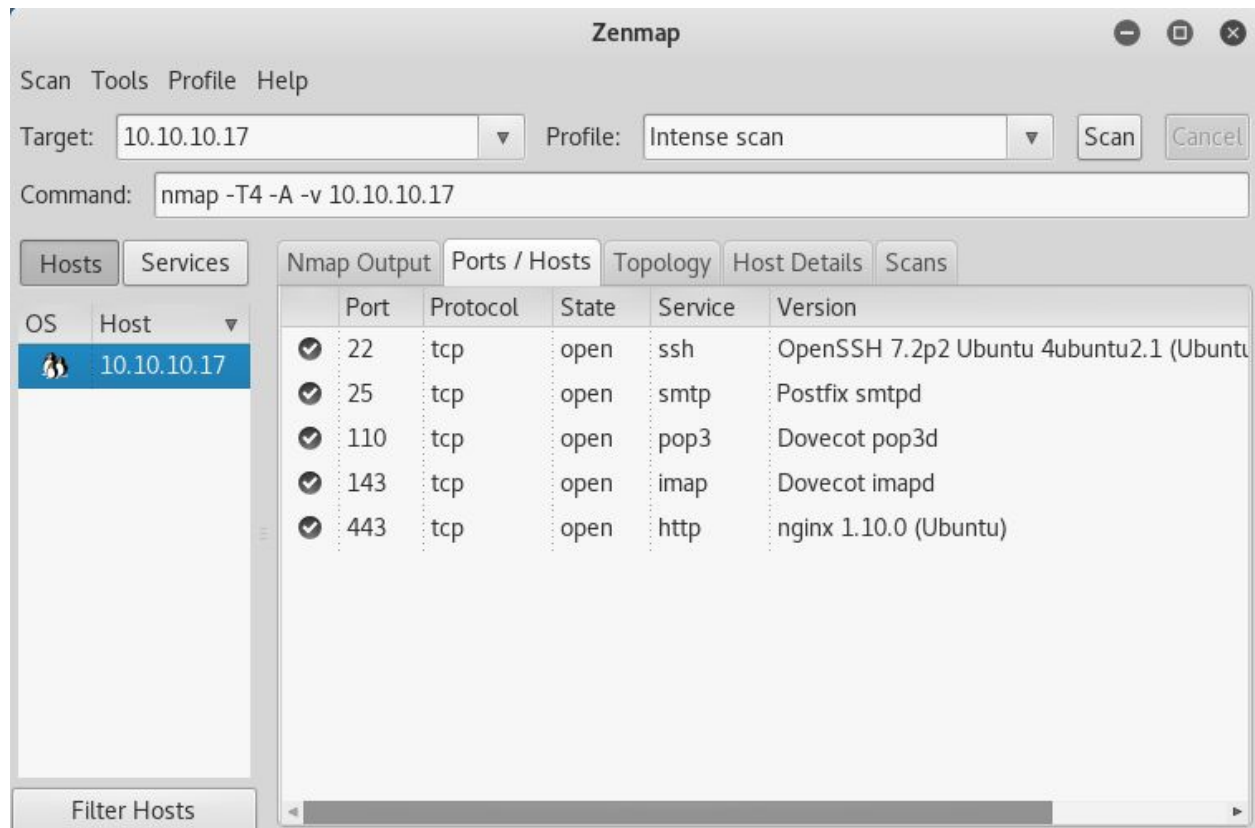
### Skills Learned

- Enumerating SSL certificates
- Exploiting Wordpress
- Exploit modification
- Enumerating mail servers
- Decoding Vigenere ciphers
- SSH key brute forcing
- RSA decryption techniques



## Enumeration

### Nmap



```
|_ ssl-cert: Subject: commonName=brainfuck.htb/  
organizationName=Brainfuck Ltd./stateOrProvinceName=Attica/  
countryName=GR  
| Subject Alternative Name: DNS:www.brainfuck.htb,  
DNS:sup3rs3cr3t.brainfuck.htb  
| Issuer: commonName=brainfuck.htb/organizationName=Brainfuck  
Ltd./stateOrProvinceName=Attica/countryName=GR
```

Nmap reveals several open services as well as several hostnames that were enumerated through the SSL certificate. Adding the hostnames to **/etc/hosts** is required to view the sites.



## WPScan

```
root@kali: ~  
File Edit View Search Terminal Help  
| Last updated: 2017-05-20T13:53:00.000Z  
| Location: https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsiv  
e-ticket-system/  
| Readme: https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-  
ticket-system/readme.txt  
[!] The version is out of date, the latest version is 8.0.7  
[!] Directory listing is enabled: https://brainfuck.htb/wp-content/plugins/wp-su  
pport-plus-responsive-ticket-system/  
  
[!] Title: WP Support Plus Responsive Ticket System <= 7.1.3 – Authenticated SQL  
Injection  
Reference: https://wpvulndb.com/vulnerabilities/8699  
Reference: http://lenonleite.com.br/en/blog/2016/12/13/wp-support-plus-respo  
nsive-ticket-system-wordpress-plugin-sql-injection/  
Reference: https://plugins.trac.wordpress.org/changeset/1556644/wp-support-p  
lus-responsive-ticket-system  
Reference: https://www.exploit-db.com/exploits/40939/  
[i] Fixed in: 8.0.0  
  
[+] Finished: Tue Oct 17 16:29:55 2017  
[+] Requests Done: 55  
[+] Memory used: 76.242 MB  
[+] Elapsed time: 00:00:08  
root@kali:~#
```

WPScan finds an authenticates SQL injection vulnerability, however the results overall do not find anything of much use. Searching Exploit-DB for more exploits related to the ticket system yields <https://www.exploit-db.com/exploits/41006/>



## Exploitation

### Wordpress

Gaining access to the Wordpress admin account is trivial using the above exploit. All that is required is setting the target URL and user. The username, **admin**, can be easily guessed and it is the default username when installing Wordpress. After running the exploit, the admin panel can be accessed at **/wp-admin/**

After gaining access, some credentials can be found on the **Settings > Easy WP SMTP** page. The password can be extracted simply by viewing the page source.

SMTP username

*The username to login to your mail server*

SMTP Password

*The password to login to your mail server*

### Mail Server

Using the credentials obtained from wordpress, it is trivial to extract the emails from the server. Any IMAP-capable mail client or even Telnet can be used here. The example below will use Telnet.

1. telnet brainfuck.htb 143
2. a1 LOGIN orestis kHGuERB29DNiNE
3. a2 LIST "" "\*"
4. a3 EXAMINE INBOX
5. a4 FETCH 1 BODY[]
6. a5 FETCH 2 BODY[]

The second email exposes credentials that can be used to log in at **sup3rs3cr3t.brainfuck.htb**



## Forums

Tool: <http://rumkin.com/tools/cipher/vigenere.php>

Looking at the **Key** discussion, it appears that the post is encrypted. In this case, the cipher used is basic Vigenere. By comparing the last line of text in each of orestis' posts to the posts in the **SSH Access** discussion, it is possible to extract the key.

Decrypt ▼

Passphrase: Orestis Hacking for fun and g

Your message:

Qbqquzs - Pnhekxs dpi fca fhf zdmgzt  
QbqquzsPnhekxsdpifcafhfzdmgzt

This is your encoded or decoded text:

Ckmybra - Infuckm ybr ain fuc kmybra  
CkmybraInfuckmybrainfuckmybra

After a bit of playing around with the output, the key appears to be **fuckmybrain**. Using that, it is possible to decrypt the posts.

Decrypt ▼

Passphrase: fuckmybrain

Your message:

Ybqba wpl gw lto udnju fcpp, C jvbc zfu zrryolap zfu zis rkeqxfri oiwceec J uovg :)  
mnyze://10.10.10.17/8zb5ra10m915218697d1h658wfoq0zc8/frmfycu/sp\_ptr

This is your encoded or decoded text:

There you go you stupid fuck, I hope you remember your key password because I dont :)  
https://10.10.10.17/8ba5aa10e915218697d1c658cdee0bb8/orestis/id\_rsa

The RSA key has a passphrase that must be cracked. This can be achieved by running **ssh2john id\_rsa > id\_john** and then **john id\_john --wordlist=<PATH TO ROCKYOU.TXT>**

```
root@kali:~/Desktop/writeups/brainfuck# ls
brainfuck.html id_john id_rsa
root@kali:~/Desktop/writeups/brainfuck# chmod 600 id_rsa
root@kali:~/Desktop/writeups/brainfuck# ssh -i id_rsa orestis@brainfuck.htb
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-75-generic x86_64)

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

The user flag can be obtained from **/home/orestis/user.txt**





## RSA Decryption

Script: <https://crypto.stackexchange.com/a/19530>

Looking at the contents of the files in the **/home/orestis** directory, specifically **encrypt.sage**, it appears that the file **output.txt** contains an encrypted root flag and the file **debug.txt** contains the P, Q and E values used to do the encryption. By using the above Python script, it is possible to decrypt the ciphertext and get the root flag.

```
root@kali:~/Desktop/writeups/brainfuck# python rsa.py
n: 8730619434505424202695243393110875299824837916005183495711605871599704226978
29509624135727770919760163726737095730026723557679458891077938400356544917133668
55473987716180186966474046572667055368591252274362282022697478098844388858375993
21762997276849457397006548009824608365446626232570922018165610149151977
pt: 2460405202940138604998029695378428707905924586788096694424666284934150700375
0
root@kali:~/Desktop/writeups/brainfuck#
```

To convert the plaintext result from decimal to ASCII, the following command can be used:

```
python -c "print format(<DECIMAL NUMBER>, 'x').decode('hex')"
```

The output of the command is the hash value from **root.txt**.

```
root@kali: ~/Desktop/writeups/brainfuck
File Edit View Search Terminal Help
root@kali:~/Desktop/writeups/brainfuck# python -c "print format(2460405202940138
6049980296953784287079059245867880966944246662849341507003750, 'x').decode('hex'
)"
6efc1a5dbb8904751ce6566a305bb8ef
root@kali:~/Desktop/writeups/brainfuck#
```